

## SIMATIC

### SIMATIC Controls for Computing version 2

#### User Manual

This manual is a part of the product  
with the order number

**6ES7673-2CC00-0YX0**  
**6ES7673-6CC00-0YX0**

**C79000–G7076–C226–01**

**Edition: 1**

Preface, Contents	
SIMATIC Button Control	<b>1</b>
SIMATIC Number Control	<b>2</b>
SIMATIC Slider Control	<b>3</b>
Properties and Methods	<b>A</b>
Events	<b>B</b>
Guidelines for Programming with Computing	<b>C</b>
Index	

## Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:



---

### Danger

indicates that death, severe personal injury or substantial property damage **will** result if proper precautions are not taken.

---



---

### Warning

indicates that death, severe personal injury or substantial property damage **can** result if proper precautions are not taken.

---



---

### Caution

indicates that minor personal injury or property damage can result if proper precautions are not taken.

---

---

### Note

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

---

## Qualified Personnel

Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

## Correct Usage

Note the following:



---

### Warning

This device and its components may only be used for the applications described in the catalog or the technical descriptions, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

---

## Trademarks

SIMATIC®, SIMATIC HMI® and SIMATIC NET® are registered trademarks of SIEMENS AG.

Some of other designations used in these documents are also registered trademarks; the owner's rights may be violated if they are used by third parties for their own purposes.

## Copyright Siemens AG 1999 All rights reserved

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Siemens AG  
Automation and Drives (A&D)  
Industrial Automation Systems (AS)  
Postfach 4848, D- 90327 Nürnberg

## Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

© Siemens AG 1999  
Technical data subject to change.

# Preface

Version 3.0 of the Computing software provides new ActiveX control with new names and new properties, methods and events. To maintain support for programs that were developed with earlier versions of the Computing software, the older controls can be installed. This manual describes these older ActiveX controls:

- Button control
- Number control
- Panel control
- Slider control

## Audience

This manual is intended for engineers, programmers, and maintenance personnel who have a general knowledge of programmable logic controllers (PLCs).

## Scope of the Manual

This manual describes the ActiveX controls of version 2.0 of the Computing software.

## How to Use This Manual

This manual provides information about the ActiveX controls that were shipped with earlier versions of Computing.

## Other Manuals

For additional information, refer to the following manuals:

Title	Content
Computing User Manual (edition 4)	This manual describes the Computing (version 3.0) software.

You can also find information in the online help for the WinAC software. The online help for the STEP 7 programming software provides information about the S7 family of programmable logic controllers (PLCs).

## Additional Assistance

If you have any questions not answered in this or one of the other STEP 7 manuals, if you need information on ordering additional documentation or equipment, or if you need information on training, please contact your Siemens distributor or sales office.

To contact Customer Service for Siemens in North America:

- Telephone:
  - (609) 734–6500
  - (609) 734–3530
- E-mail:
  - ISBU.Hotline@sea.siemens.com
  - simatic.hotline@sea.siemens.com
- Internet:
  - <http://www.aut.sea.siemens.com/winac/>
  - <http://www.aut.sea.siemens.com/simatic/support/index.htm>
  - [http://www.ad.siemens.de/support/html\\_76/index.shtml](http://www.ad.siemens.de/support/html_76/index.shtml)

To contact Customer Service for Siemens in Europe:

- Telephone: ++49 (0) 911 895 7000
- Fax: ++49 (0) 911 895 7001
- E-mail: [simatic.support@nbgm.siemens.de](mailto:simatic.support@nbgm.siemens.de)
- Internet: <http://www.ad.siemens.de/simatic-cs>

# Contents

<b>1</b>	<b>SIMATIC Button Control</b>	<b>1-1</b>
1.1	Connecting the Button Control to the Process Data	1-2
1.2	Properties, Methods, and Events of the Button Control	1-6
<b>2</b>	<b>SIMATIC Number Control</b>	<b>2-1</b>
2.1	Connecting the Number Control to the Process Data	2-2
2.2	Properties and Methods of the Number Control	2-9
2.3	Events of the Number Control	2-11
2.4	Error Codes for the Number Control	2-12
<b>3</b>	<b>SIMATIC Slider Control</b>	<b>3-1</b>
3.1	Connecting the Slider Control to the Process Data	3-2
3.2	Properties, Methods, and Events of the Slider Control	3-5
<b>A</b>	<b>Properties and Methods</b>	<b>A-1</b>
A.1	Alignment Property	A-1
A.2	Appearance Property	A-1
A.3	BackColor Property	A-2
A.4	BorderStyle Property	A-3
A.5	Caption1 and Caption2 Properties	A-4
A.6	Caption1Alignment and Caption2Alignment Properties	A-4
A.7	Caption1BackColor, Caption2BackColor Properties	A-5
A.8	Caption1Font and Caption2Font Properties	A-5
A.9	Caption1ForeColor, and Caption2ForeColor Properties	A-6
A.10	Caption1Size and Caption2Size Properties	A-6
A.11	ConvertedValue Property	A-7
A.12	DataType Property	A-8
A.13	Direction Property	A-9
A.14	Enabled Property	A-9
A.15	Factor Property	A-10
A.16	FalseCaption Property	A-11
A.17	FalseColor Property	A-11
A.18	Font Property	A-12
A.19	ForeColor Property	A-12
A.20	LargeChange Property	A-13
A.21	LayoutType Property	A-14

A.22	LimitCheck Property .....	A-14
A.23	Locked Property .....	A-15
A.24	LowerLimit Property .....	A-16
A.25	Max Property .....	A-17
A.26	Min Property .....	A-17
A.27	Offset Property .....	A-18
A.28	Precision Property .....	A-18
A.29	Radix Property .....	A-19
A.30	RangeRawMax, RangeRawMin, RangeScaledMax, and RangeScaledMin Properties .....	A-20
A.31	ScaleMode Property .....	A-21
A.32	ShowButtons Property .....	A-22
A.33	ShowErrorBoxes Property .....	A-23
A.34	SmallChange Property .....	A-24
A.35	TagServerName Property .....	A-24
A.36	Ticks Property .....	A-24
A.37	TrueCaption Property .....	A-25
A.38	TrueColor Property .....	A-25
A.39	UpperLimit Property .....	A-26
A.40	Value Property .....	A-27
A.41	Write Method .....	A-28
A.42	WriteMode Property .....	A-28
A.43	ZeroPad Property .....	A-29
<b>B</b>	<b>Events of the SIMATIC Controls .....</b>	<b>B-1</b>
B.1	Change Event .....	B-1
B.2	Click Event .....	B-1
B.3	DbClick Event .....	B-1
B.4	Error Event .....	B-2
B.5	KeyDown Event .....	B-3
B.6	KeyPress Event .....	B-4
B.7	KeyUp Event .....	B-4
B.8	LimitCheckFailed Event .....	B-5
B.9	MouseDown Event .....	B-6
B.10	MouseMove Event .....	B-7
B.11	MouseUp Event .....	B-8

<b>C</b>	<b>Guidelines for Programming with Computing .....</b>	<b>C-1</b>
C.1	Guidelines for Third-Party Containers .....	C-2
C.2	Programming Guidelines .....	C-3
	<b>Index .....</b>	<b>Index-1</b>





# SIMATIC Button Control

## Chapter Overview

The Button control allows you to associate a button display with a data bit from your process. You associate the button with your process by assigning a variable (namely, the desired bit location) to it. You can then toggle the button display to change the state of the bit; the button color also changes automatically as the state of the bit changes within the process.

The Button control provides access to individual memory bits in the control engine. Figure 1-1 shows the toolbox icon and the Button control. The Button control has two states of animation: 0 (off) or 1 (on). Clicking on the Button control changes the data in the control engine.



Figure 1-1 SIMATIC Button Control

The SIMATIC Button control reads and writes Boolean (single bit) values.

Section	Description	Page
1.1	Connecting the Button Control to the Process Data	1-2
1.2	Properties, Methods, and Events of the Button Control	1-6

## 1.1 Connecting the Button Control to the Process Data

To establish a connection between the Button control and your process data, you assign a (single-bit) variable to the Value property of the Button control. The variable cannot be assigned within the Properties dialog box of the Button control. Instead, use the Properties dialog box of the Data control and select the button from the expandable list of controls under the Connections tab. See Figure 1-2.

To set properties for anything other than the Value property, you can use the Properties dialog box of the Button control itself. Use the **Edit** menu or right click the mouse button and select the **Properties** command for the control.



### Warning

After you assign a variable to the Value property of a SIMATIC or a third-party ActiveX control, the control is able to access process data. When you change the value that is displayed in the control, you are changing the value in the actual process.

Altering process data can cause unpredictable process operation, and unpredictable process operation could result in death or serious injury to personnel, and/or damage to equipment.

Exercise caution to ensure that you do not access any data that could cause process equipment to operate erratically. Always install a physical emergency stop circuit for your machine or process.

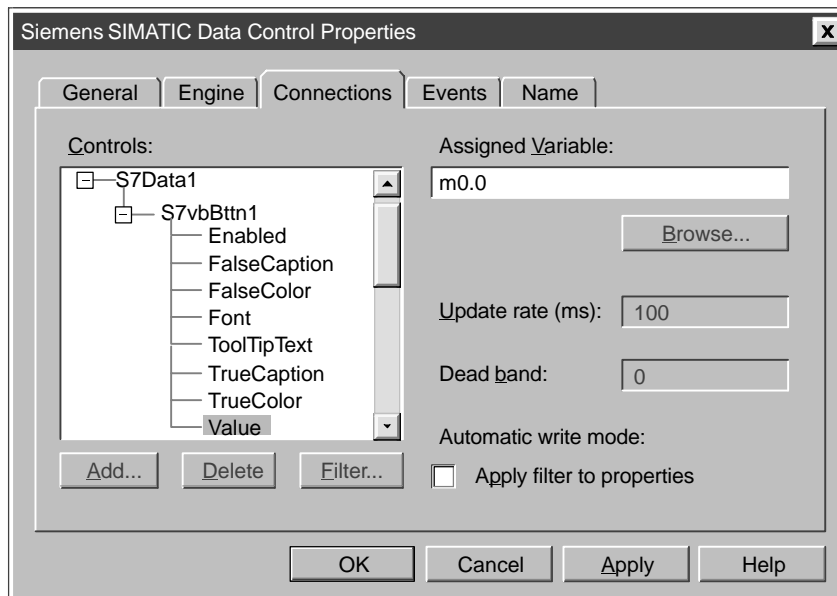


Figure 1-2 Assigning Variables for the Button Control

While Computing allows you to specify a data type when you assign a variable to one of the properties of an S7 control, remember that the Button control can be assigned only to an individual bit in the control engine. The only valid data type for an Button control is BOOL.

### Defining the Label and Enabling the Control (Using the General Tab)

The General tab of the Properties dialog (see Figure 1-3) allows you to define the two labels for the Button control:

- “TrueCaption”: Enter the text to be displayed in the control when the bit is “true” (equal to 1 or “on”).
- “FalseCaption”: Enter the text to be displayed in the control when the bit is “false” (equal to 0 or “off”).

The General tab also includes a check box for enabling the control. When you select this option, the control can actively access the control engine. Deselecting this option deactivates the control: it does not access the memory location of the control engine and does not react on any mouse clicks. It also does not generate events while disabled. The default setting for this option is enabled (selected).

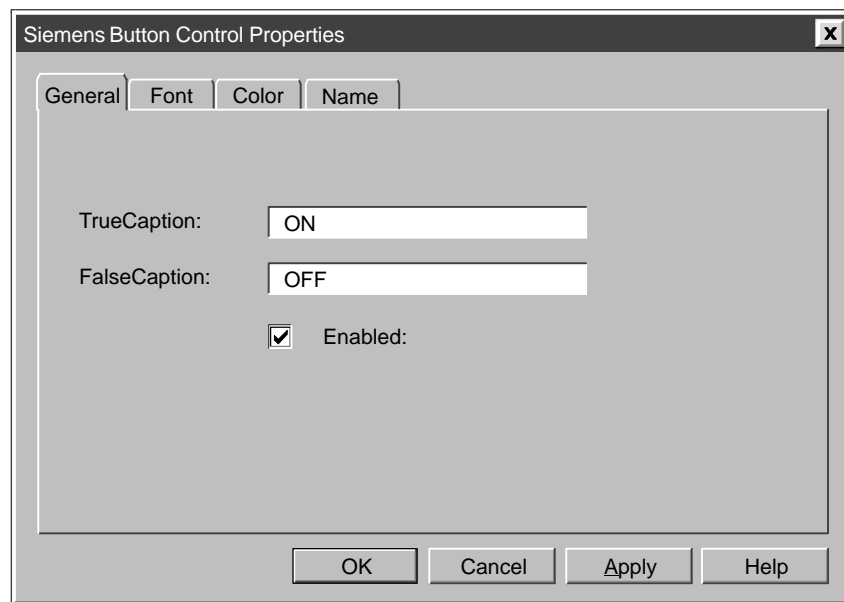


Figure 1-3 Button Control Properties (General Tab)

### Defining the Typeface of the Label (Using the Font Tab)

The Font tab of the Properties dialog (see Figure 1-4) allows you to define the typeface and size for the labels of the Button control:

- “Font”: select the typeface for the label from a list of standard typefaces.
- “Size”: select the point size for the label or enter a specific point size for the label.
- “Effects”: select other typographical options (boldface, italic, underline, or strike-through) for the label.

The “Sample Text” field displays the selection of the Font property.

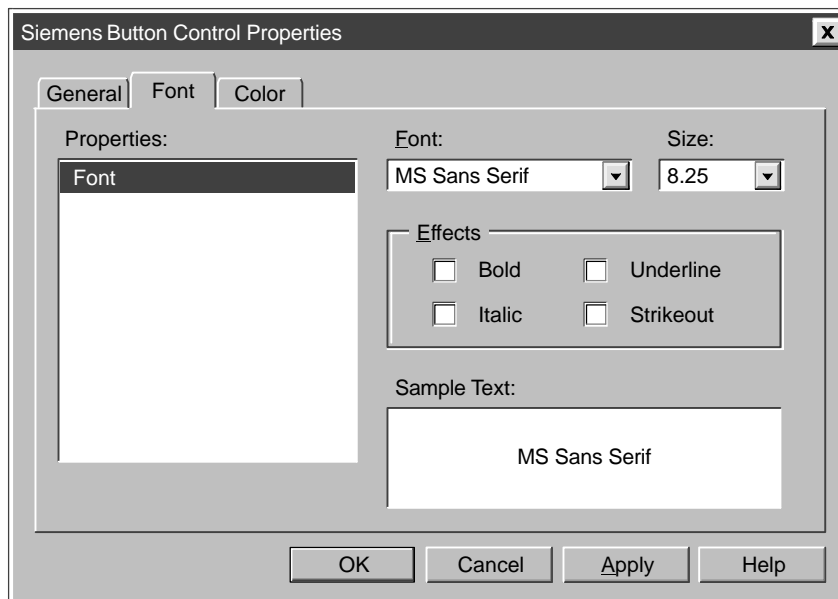


Figure 1-4 Button Control Properties (Font Tab)

### Defining the Color of the Control (Using the Color Tab)

The Color tab of the Properties dialog (see Figure 1-5) allows you to define the colors for the two states of the Button control. You select the “Off” state (FalseColor) or the “On” state (TrueColor) and then select the color to be displayed for that state from the color palette. You can choose from a palette of standard colors, or you can create custom colors.

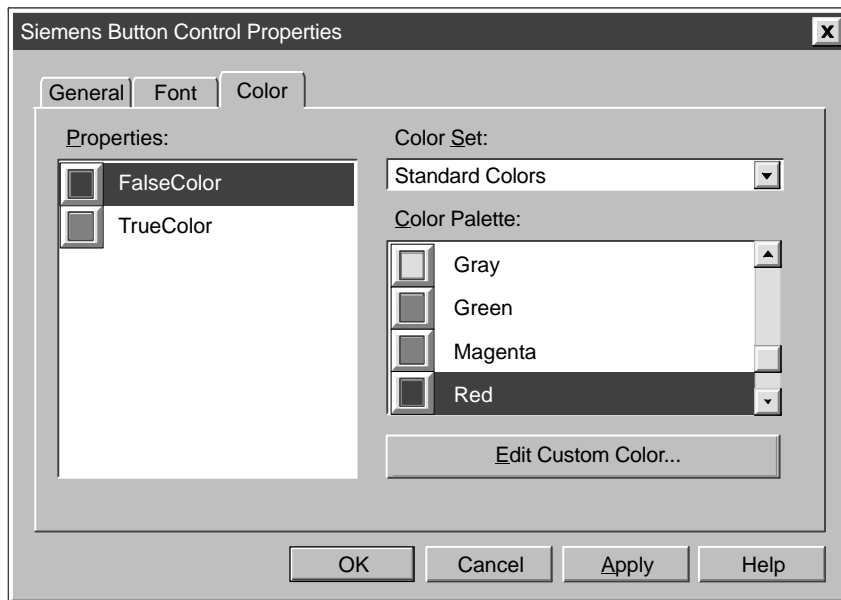


Figure 1-5 Button Control Properties (Color Tab)

## 1.2 Properties, Methods, and Events of the Button Control

You use the properties and methods listed in Table 1-1 to manipulate the Button control. The control responds to the events listed in Table 1-2.

Table 1-1 Properties and Methods of the Button Control

Property or Method	Description	Page
Enabled property	Determines whether the control reacts to changes of the Value property and fires events	A-9
FalseCaption property	Determines the text that is displayed in the control when the Value property is False (equal to 0, or "Off")	A-11
FalseColor property	Determines the color of the control when the Value property is False (equal to 0, or "Off")	A-11
Font property	Returns a Font object for the main font of the control	A-12
TrueCaption property	Determines the text that is displayed in the control when the Value property is True (equal to 1, or "On")	A-25
TrueColor property	Determines the color of the control when the Value property is True (equal to 1, or "On")	A-25
Value property	Contains the value that is linked to the control engine	A-27

Table 1-2 Events of the Button Control

Event	Description	Page
Change event	Occurs when the value of the Value property changes	B-1
Click event	Occurs when a mouse button is pressed and released while the mouse cursor is over the control	B-1

# SIMATIC Number Control

# 2

## Chapter Overview

The Number control allows you to display process data in a numeric format and to modify that data. You associate the number display with your process by assigning a variable (the process value) to it. You can type a new value into the display; the display also updates automatically when the variable associated with it changes within the process.

The Number control provides access to the memory locations of the control engine. Entering a new value in the control changes the data in the control engine. Figure 2-1 shows the toolbox icon and the Number control.

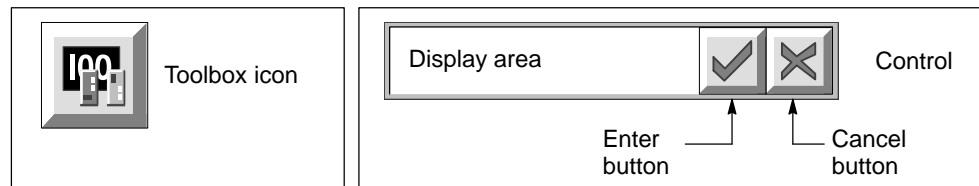


Figure 2-1 SIMATIC Number Control

---

### Note

Computing does not allow you to write to timers.

---

Section	Description	Page
2.1	Connecting the Number Control to the Process Data	2-2
2.2	Properties and Methods of the Number Control	2-9
2.3	Events of the Number Control	2-11
2.4	Error Codes for the Number Control	2-12

## 2.1 Connecting the Number Control to the Process Data

To establish a connection between the Number control and your process data, you assign a variable to the Value property of the Number control. The variable cannot be assigned within the Properties dialog box of the Number control. Instead, use the Properties dialog box of the Data control and select the number display from the expandable list of controls under the Connections tab. See Figure 2-2.

### Note

In order to connect the Number control to actual process data, you must establish a connection through the Data control.

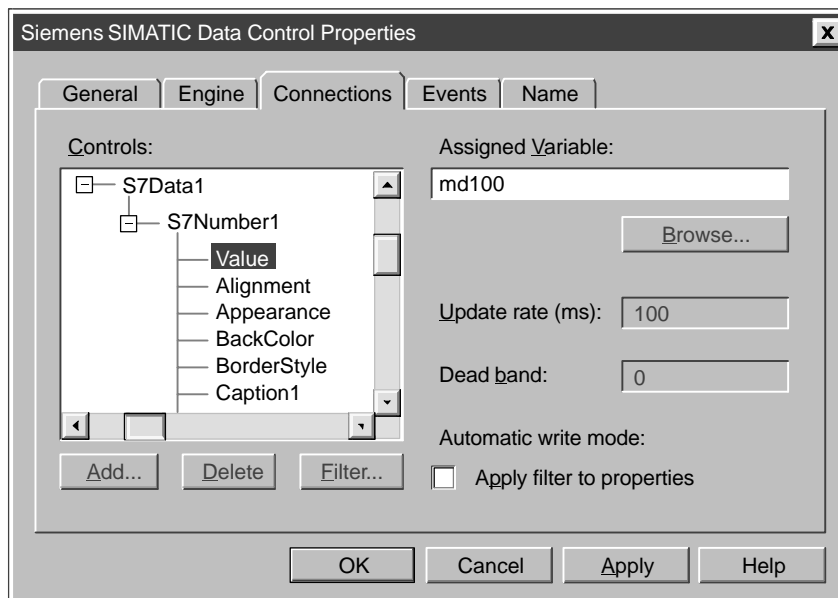


Figure 2-2 Assigning Variables for the Number Control

Computing allows you to specify a data type when you assign a variable to one of the properties of a SIMATIC control. You define the data type by entering the absolute address for the memory location, followed by a colon (:) and the data type. For example, you can define an assigned variable as a REAL data type by entering "MD100:real" when you assign the variable. You also have to set the Datatype property for the Number control to match this data type designation.





### Warning

After you assign a variable to the Value property of a SIMATIC or a third-party ActiveX control, the control is able to access process data. When you change the value that is displayed in the control, you are changing the value in the actual process.

Altering process data can cause unpredictable process operation, and unpredictable process operation could result in death or serious injury to personnel, and/or damage to equipment.

Exercise caution to ensure that you do not access any data that could cause process equipment to operate erratically. Always install a physical emergency stop circuit for your machine or process.

To set properties for anything other than the Value property, you can use the Properties dialog box of the Number control itself. Use the **Edit** menu or right click the mouse button and select the **Properties** command for the control.

### Defining How the Data is Displayed (Using the General Tab)

The General tab of the Properties dialog box (see Figure 2-3) allows you to define the presentation of the data accessed by the Number control.

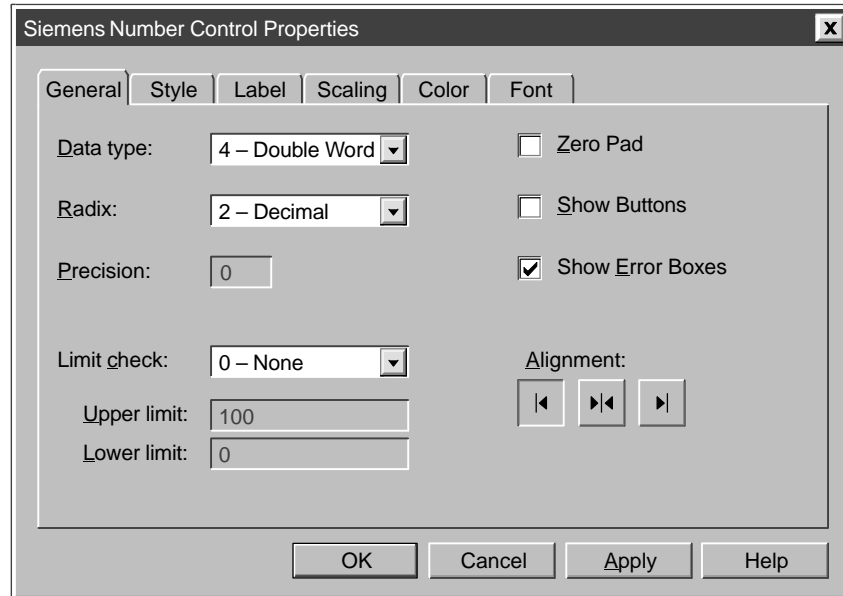


Figure 2-3 Number Control Properties (General Tab)

The fields on the General tab allow you to define the following properties concerning how the data will be displayed:

- “Data type” defines the size of the presentation for the data. This entry does not affect the size of the memory location being accessed; the control displays the value from the control according to the size configured by this field. Table 2-1 describes the sizes of the data types used by the Number control.

If you entered a specific data type for the assigned variable in the SIMATIC Data control, you must enter the same data type here.

- “Radix” configures the data to be accessed (displayed or entered) as a binary, octal, decimal or hexadecimal value.
- “Precision” defines the decimal place for the real (floating-point) number. You enter the number of digits to the right of the decimal. (The default value is three digits.) This field is enabled only for the Real data type.

Table 2-1 Size of Data Types for the Number control

Data Type	Setting	Size	Description
Boolean	0	1 bit	Single bit value
Byte	1	1 byte	Unsigned single-byte value
Word	2	1 byte	Unsigned two-byte value
Integer	3	2 bytes	Signed two-byte integer value
Double Word	4	4 bytes	Unsigned four-byte value (default)
Double Integer	5	4 bytes	Signed four-byte integer value
Real	6	4 bytes	Signed four-byte real (floating-point) value
Timer	7	2 bytes	Unsigned two-byte value
Counter	8	2 bytes	Unsigned two-byte value

Using the alignment buttons, you can define how the value will be displayed in the Number control: aligned to the left side of the field, centered in the field, or aligned to the right side of the field.

You can also enable the Number control to check for upper or lower limits on the value that you enter. You can select limit checking for a maximum value, a minimum value, both maximum and minimum value, or for no limit checking. When you enable the Limit Checking option, the Number control does not write an out-of-range value (larger than the upper limit or smaller than the lower limit) to the control engine. If the Number control has been configured to show the error messages, the out-of-range value generates an error message.

---

**Note**

Both Visual Basic and the SIMATIC Number control use the keyword "Right":

- As described in Sections A.1 and C.2, the SIMATIC Number control uses an alignment constant to determine the appearance (for example, to align the number to the right edge of the field). The buttons that select these alignment constants appear on the "General" tab of the "Properties" dialog box for the Number control.
- Visual Basic provides a string function Right that returns a specified number of characters from a string.

When you insert a Number control onto a Visual Basic form, the Right alignment constant of the Number control overrides the Right string function of Visual Basic. In order to use the Right string function of Visual Basic, you must explicitly identify the Visual basic Right function by entering: **VBA.Right**

---

Using the check boxes on the General tab, you can define other operations for the control:

- "Zero Pad": When you enable this option, the Number control fills out the data type by inserting zeroes (0) to the left of the value. The default setting for this option is disabled (not selected).
- "Show Buttons": When you enable this option, the Number control displays the Enter and Cancel buttons. (See Figure 2-3.) When this option is checked, you must confirm any new value in order to enter it. Typing a new value in the display field of the control enables the buttons. Clicking on the Enter button writes the new value to the control engine; clicking on the Cancel button replaces the new value with the value from the control engine. The default setting for this option is disabled (not selected).

You can also use the Return (or Enter) key on the keyboard instead of the Enter button, and the Escape (Esc) key instead of the Cancel button.

- "Show Error Boxes": when you enable this option, the Number control displays the messages ("error boxes") in response to pre-defined error conditions (for example, for out-of-range or invalid numbers). The default setting for this option is enabled (selected).

## Defining the Appearance of the Number Control

The Style tab of the Properties dialog box (see Figure 2-4) provides you with the following options for the appearance of the control:

- "Appearance": If you set this property to 3D, the control will have a three-dimensional appearance. (You must also set the border style to Fixed Single to enable the three-dimensional appearance.) The other option is Flat, which displays a two-dimensional, rectangular border around the control.
- "Border Style": If you set this property to Fixed Single, the control is displayed with a rectangular border; if you set the property to None, no border will be displayed.

In addition to fields that determine the appearance of the control, the Style tab also includes a “Write Mode” field that controls whether changes to the value are written automatically to the control engine, or are written only when requested by the program code (using a Write command) that you associated with the Number control.

Using the check boxes on the Style tab, you can define other operations for the Number control:

- “Enabled”: When you enable this option, the control can actively access the control engine. Disabling this option deactivates the Number control: it does not access the memory location of the control engine and does not react on any mouse clicks. It also does not generate events while disabled. The default setting for this option is enabled (selected).
- “Locked”: When you enable this option, the control becomes a read-only display: you can view the value in the memory location of the control engine, but you cannot change the values from this control. The default setting for this option is disabled (not selected).

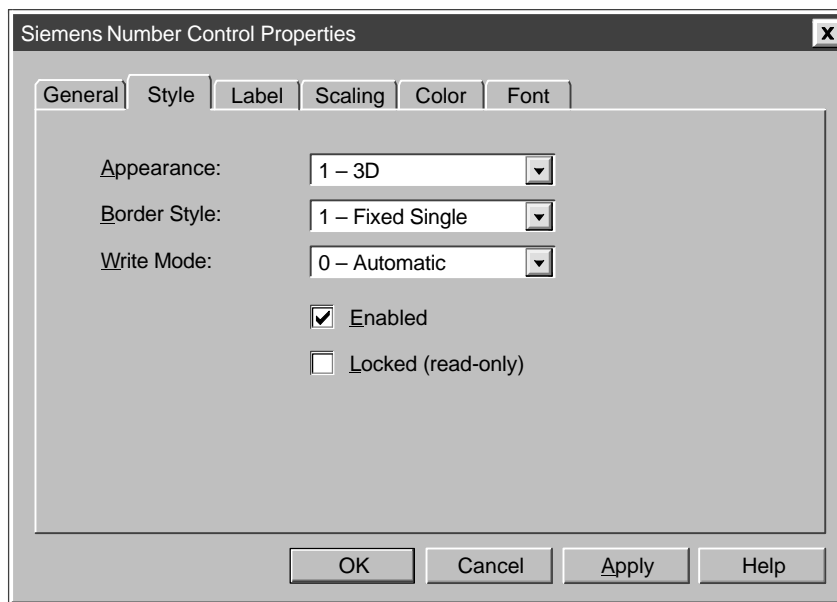


Figure 2-4 Number Control Properties (Style Tab)

### Creating a Label for the Number Control (Using the Label Tab)

The Label tab of the Properties dialog box (see Figure 2-5) allows you to define a label (or caption) for the control. You can define two different captions for the control (“Caption 1” and “Caption 2”).

As shown in Figure 2-6, the labels can be oriented either horizontally or vertically in relationship to the Number control:

- If you select Horizontal, Caption 1 is on the left and Caption 2 is on the right side of the control.
- If you select Vertical, Caption 1 on the top and the Caption 2 on the bottom.

Use the alignment buttons for each caption to determine whether the label is centered, aligned to the left, or aligned to the right.

Use the Font tab and the Color tab to specify the font, the font color, and the background color of the label.

You can also specify a specific width (for horizontal labels) or height (for vertical labels). Figure 2-6 shows the labeling options for the Number control.

If you keep the default value for the size of the label (–1), the Number control automatically adjusts the size of the label to the amount text entered or to the font size for the caption.

Entering a specific size for the labels allows you to control the appearance of the Number control. For example, you may want to align the display area for a series of Number controls, but the labels vary in length for each control. Choosing a constant size for the labels would align the display areas.

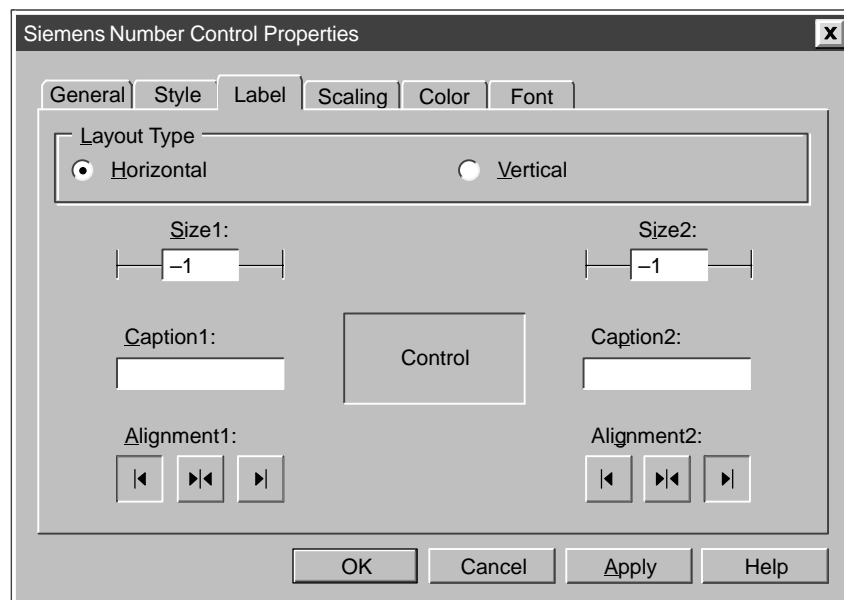


Figure 2-5 Number control Properties (Label Tab)

If you enter a specific size for the captions, you enter the size in twips. (A twip is a screen-independent unit that ensures that the placement and proportion of screen elements are the same on all display systems. A twip is approximately  $\frac{1}{20}$  of a printer's point: 1440 twips=1 in. or 56.7 twips=1 mm when printed.)

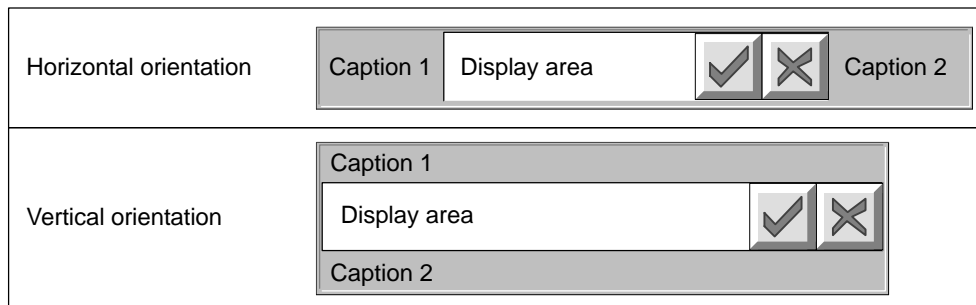


Figure 2-6 Labeling Options for the Number Control

## Using the Scaling Tab

The Scaling tab of the Properties dialog box (see Figure 2-7) allows you to define a scale for displaying the value in the memory location. This scaling factor is used both in reading a value from and writing a value to the control engine. You can select one of three scaling options:

- No scaling of the data (default)
- Scaling by formula
- Scaling by ranges

If you choose to scale by formula, you enter the following information:

- Scale represents a percentage of change (scaling factor) from the value in the control engine to the value in the Number control.
- Offset represents a fixed value to be added to the scaled result before being displayed.

The Number control uses the following formula to calculate the scaled value:

$$(\text{PLC} \times \text{Scale}) + \text{Offset} = \text{Display}$$

where:

PLC = the value stored in the control engine

Scale = the scaling factor

Offset = the offset factor

Display = the value displayed in the Number control

When the Number control writes data to the control engine, the inverse of the formula is used to scale the value.

If you choose to scale by range transformation, you specify the upper and lower values for a source range (for the value in the control engine) and for a destination range (for the value displayed in the Number control or Display). The Number control then transforms the value from one range into the equivalent value for the other range.

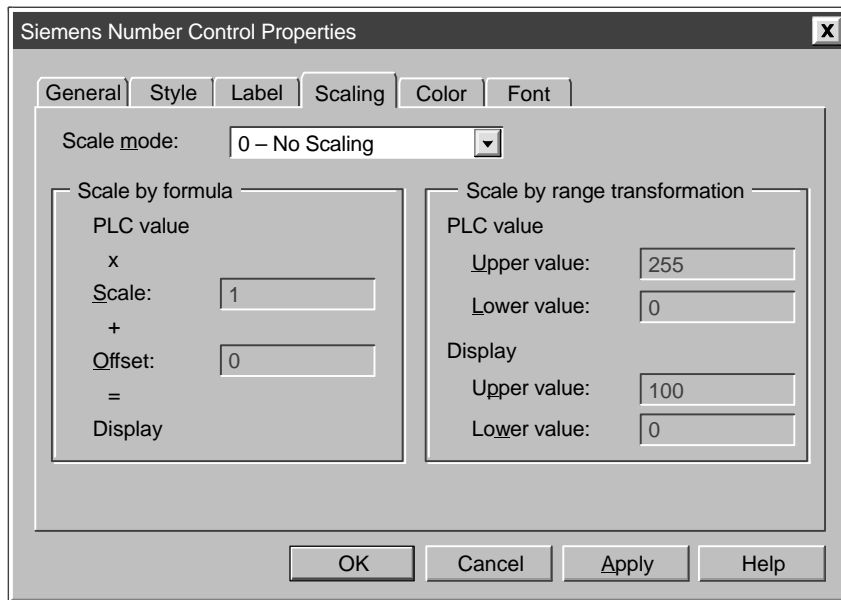


Figure 2-7 Number Control Properties (Scaling Tab)

These ranges define only the relationship between the data in the control engine and the data in the Number control: if the value is above or below the ranges entered for the transformation, the transformation uses the formula to extrapolate the scaled value. The upper and lower limits are not “minimum and maximum values” for the data: there is no “limit checking” with the scaling factors.

## 2.2 Properties and Methods of the Number Control

You use the properties and methods listed in Table 2-2 to manipulate the Number control.

Table 2-2 Properties and Methods of the Number Control

Property or Method	Description	Page
Alignment property	Specifies the alignment of the number in the control	A-1
Appearance property	Specifies whether the control is displayed 3D or flat	A-1
BackColor property	Returns or sets the background color	A-2
BorderStyle property	Selects the border style (fixed single, or none)	A-3
Caption1 and Caption2 properties	Specifies the displayed text of the first or second label	A-4
Caption1Alignment and Caption2Alignment properties	Specifies the alignment of the first or second label	A-4
Caption1BackColor and Caption2BackColor properties	Returns or sets the background color for the first or second label	A-5

Table 2-2 Properties and Methods of the Number Control, continued

Property or Method	Description	Page
CaptionFont property	Returns a Font object for the labels	A-5
Caption1ForeColor and Caption2ForeColor properties	Returns or sets the foreground color for the first or second label	A-6
Caption1Size and Caption2Size properties	Selects the size of the font for the text of the first or second label	A-6
ConvertedValue property	Holds the value from the control engine (read-only)	A-7
DataType property	Defines the storage type used for converted values	A-8
Enabled property	Determines whether the control reacts to changes of the Value property and fires events	A-9
Factor property	Specifies the scaling factor used when the scale-by-formula option has been enabled (Used with ScaleMode property)	A-10
Font property	Returns a Font object for the main font of the control	A-12
ForeColor property	Returns or sets the foreground color used to display text and graphics	A-12
LayoutType property	Specifies how the labels are arranged around the display area	A-14
LimitCheck property	Determines whether to check that a value is within specified limits, and to define the range of values that determine the limits	A-14
Locked property	Sets the control to a read-only state	A-15
LowerLimit property	Allows access to the lower limit value	A-16
Offset property	Specifies the offset used when the scale-by-formula option has been enabled (Used with ScaleMode property)	A-18
Precision property	Selects the precision of the Real number	A-18
Radix property	Selects the radix for the number to be displayed (binary, octal, decimal, or hexadecimal)	A-19
RangeRawMax, RangeRawMin, RangeScaledMax, and RangeScaledMin properties	Defines the ranges for scaling a value by a "range transformation" by specifying the upper and lower limits for the source and the upper and lower limits for the destination	A-20
ScaleMode property	Specifies the scaling mode to be used for scaling the values	A-21
ShowButtons property	Displays the Enter and Cancel buttons for confirming the entering of new values	A-22
ShowErrorBoxes property	Specifies whether to display the default error boxes when there is a user-generated error	A-23
TagServerName property	Determines the name of the Data control which shall be used for a connection to the control engine	A-24
UpperLimit property	Allows access to the upper limit value	A-26



Table 2-2 Properties and Methods of the Number Control, continued

Property or Method	Description	Page
Value property	Contains the value that is linked to the control engine	A-27
Write method	Writes the value of the ConvertedValue property to the Value property	A-28
WriteMode property	Selects whether to write new values automatically or manually	A-28
ZeroPad	Determines whether the displayed number is padded with zeroes (to the left of the value) to the size of the data type	A-29

## 2.3 Events of the Number Control

The Number control responds to the events listed in Table 2-3.

Table 2-3 Events of the Number Control

Event	Description	Page
Change	Occurs when the value of the Value property changes	B-1
Click	Occurs when a mouse button is pressed and released while the mouse cursor is over the control	B-1
DbClick	Occurs when a mouse button is double-clicked while the cursor is over the control	B-1
Error	Occurs when the control encounters an error	B-2
KeyDown	Occurs when the user presses a key while the control has the focus	B-3
KeyPress	Occurs when an ANSI key is pressed and released while the control has the focus	B-4
KeyUp	Occurs when a key is released while the control has the focus	B-4
LimitCheckFailed	Occurs when the limit check fails	B-5
MouseDown	Occurs when a mouse button is pressed while the mouse cursor is over the control	B-6
MouseMove	Occurs when the mouse cursor moves over the control	B-7
MouseUp	Occurs when a mouse button is released while the mouse cursor is over the control	B-8

## 2.4 Error Codes for the Number Control

When an error occurs in the Number control, the control generates an Error event. Your program can capture this Error event and respond to specific situations. Table 2-4 lists the error codes for the Number control.

Table 2-4 Error Codes for the Number Control

Error Code	Description
C0040002	<p>The scaling cannot proceed because of an error in the formula used.</p> <p>This error only appears if you are using the Number control with range scaling. In this case it is possible that you have specified a raw value range (RangeRawMin, RangeRawMax) of the length of zero (<i>min</i> equal to <i>max</i>). This would lead to a division by zero, which means the scaling is impossible.</p> <p>To correct the error, specify a raw value range where RangeRawMin is not equal to RangeRawMax.</p>
C0040003	<p>The set value at the Value property is invalid.</p> <p>The value which came from the control engine or from a script that is accessing the Value property is not interpretable.</p> <p>To correct the error, check the values that you have written to the control.</p>
C0040004	<p>The set value at the Text property is invalid.</p> <p>This is a common error, which occurs if the user enters an incorrect value in the control. Normally, it means that the entered text contains characters that are not allowed.</p> <p>The allowed characters are dependent on the Radix and DataType used. For a description of the allowed characters, see the description of the Radix property.</p> <p>To correct the error, reenter a value that is allowed.</p>
C0040005	<p>The other OLE components could not be found.</p> <p>An error occurred in the installation of Computing or of Windows itself. The control is unable to access the other necessary parts that are needed for the software to work properly.</p> <p>To correct the error, check the installation.</p>
C0040006	<p>The Microsoft standard controls could not be created.</p> <p>Something went wrong with the installation of Computing or Windows itself. The control is unable to access the other necessary parts that are needed for the software to work properly.</p> <p>To correct the error, check the installation.</p>
C0040010	<p>The limit check cannot proceed, because the lower limit is greater than the upper limit.</p> <p>This error can only appear if you are using the Number control with limit checking (checking for upper and lower limit). In this case it is possible that you've specified a lower limit (LowerLimit) that is greater than the upper limit (UpperLimit).</p> <p>To correct the error, specify a valid range for limit checking. The lower limit has to be less than the upper limit.</p>

# SIMATIC Slider Control

## Chapter Overview

The Slider control allows you to display process data in a visual format (as a sliding indicator) and to modify that data. You associate the slider with your process by assigning a variable (the process value) to it. You can then adjust the slider indicator in order to modify the process value; the slider also changes its indicator position automatically as the variable associated with it changes within the process.

The Slider control provides access to the memory locations of the control engine. Entering a new value in the control changes the data in the control engine. Figure 3-1 shows the toolbox icon and the Slider control.

---

### Note

Computing does not allow you to write to timers.

---

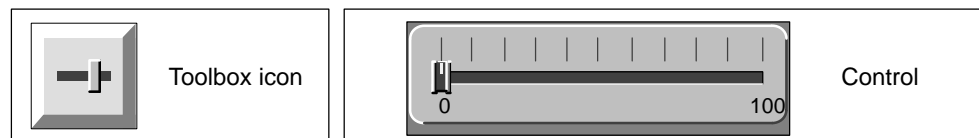


Figure 3-1 SIMATIC Slider Control

Section	Description	Page
3.1	Connecting the Slider Control to the Process Data	3-2
3.2	Properties, Methods, and Events of the Slider Control	3-5

### 3.1 Connecting the Slider Control to the Process Data



#### Warning

After you assign a variable to the Value property of a SIMATIC or a third-party ActiveX control, the control is able to access process data. When you change the value that is displayed in the control, you are changing the value in the actual process.

Altering process data can cause unpredictable process operation, and unpredictable process operation could result in death or serious injury to personnel, and/or damage to equipment.

Exercise caution to ensure that you do not access any data that could cause process equipment to operate erratically. Always install a physical emergency stop circuit for your machine or process.

To establish a connection between the Slider control and your process data, you assign a variable to the Value property of the control. The variable cannot be assigned within the Properties dialog box of the Slider control. Instead, use the Properties dialog box of the SIMATIC Data control and select the slider from the expandable list of controls under the Connections tab. See Figure 3-2.

#### Note

In order to connect the Slider control to actual process data, you must establish a connection through the SIMATIC Data control.

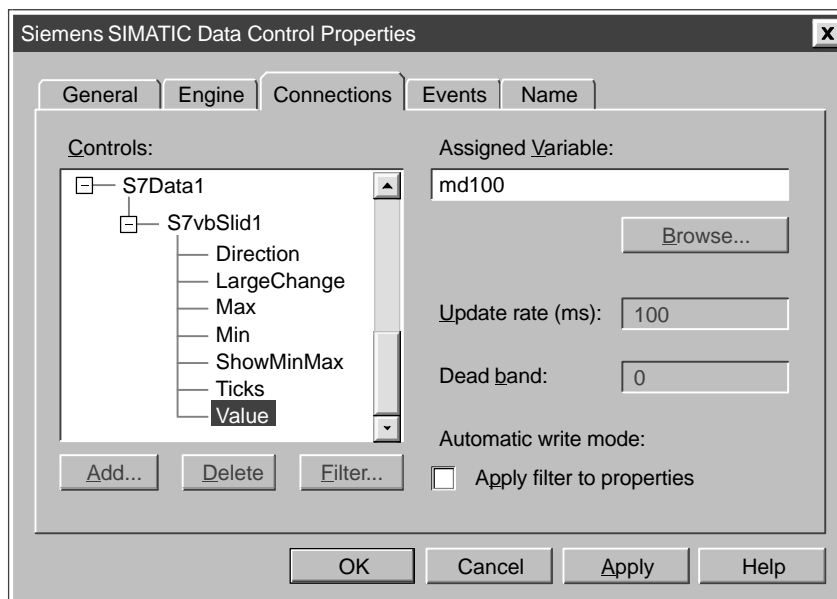


Figure 3-2 Assigning Variables for the Slider Control

Computing allows you to specify a data type when you assign a variable to one of the properties of a SIMATIC control. You define the data type by entering the absolute address for the memory location, followed by a colon (:) and the data type. For example, you can define an assigned variable as a REAL data type by entering "MD100:real" when you assign the variable.

To set properties for anything other than the Value property, you can use the Properties dialog box of the Slider control. Use the **Edit** menu or right click the mouse button and select the **Properties** command for the control.

### Defining How the Data is to be Displayed (Using the General Tab)

The General tab of the Properties dialog box (see Figure 3-3) allows you to define the presentation of the data accessed by the Slider control.

The Properties dialog box also allow you to determine the amount that the value displayed by the Slider control increases or decreases when you press an arrow key (SmallChange) or press the Page Up and Page Down keys (LargeChange).

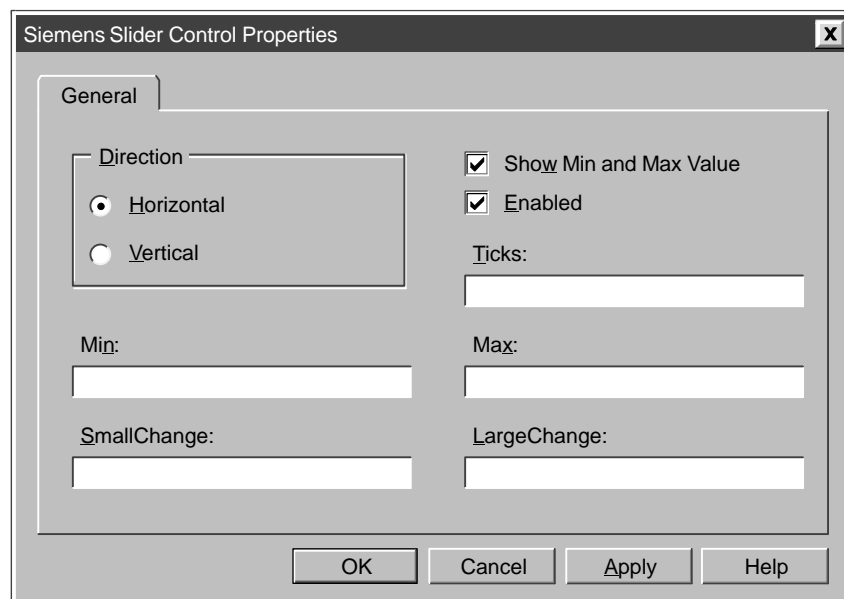


Figure 3-3 Slider Control Properties (General Tab)

Figure 3-4 shows the other elements that are defined by the Properties dialog box. You can enter the minimum (Min) and maximum (Max) values for the Slider control, and you can choose whether or not to display these values. You can also define the number of interim units (ticks) between the minimum and maximum values.

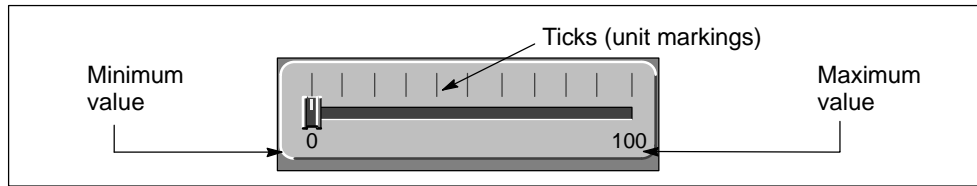


Figure 3-4 Elements of the Slider Control

As shown in Figure 3-5, the Slider control can be displayed in a vertical or horizontal orientation. You choose the orientation by selecting either a horizontal or vertical direction.

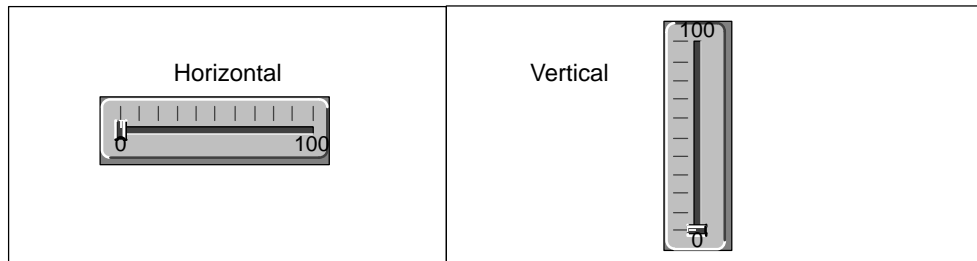


Figure 3-5 Orientation of the Slider Control

## 3.2 Properties, Methods, and Events of the Slider Control

You use the properties and methods listed in Table 3-1 to manipulate the Slider control. The control responds to the events listed in Table 3-2.

Table 3-1 Properties and Methods of the Slider Control

Property or Method	Description	Page
Direction property	Sets the orientation (horizontal or vertical)	A-9
Enabled property	Determines whether the control reacts to changes of the Value property and fires events	A-9
LargeChange property	Determines how far the slider indicator moves when the control has focus and you press the Page Up or Page Down key	A-13
Max property	Determines the value of the rightmost (or top) position of the slider indicator	A-17
Min property	Determines the value of the leftmost (or bottom) position of the slider indicator	A-17
SmallChange property	Determines how far the slider indicator moves when the control has focus and you press the up/down or right/left arrow keys	A-24
Ticks property	Sets the number of ticks (unit markers)	A-24
Value property	Contains the value that is linked to the control engine	A-27

Table 3-2 Events of the Slider Control

Event	Description	Page
Change event	Occurs when the value of the Value property changes	B-1
Click event	Occurs when a mouse button is pressed and released while the mouse cursor is over the control	B-1
DbClick event	Occurs when a mouse button is double-clicked while the cursor is over the control	B-1
KeyDown event	Occurs when the user presses a key while the control has the focus	B-3
KeyPress event	Occurs when an ANSI key is pressed and released while the control has the focus	B-4
KeyUp event	Occurs when a key is released while the control has the focus	B-4
MouseDown event	Occurs when a mouse button is pressed while the mouse cursor is over the control	B-6
MouseMove event	Occurs when the mouse cursor moves over the control	B-7
MouseUp event	Occurs when a mouse button is released while the mouse cursor is over the control	B-8





## Properties and Methods

### A.1 Alignment Property

Applies to: SIMATIC Number

This property specifies the alignment of the number in the control.

Syntax:

*object*.**Alignment** [= *value*]

The Alignment property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the alignment.

The settings for *value* are:

Setting	Description
0 or Left	(default) The number is aligned to the left side of the field.
1 or Right	The number is aligned to the right side of the field.
2 or Center	The number is centered in the field.

### A.2 Appearance Property

Applies to: SIMATIC Number

If this property is set to 3D (1), the Appearance property draws controls with three-dimensional effects if the BorderStyle property is set to "Fixed Single" (1). If the property is set to Flat (0), a flat border will surround the controls rectangle.

---

**Note**

This property only has an effect if the BorderStyle property is set to "Fixed Single" (1).

---

Syntax:

*object*.Appearance [= *value*]

The Appearance property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A <i>value</i> or constant that determines the appearance of <i>object</i> .

The settings for *value* are:

Setting	Description
0 or Flat	Paints the controls and forms without visual effects.
1 or ThreeD	(default) Paints the controls with three-dimensional (3-D) effects.

## A.3 BackColor Property

Applies to: SIMATIC Number

This property returns or sets the background color of an object.

Syntax:

*object*.BackColor [= *value*]

The BackColor property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A <i>value</i> or constant that determines the background color of an object.

The settings for *value* are:

Setting	Description
Standard Colors	Colors specified by using the RGB Color palette
Windows System Colors	Colors specified by system color constants (depending on the container); for example, colors listed in the Visual Basic (VB) object library in the Object Browser

## A.4 BorderStyle Property

Applies to: SIMATIC Number

If the property has the value "Fixed Single" (1), the control is surrounded by a rectangular border. If the property has the value None (0), no border will be displayed.

---

### Note

This property determines whether the Appearance property has any effect.

---

Syntax:

*object*.BorderStyle [= *value*]

The BorderStyle property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the border style.

The settings for *value* are:

Setting	Description
0 or None	(default) No border or border-related elements
1 or FixedSingle	A fixed, single-line border

## A.5 Caption1 and Caption2 Properties

Applies to: SIMATIC Number

These properties specify the displayed text of the first (Caption1) and second (Caption2) labels.

Syntax:

```
object.Caption1 [= value]
```

```
object.Caption2 [= value]
```

The Caption1 and Caption2 properties have these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A String value that specifies the text of the label.

## A.6 Caption1Alignment and Caption2Alignment Properties

Applies to: SIMATIC Number

These properties specify the alignment of the first (Caption1Alignment) or second (Caption2Alignment) labels.

Syntax:

```
object.Caption1Alignment [= value]
```

```
object.Caption2Alignment [= value]
```

The Caption1Alignment and Caption2Alignment properties have these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the alignment of the label.

The settings for *value* are:

Setting	Description
0 or Left	(Default for the first label) The label is aligned to the left side of the field.
1 or Right	(Default for the second label) The label is aligned to the right side of the field.
2 or Center	The label is centered in the field.

## A.7 Caption1BackColor, Caption2BackColor Properties

Applies to: SIMATIC Number

These properties return or set the background colors of the labels. The first label is determined by the Caption1BackColor property, and the second label is determined by the Caption2BackColor property.

Syntax:

`object.Caption1BackColor [= value]`

`object.Caption2BackColor [= value]`

The Caption1BackColor and Caption2BackColor properties have these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the background color of <i>object</i> .

The settings for *value* are:

Setting	Description
Standard Colors	Colors specified by using the RGB Color palette
Windows System Colors	Colors specified by system color constants (depending on the container); for example, colors listed in the Visual Basic (VB) object library in the Object Browser

## A.8 Caption1Font and Caption2Font Properties

Applies to: SIMATIC Number

This property returns a Font object for the font of the first (Caption1Font) or second (Caption2Font) label.

Syntax:

`object.Caption1Font`

`object.Caption2Font`

The Caption1Font and Caption2Font properties have these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.

## A.9 Caption1ForeColor, and Caption2ForeColor Properties

Applies to: SIMATIC Number

These properties return or set the foreground colors of the labels. The first label is determined by the Caption1ForeColor property, and the second label is determined by the Caption2ForeColor property.

Syntax:

```
object.Caption1ForeColor [= value]
```

```
object.Caption2ForeColor [= value]
```

The Caption1ForeColor and Caption2ForeColor properties have these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the background or foreground colors of <i>object</i> .

The settings for *value* are:

Setting	Description
Standard Colors	Colors specified by using the RGB Color palette
Windows System Colors	Colors specified by system color constants (depending on the container); for example, colors listed in the Visual Basic (VB) object library in the Object Browser

## A.10 Caption1Size and Caption2Size Properties

Applies to: SIMATIC Number

If the layout type is horizontal, then you can specify a width (in twips) of the first (Caption1Size) and second (Caption2Size) labels. If the layout type is vertical, then it specifies the height (in twips) of the labels. If you do not specify a size (if you accept the default value of -1), then the size will be calculated by the space required for the text.

Syntax:

```
object.Caption1Size [= value]
```

```
object.Caption2Size [= value]
```

The Caption1Size and Caption2Size properties have these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that allows the user is to determine a specific size for a label. The default value (-1) specifies that the size be determined by the length of the text for the label.

## A.11 ConvertedValue Property

Applies to: SIMATIC Number

This read-only property holds the value from the control engine (after data typing and range transformation have been performed). Use the following descriptions to understand the differences between the Value, ConvertedValue and Text properties:

- The Value property is a Variant. It must be connected, by way of the SIMATIC Data Control, to a value in the control engine. It can hold values of different data types.
- The ConvertedValue property is also a Variant and contains the value from the Value property, but it is already scaled and converted to the specified data type (DataType property).
- The Text property contains a String with the displayed String, which means, inclusively, radix conversion (Radix property), zero padding (ZeroPad property), and precision (Precision property) usage.

If the Enabled property is False, the bi-directional connection between Value and ConvertedValue is broken and changes to the Text or ConvertedValue property will not be written to the Value property.

Syntax:

***object*.ConvertedValue** [= *value*]

The ConvertedValue property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A Variant that specifies the value of the control.

## A.12 DataType Property

Applies to: SIMATIC Number

This property defines the storage type used for converted values. If you are using a data type for displaying a value which is too large, the value will be truncated. The data type specified in the SIMATIC Number Control must match any data type specified in the “Assigned Variable” field of the SIMATIC Data Control.

---

### Note

This property determines whether the Precision property has any effect.

---

Syntax:

*object*.**DataType** [= *value*]

The DataType property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the data type. Table A-1 lists the data types and settings for <i>value</i> .

Table A-1 Settings for the Data Types

Data Type	Setting	Size	Description
Boolean	0	1 bit	Single bit value
Byte	1	1 byte	Unsigned single-byte value
Word	2	2 bytes	Unsigned two-byte value
Integer	3	2 bytes	Signed two-byte integer value
Double Word	4	4 bytes	Unsigned four-byte value (default)
Double Integer	5	4 bytes	Signed four-byte integer value
Real	6	4 bytes	Signed four-byte real (floating-point) value
Timer	7	2 bytes	Unsigned two-byte value
Counter	8	2 bytes	Unsigned two-byte value



**Note**

If the data size configured to be accessed in the control engine is larger than the data being displayed in the SIMATIC control and the value of the data from the control engine is larger than can be displayed by the data type, the value is displayed with “...” preceding it. For example, you can assign a double-word variable, such as MD100, in the SIMATIC Data Control and select the data type “Byte” in the SIMATIC Number Control. If the value in the control engine exceeds 255 (the maximum value for a byte), “...” precedes the value. Before the value can be changed from the SIMATIC Number Control, the “...” preceding the value must be deleted.

**A.13 Direction Property**

Applies to: SIMATIC Slider

The Direction property sets the orientation (horizontal or vertical) of the SIMATIC control.

Syntax:

*object.Direction* [= *value*]

The Direction property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the orientation.

The settings for *value* are:

Setting	Description
0	Horizontal
1	Vertical

**A.14 Enabled Property**

Applies to: SIMATIC Button, SIMATIC Number, SIMATIC Slider

When this property is True, the control reacts on changes of the Value property and fires events. If this property is False, then the control is disabled and does not react on changes in the Value property and does not fire any event (except the error event).

Syntax:

`object.Enabled [= value]`

The Enabled property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A Boolean expression that specifies whether <i>object</i> can respond to user-generated events.

The settings for *value* are:

Setting	Description
True	(default) Allows the object to respond to events
False	Prevents object from responding to events

## A.15 Factor Property

Applies to: SIMATIC Number

The Factor and Offset properties specify the scaling factor and the offset used when the scale-by-formula option has been enabled.

---

### Note

The ScaleMode property must be set to "ByFormula" (1) for the Factor and Offset properties to have any effect.

---

You can use a formula to scale the value. In the following formula, "PLC Value" is similar to the contents of the Value property if the control is connected to the control engine; "Scale" is the value of the Factor property; "Offset" is the value of the Offset property; and "Displayed value" is also the contents of the Text property.

$\text{PLC Value} * \text{Factor} + \text{Offset} = \text{Displayed value}$

Syntax:

`object.Factor [= value]`

The Factor property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A floating-point value that defines the factor for the scaling formula.

---

#### Note

The default value of the factor is 1.0, and the default value of the offset is 0.0.

---

## A.16 FalseCaption Property

Applies to: SIMATIC Button

This property determines the text that is displayed in the control when the Value property is False (equal to 0, or "Off").

Syntax:

*object.FalseCaption* [= *value*]

The FalseCaption property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A String value that specifies the text of the label.

## A.17 FalseColor Property

Applies to: SIMATIC Button

This property determines the color of the control when the Value property is False (equal to 0, or "Off").

Syntax:

*object.FalseColor* [= *value*]

The FalseColor property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the background or foreground colors of an object.

The settings for *value* are:

Setting	Description
Standard Colors	Colors specified by using the RGB Color palette
Windows System Colors	Colors specified by system color constants (depending on the container); for example, colors listed in the Visual Basic (VB) object library in the Object Browser

## A.18 Font Property

Applies to: SIMATIC Button, SIMATIC Number

This property returns a Font object for the main font of the control.

Syntax:

*object*.Font

The Font property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.

## A.19 ForeColor Property

Applies to: SIMATIC Number

The ForeColor property returns or sets the foreground color used to display text and graphics in an object.

Syntax:

*object*.ForeColor [= *value*]

The ForeColor property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the foreground colors of <i>object</i> .

The settings for *value* are:

Setting	Description
Standard Colors	Colors specified by using the RGB Color palette
Windows System Colors	Colors specified by system color constants (depending on the container); for example, colors listed in the Visual Basic (VB) object library in the Object Browser

## A.20 LargeChange Property

Applies to: SIMATIC Slider

The LargeChange property determines how far the slider indicator moves when the control has focus and you press the Page Up or Page Down key. The Value property is increased by LargeChange if you press the Page Up key or click to the right/top of the indicator. It is decreased by LargeChange if you press the Page Down key or click to the left/bottom of the indicator.

Syntax:

*object*.LargeChange [= *value*]

The LargeChange property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the amount of change.

## A.21 LayoutType Property

Applies to: SIMATIC Number

This property specifies how the labels are arranged around the display area of the control. When the layout type is Horizontal (0), caption 1 is on the left of the display area, and caption 2 on the right side of the display area. When the layout type Vertical (1), caption 1 is located above the display area, and caption 2 is below the display area.

Syntax: `object.LayoutType [= value]`

The LayoutType property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the layout type.

The settings for *value* are:

Setting	Description
Horizontal (0)	(default) Arranges the labels to the left and right of the display area of the control
Vertical (1)	Arranges the labels above and below the display area of the control

## A.22 LimitCheck Property

Applies to: SIMATIC Number

This property determines whether to check that a value entered by the control is within specified limits and to define the range of values that determine the limits. A value changed by the control engine is not checked against the defined range.

You can specify a range of valid values for any entry. (The lower limit equals the smallest number allowed, and the upper limit equals the largest number allowed). If an invalid value (a value which is outside of the range) is entered, the control does not write that value to the Value property.

If the limit-check fails, the Error event and the LimitCheckFailed events will be fired. The Error event will be fired only if the ShowErrorBoxes property is enabled (selected).

**Note**

This Limit Check property must be enabled for the UpperLimit and LowerLimit properties to have any effect.

Syntax:

*object.LimitCheck* [= *value*]

The LimitCheck property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the kind of limit check.

The settings for *value* are:

Setting	Description
CheckNone (0)	(default) Does not check the entry against limits
Lower (1)	Checks the value against the lower limit only
Upper (2)	Checks the value against the upper limit only
Both (3)	Checks the value against both the upper and lower limits

## A.23 Locked Property

Applies to: SIMATIC Number

If the control is locked it is in a read-only state. The user is unable to change any values, but the current value is nevertheless shown. By default the control is not in locked mode, so the user can enter numbers.

Syntax:

*object.Locked* [= *value*]

The Locked property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A Boolean expression that specifies whether <i>object</i> can be edited.

The settings for *value* are:

Setting	Description
True	You can scroll and highlight the text in the control, but you cannot edit it. The program can still modify the text by changing the Value property. Changes to the Value property are reflected: the control displays the values in the control engine, but you are unable to change them.
False	(default) You can edit the text in the control.

## A.24 LowerLimit Property

Applies to: SIMATIC Number

These properties allow access to the lower limit value. If the value entered by the user is lower than the LowerLimit value or higher than the UpperLimit value, it will not be written to the control engine.

You can specify a range of valid values for any entry. (The lower limit equals the smallest number allowed.) If an invalid value (a value which is outside of the range) is entered, the control does not write that value to the Value property.

If the limit-check fails, the Error event and the LimitCheckFailed events will be generated. The Error event will be generated only if the ShowErrorBoxes property is enabled (selected).

---

### Note

The LimitCheck property must be set to "Lower" (1) or "Both" (3) before the LowerLimit property can have an effect.

The LimitCheck property must be set to "Upper" (2) or "Both" (3) before the UpperLimit property can have an effect.

---

Syntax:

`object.LowerLimit [= value]`

The LowerLimit property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A floating-point value that defines the lower or upper limit used at limit checking.



---

**Note**

The default value for the LowerLimit is 0.0, and the default value for the UpperLimit is 100.0.

---

## A.25 Max Property

Applies to: SIMATIC Slider

The Max property determines the position of the slider indicator of the SIMATIC control. When the Value property is greater than or equal to Max, the indicator is in its rightmost (or top) position.

Syntax:

*object*.**Max** [= *value*]

The Max property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the maximum number displayed.

## A.26 Min Property

Applies to: SIMATIC Slider

The Min property determines the position of the slider indicator of the SIMATIC control. When the Value property is less than or equal to Min, the indicator is in its leftmost (or bottom) position.

Syntax:

*object*.**Min** [= *value*]

The Min property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the minimum number displayed.

## A.27 Offset Property

Applies to: SIMATIC Number

The Factor and Offset properties specify the scaling factor and the offset used when the scale-by-formula option has been enabled.

---

### Note

The ScaleMode property must be set to "ByFormula" (1) for the Factor and Offset properties to have any effect.

---

You can use a formula to scale the value. In the following formula, "PLC Value" is similar to the contents of the Value property if the control is connected to the control engine; "Scale" is the value of the Factor property; "Offset" is the value of the Offset property; and "Displayed value" is also the contents of the Text property.

$$\text{PLC Value} * \text{Factor} + \text{Offset} = \text{Displayed value}$$

Syntax:

`object.Offset [= value]`

The Offset property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A floating-point value that defines the factor or the offset for the scaling formula.

---

### Note

The default value of the factor is 1.0, and the default value of the offset is 0.0.

---

## A.28 Precision Property

Applies to: SIMATIC Number

This property is available if the DataType is set to "Real" (6) (data type with precision). In that case you can change the precision (number of digits behind the decimal point) of the number. The number will be rounded at the specified precision.

**Note**

The DataType property must be set to "Real" (6) before this property can have an effect.

Syntax:

*object.Precision* [= *value*]

The Precision property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	An integer value that defines the precision of the number. The default precision is 3.

## A.29 Radix Property

Applies to: SIMATIC Number

This property defines the radix (binary, octal, decimal, or hexadecimal) for the number to be displayed. Any value entered must also be in the radix defined by this property.

Syntax:

*object.Radix* [= *value*]

The Radix property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the radix used for displaying the value.

The settings for *value* are:

Setting	Description
Binary (0)	Binary mode (valid characters for input: 0 and 1)
Octal (1)	Octal mode (valid characters for input: 0 to 7)
Decimal (2)	(default) Decimal mode (valid characters for input: 0 to 9, '+', '-', and '.')
Hexadecimal (3)	Hexadecimal mode (valid characters for input: 0 to 9, a to f, and A to F)

### A.30 RangeRawMax, RangeRawMin, RangeScaledMax, and RangeScaledMin Properties

Applies to: SIMATIC Number

These properties define the ranges for scaling a value by a “range transformation”:

- RangeRawMax specifies the upper value of the source range.
- RangeRawMin specifies the lower value of the source range.
- RangeScaledMax specifies the upper value of the destination range.
- RangeScaledMin specifies the lower value of the destination range.

#### Note

The ScaleMode property must be set to “ByRange” (2) before these properties can have an effect.

When you use a range transformation to scale the value, you specify a source range (for the values in the control engine) and a destination range (for the values that are displayed by the control). The values of one range will be transformed to the other range. See Figure A-1. The source and destination ranges define a ratio for the transformation; they do not define upper or lower limits. A value can be larger or smaller than the range; the transformation will use the two ranges to extrapolate the other value.

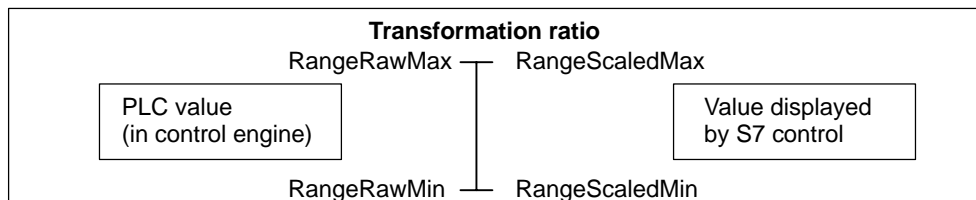


Figure A-1 Scaling Values by Range Transformation

Syntax:

```
object.RangeRawMax [= value]
object.RangeRawMin [= value]
object.RangeScaledMax [= value]
object.RangeScaledMin [= value]
```

The RangeRawMin, RangeRawMax, RangeScaleMin, RangeScaleMax properties have these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A floating-point value that defines the range boundaries for the range mode scaling.

---

#### Note

The default value of the RangeRawMax is 255.0; the default value of the RangeRawMin is 0.0; the default value of the RangeScaledMax is 100.0; and the default value of the RangeScaledMin is 0.0.

---

## A.31 ScaleMode Property

Applies to: SIMATIC Number

This property specifies the scaling mode to be used for scaling values. The values can also be specified at the Scale Property Tab. There are two choices for scaling mode:

- By formula:  $\text{PLC Value} * \text{Factor} + \text{Offset} = \text{Displayed value}$   
 where: PLC Value is similar to the contents of the Value property if the control is connected to the control engine; Scale is the value of the Factor property; Offset is the value of the Offset property; and Displayed value is the contents of the Text property.
- By range transformation: you specify a source range (of PLC values) and a destination range (of displayed values), and the values of the one range are transformed to the other range. See Figure A-1 and the Range properties.

---

#### Note

The Scale Mode property determines whether the RangeRawMax, RangeRawMin, RangeScaledMax, RangeScaledMin, Factor, and Offset properties have any effect.

---

Syntax:

*object*.ScaleMode [= value]

The ScaleMode property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the kind of scaling.

The settings for *value* are:

Setting	Description
ScaleNone (0)	(default) No scaling
ByFormula (1)	Use the formula containing the factor and offset to scale the value
ByRange (2)	Use the range transformation method to scale the value

## A.32 ShowButtons Property

Applies to: SIMATIC Number

This property displays the Enter and Cancel buttons for the control. When this property is enabled (True), any new value entered with the control must be confirmed before it is written to the SIMATIC Data Control. Typing a new value in the display field of the control enables the buttons. Clicking on the Enter button writes the new value to the control engine; clicking on the Cancel button replaces the new value with the value from the control engine. The default setting for this option is False (disabled).

Syntax:

*object*.ShowButtons [= value]

The ShowButtons property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A Boolean expression that specifies whether or not to display buttons.

The settings for *value* are:

Setting	Description
True	Two buttons will be shown on the right side of the display field. When you enter a new value in the display field, the buttons become active. To confirm the change and write this value to the Value property (and to the control engine), click on the Enter button; to cancel the value, click on the Cancel button.
False	(default) The buttons are hidden.

### A.33 ShowErrorBoxes Property

Applies to: SIMATIC Number, Panel

This property specifies whether to display the default error boxes when there is a user-generated error. Every time an error occurs, an Error event will be generated. If the ShowErrorBoxes property is enabled (selected), a default error message box will be displayed.

All errors on connections are reported by the Connection Error event.

---

#### Note

Computing provides error messages in English only. If you want to display messages in other languages, you must disable (deselect) the ShowErrorBoxes option and write program code to react on the error event.

---

Syntax:

*object*.ShowErrorBoxes [= *value*]

The ShowErrorBoxes property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A Boolean expression that specifies whether the control displays error boxes.

The settings for *value* are:

Setting	Description
True	(default) The control shows the default error boxes.
False	The error boxes are hidden.

### A.34 SmallChange Property

Applies to: SIMATIC Slider

This property determines how far the slider indicator moves when the control has focus and you press the up/down or right/left arrow keys. The Value property is increased by SmallChange if you press the right (or up) arrow key. It is decreased by SmallChange if you press the left (or down) arrow key.

Syntax:

*object*.SmallChange [= *value*]

The SmallChange property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the amount of change

### A.35 TagServerName Property

Applies to: SIMATIC Number

This property determines the name of the SIMATIC Data Control which shall be used for a connection to the control engine. The TagServerName property applies to the SIMATIC Number Control, but it is created automatically by the SIMATIC Data Control. It can also be created by your program code.

Syntax:

*object*.TagServerName [= *value*]

The TagServerName property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A String value that specifies the name of an available SIMATIC Data Control.

### A.36 Ticks Property

Applies to: SIMATIC Slider

This property sets the number of ticks, or unit markers, of the control. For example, if Ticks = 10, the scale of the control will be divided into 10 sections.



Syntax:

*object.Ticks* [= *value*]

The Ticks property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A constant that determines the number of unit markers to be displayed.

### A.37 TrueCaption Property

Applies to: SIMATIC Button

This property determines the text that is displayed in the control when the Value property is True (equal to 1, or “On”).

Syntax:

*object.TrueCaption* [= *value*]

The TrueCaption property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A String value that specifies the text of the label.

### A.38 TrueColor Property

Applies to: SIMATIC Button

This property determines the color of the control when the Value property is True (equal to 1, or “On”).

Syntax:

*object.TrueColor* [= *value*]

The TrueColor property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant that determines the background or foreground colors of <i>object</i> .

The settings for *value* are:

Setting	Description
Standard Colors	Colors specified by using the RGB Color palette
Windows System Colors	Colors specified by system color constants (depending on the container); for example, colors listed in the Visual Basic (VB) object library in the Object Browser

## A.39 UpperLimit Property

Applies to: SIMATIC Number

This property allows access to the upper limit value. If the value entered by the user is higher than the UpperLimit value, it will not be written to the control engine.

You can specify a range of valid values for any entry. (The upper limit equals the largest number allowed.) If an invalid value (a value which is outside of the range) is entered, the control does not write that value to the Value property.

If the limit-check fails, the Error event and the LimitCheckFailed events will be generated. The Error event will be generated only if the ShowErrorBoxes property is enabled (selected).

---

### Note

The LimitCheck property must be set to "Lower" (1) or "Both" (3) before the LowerLimit property can have an effect.

The LimitCheck property must be set to "Upper" (2) or "Both" (3) before the UpperLimit property can have an effect.

---

Syntax:

`object.UpperLimit [= value]`

The UpperLimit property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A floating-point value that defines the lower or upper limit used at limit checking.

---

**Note**

The default value for the LowerLimit is 0.0, and the default value for the UpperLimit is 100.0.

---

## A.40 Value Property

Applies to: SIMATIC Button, SIMATIC Number, SIMATIC Slider

This property should be linked, using the SIMATIC Data Control, to a value in the control engine. It is bindable. Use the following descriptions to understand the differences between the Value, ConvertedValue, and Text properties:

- The Value property is a Variant. It must be connected, by way of the SIMATIC Data Control, to a value in the control engine. It can hold values of different data types.
- The ConvertedValue property is also a Variant and contains the value from the Value property, but it is already scaled and converted to the specified data type (DataType property).
- The Text property contains a String with the displayed String, which means, inclusively, radix conversion (Radix property), zero padding (ZeroPad property), and precision (Precision property) usage.

If the Enabled property is False, the bi-directional connection between Value and ConvertedValue is broken. The result is that changes to the Text or ConvertedValue property will not be written to the Value property.

---

**Note**

If the value of the Value property changes, the Change event will be generated.

---

Syntax:

*object.Value* [= *value*]

The Value property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A Variant that specifies the value of the control.

## A.41 Write Method

Applies to: SIMATIC Number

This method writes the value of the ConvertedValue property to the Value property. You must use this method only if the WriteMode is set to Manual (1).

Syntax:

*object*.Write

The Write method has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.

## A.42 WriteMode Property

Applies to: SIMATIC Number

This property determines how the control responds when the user enters a new value. If the write mode is set to Automatic (0), the value (if valid) is written automatically into the Value property (and to the control engine). If the write mode is Manual (1), the value is not written to the value property unless your program code calls the method "Write" at the control.

Syntax:

*object*.WriteMode [= value]

The WriteMode property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A value or constant expression that specifies whether the control automatically passes entered values to the Value property.

The settings for *value* are:

Setting	Description
Automatic (0)	(default) Automatically passes the new (input) value to the Value property
Manual (1)	Does not write the new (input) value unless the control processes a Write method

### A.43 ZeroPad Property

Applies to: SIMATIC Number

This property determines whether the number displayed by the control is padded with zeros (to the left of the value) to the size of the data type.

Syntax:

`object.ZeroPad [= value]`

The ZeroPad property has these parts:

Part	Description
<i>object</i>	An object expression (identifier for the specific SIMATIC control) that evaluates to an object in the Applies To list.
<i>value</i>	A Boolean expression that specifies whether or not the displayed number is filled with leading zeros.

The settings for *value* are:

Part	Description
True	Fills the number with leading zeros to the size specified by the DataType property.
False	(default) Does not fill the number with leading zeros.



## Events of the SIMATIC Controls

### B.1 Change Event

Applies to: SIMATIC Button, SIMATIC Number, SIMATIC Slider

This event occurs when the value of the Value property changes. Either the control engine or the S7 control object can change the value in the Value property.

Syntax: **Change**( )

### B.2 Click Event

Applies to: SIMATIC Button, SIMATIC Number, SIMATIC Slider

This event occurs when a mouse button is pressed and released while the mouse cursor is over the control.

Syntax: **Click**( )

---

#### Note

To distinguish between the left, right, and middle mouse buttons, use the MouseDown and MouseUp events. If there is code in the Click event, the DbClick event will never trigger, because the Click event is the first event to trigger between the two. As a result, the mouse click is intercepted by the Click event, so the DbClick event does not occur.

---

### B.3 DbClick Event

Applies to: SIMATIC Number, SIMATIC Slider

This event occurs when a mouse button is double-clicked while the cursor is over the control.

Syntax: **DbClick**( )

---

**Note**

To distinguish between the left, right, and middle mouse buttons, use the MouseDown and MouseUp events.

If there is code in the Click event, the DblClick event will never trigger, because the Click event is the first event to trigger between the two. As a result, the mouse click is intercepted by the Click event, so the DblClick event does not occur.

---

## B.4 Error Event

Applies to: SIMATIC Number

This event occurs when the control encounters an error.

Syntax:

```
Error(long SCode, BSTR lpzDescription, BSTR lpzHelpFileName, _  
long nHelpId)
```

The Error event has these parts:

Part	Description
<i>SCode</i>	See Table B-1
<i>lpzDescription</i>	String with a description of the error condition
<i>lpzHelpFileName</i>	Name of the Help file in which the error is described
<i>nHelpId</i>	Help topic ID with a description of the error

Table B-1 SCodes (Error Event Codes)

Value	Description
0xC0040002	The scaling cannot proceed because of an error in the formula used.
0xC0040003	The set value at the Value property is invalid.
0xC0040004	The set value at the Text property is invalid.
0xC0040005	The other OLE components could not be found.
0xC0040006	The Microsoft standard controls could not be created.
0xC0040010	The limit check cannot proceed because the lower limit is greater than the upper limit. This error will only occur if the LimitCheck property is set properly.



## B.5 KeyDown Event

Applies to: SIMATIC Number, SIMATIC Slider

This event occurs when the user presses a key while the control has the focus. See also the KeyUp Event.

Syntax: **KeyDown(long KeyID, long Shift)**

The KeyDown event has these parts:

Part	Description
<i>KeyID</i>	Key code, such as vbKeyF1 (the F1 key) or vbKeyHome (the HOME key) To specify key codes, use the constants in the Visual Basic (VB) object library in the Object Browser.
<i>Shift</i>	An integer that corresponds to the state of the SHIFT, CTRL, and ALT keys at the time of the event The shift argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys have been pressed. For example, if both CTRL and ALT are pressed, the value of shift is 6.

Use the KeyDown and KeyUp event procedures if you need to respond to both the pressing and releasing of a key.

KeyDown and KeyUp interpret the uppercase and lowercase of each character by means of two arguments: keycode, which indicates the physical key (thus returning A and a as the same key) and shift, which indicates the state of shift+key and therefore returns either "A" or "a".

If you need to test for the shift argument, you can use the shift constants that define the bits within the argument. The constants have the following values:

- vbShiftMask (1): SHIFT key bit mask
- vbCtrlMask (2): CTRL key bit mask
- vbAltMask (4): ALT key bit mask

The constants act as bit masks that you can use to test for any combination of keys.

You test for a condition by first assigning each result to a temporary integer variable and then comparing Shift to a bit mask. Use the And operator with the Shift argument to test whether the condition is greater than 0, indicating that the modifier was pressed.

## B.6 KeyPress Event

Applies to: SIMATIC Number, SIMATIC Slider

This event occurs when an ANSI key is pressed and released while the control has the focus.

Syntax: **KeyPress (long keyAscii)**

The KeyPress event has these parts:

Part	Description
<i>keyAscii</i>	ASCII key code of the pressed key, such as vbKeyF1 (the F1 key) or vbKeyHome (the HOME key)

## B.7 KeyUp Event

Applies to: SIMATIC Number, SIMATIC Slider

This event occurs when a key is released while the control has the focus.

Syntax: **KeyUp(long KeyID, long Shift)**

The KeyDown event has these parts:

Part	Description
<i>KeyID</i>	Key code, such as vbKeyF1 (the F1 key) or vbKeyHome (the HOME key)  To specify key codes, use the constants in the Visual Basic (VB) object library in the Object Browser.
<i>Shift</i>	An integer that corresponds to the state of the SHIFT, CTRL, and ALT keys at the time of the event  The shift argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys have been pressed. For example, if both CTRL and ALT are pressed, the value of shift is 6.

Use KeyDown and KeyUp event procedures if you need to respond to both the pressing and releasing of a key.

KeyDown and KeyUp interpret the uppercase and lowercase of each character by means of two arguments: keycode, which indicates the physical key (thus returning A and a as the same key) and shift, which indicates the state of shift+key and therefore returns either "A" or "a".

If you need to test for the shift argument, you can use the shift constants which define the bits within the argument. The constants have the following values:

- vbShiftMask (1): SHIFT key bit mask
- vbCtrlMask (2): CTRL key bit mask
- vbAltMask (4): ALT key bit mask

The constants act as bit masks that you can use to test for any combination of keys.

You test for a condition by first assigning each result to a temporary integer variable and then comparing Shift to a bit mask. Use the And operator with the Shift argument to test whether the condition is greater than 0, indicating that the modifier was pressed.

## B.8 LimitCheckFailed Event

Applies to: SIMATIC Number

This event occurs when the limit check fails.

Syntax: **LimitCheckFailed**(long *IViolationInformation*)

The LimitCheckFailed event has these parts:

Part	Description
<i>IViolationInformation</i>	Describes the condition that caused the limit check to fail:

The settings for *IViolationInformation* are:

Setting	Description
1	Value is too small
2	Value is too large

## B.9 MouseDown Event

Applies to: SIMATIC Number, SIMATIC Slider

This event occurs when a mouse button is pressed while the mouse cursor is over the control.

Syntax:

```
MouseDown(short Button, short Shift, OLE_XPOS_PIXELS x, OLE_YPOS_PIXELS y)
```

The MouseDown event has these parts:

Part	Description
<i>Button</i>	<p>An integer that identifies the button that was pressed to cause the event</p> <p>The button argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that caused the event.</p>
<i>Shift</i>	<p>An integer that corresponds to the state of the SHIFT, CTRL, and ALT keys when the button specified in the button argument is pressed or released</p> <p>A bit is set if the key is down. The shift argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The shift argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of shift would be 6.</p>
<i>x,y</i>	<p>returns a number that specifies the current location of the mouse pointer</p>

## B.10 MouseMove Event

Applies to: SIMATIC Number, SIMATIC Slider

This event occurs when the mouse cursor moves over the control.

Syntax:

```
MouseMove(short Button, short Shift, OLE_XPOS_PIXELS x, _
OLE_YPOS_PIXELS y)
```

The MouseMove event has these parts:

Part	Description
<i>Button</i>	<p>An integer that identifies the button that was pressed to cause the event</p> <p>The button argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that caused the event.</p>
<i>Shift</i>	<p>An integer that corresponds to the state of the SHIFT, CTRL, and ALT keys when the button specified in the button argument is pressed or released</p> <p>A bit is set if the key is down. The shift argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The shift argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of shift would be 6.</p>
<i>x,y</i>	returns a number that specifies the current location of the mouse pointer

## B.11 MouseUp Event

Applies to: SIMATIC Number, SIMATIC Slider

This event occurs when a mouse button is released while the mouse cursor is over the control.

Syntax:

```
MouseUp(short Button, short Shift, OLE_XPOS_PIXELS x, _  
OLE_YPOS_PIXELS y)
```

The MouseUp event has these parts:

Part	Description
<i>Button</i>	<p>An integer that identifies the button that was pressed to cause the event</p> <p>The button argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that caused the event.</p>
<i>Shift</i>	<p>An integer that corresponds to the state of the SHIFT, CTRL, and ALT keys when the button specified in the button argument is pressed or released</p> <p>A bit is set if the key is down. The shift argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The shift argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of shift would be 6.</p>
<i>x,y</i>	<p>returns a number that specifies the current location of the mouse pointer</p>

# Guidelines for Programming with Computing

# C

## Chapter Overview

When you write programs that use the SIMATIC controls provided by the Computing software to access the control engine, be aware of the programming guidelines, especially those in regard to the use of breakpoints and for using timers in your code.

The Computing software provides a container (SoftContainer) for the SIMATIC controls and other ActiveX controls. You can also use other containers, such as Visual Basic, with the SIMATIC Controls. In order to use the SIMATIC controls in another container, the container must support the extended controls. If the container does not support these functions, you must supply program code to perform these functions.

Section	Description	Page
C.1	Guidelines for Third-Party Containers	C-2
C.2	Programming Guidelines	C-3

## C.1 Guidelines for Third-Party Containers

For the SIMATIC Data control to work within a third-party container, the container must support the “property browsing” functions of the Data control. To do this, the container must support the functions for “extended controls” (as defined by Microsoft for containers). An extended control is a partial control that wraps around another control to support container-specific properties, methods and events. (Refer to Microsoft’s on-line documentation for more information about containers and extended controls.)

To provide the extended control functions, the container must support the following methods:

- IOleClientSize::GetContainer
- IOleContainer::EnumObjects
- IOleControlSite::GetExtendedControl

The extended control of the container must also support a Name property.

The SoftContainer provided with the Computing software supports extended controls, as does Microsoft’s Visual Basic. Containers from other vendors (such as Borland’s Delphi version 3.0) do not support extended controls. The Siemens customer support center can help determine if your container supports the extended control functions.

If your container does not support the extended control functions, you must provide program code to perform these functions. Contact the Siemens customer support center for sample code that performs the extended control functions.



## C.2 Programming Guidelines

The following guidelines relate specifically to Visual Basic; however, they can also apply to other programming languages.



---

### Caution

Using the timer function improperly or using breakpoints with your subroutines that access Computing can cause problems that could potentially cause your computer or application to crash or lock up. Depending on the configuration, this could cause the application to lose communication with the control engine. This could cause unpredictable process operation, which could result in death or serious injury to personnel, and/or damage to equipment.

Always install a physical emergency stop circuit for your machine or process.

---

### Using Timers in Your Program

The Timer function in Visual Basic version 5 allows a timer to interrupt code in progress within the same thread, which can cause problems with potentially serious consequences. If you use VB timers with Computing, observe the following guidelines:

- Always kill (disable) the timers in the Form\_Unload subroutine. Otherwise, a timer can trigger an event while the VB program is shutting down; this condition could cause your computer or your application to crash, lock up, or to continue running invisibly.
- If you start your timer in the Form\_Load subroutine, the timer event could occur before the other objects have finished being instantiated. In order to ensure that the objects have been properly instantiated, always start a timer in the Form\_Load subroutine with a large interval (such as 1 or 2 seconds) to allow the objects to be properly instantiated. Subsequent timer intervals can be set to shorter intervals.

### Using Breakpoints in Your Program

Setting a breakpoint while calling a Computing function will halt all communication by Computing with the server (the WinLC control engine). Do not use a breakpoint with a WinLC that is controlling a process or machine.

### Using a Separate Data Control to Access Critical Data

The performance of your program can be improved by using a separate SIMATIC Data control to access frequently changing, critical data.

## Disconnecting from the Control Engine

If your subroutine accesses the Data control programmatically, always disconnect from the control engine (using a Disconnect method) in the Form\_Unload subroutine.

In addition, disable the AutoConnect property for the Data control if you explicitly call the Connect method within your program. This helps to ensure that the Data control does not connect unexpectedly to the control engine.

## Determining the Order of AutoConnects for Multiple Data Controls

If you use multiple Data controls in your program, the order in which the different Data controls automatically connect to the control engine(s) cannot be determined. If the order in which the Data controls connect to the control engine(s) is critical, disable the AutoConnect property for the Data control and use the Connect and Disconnect methods for the individual Data controls.

## Using the Visual Basic String Function (Right) in a Form with a Number Control

Both Visual Basic and the SIMATIC Number control use the keyword "Right":

- As described in Section A.1, the SIMATIC Number control uses an alignment constant to determine the appearance (for example, to align the number to the right edge of the field). The buttons that select these alignment constants appear on the "General" tab of the "Properties" dialog box for the Number control.
- Visual Basic provides a string function Right that returns a specified number of characters from a string.

When you insert a Number control onto a Visual Basic form, the Right alignment constant of the Number control overrides the Right string function of Visual Basic. In order to use the Right string function of Visual Basic, you must explicitly identify the Visual basic Right function by entering: **VBA.Right**

For example, use the following code to access the right-most character ("c") in a string of the characters "abc":

```
MyString = VBA.Right("abc", 1)
```

# Index

## A

- Accessing data
  - connect/disconnect, C-4
  - separate Data control, C-3
- Accessing memory areas (S7), ActiveX controls
  - Button, 1-1–1-5
  - Number, 2-1–2-12
  - Slider, 3-1–3-3
- Accessing process data, ActiveX controls
  - Button, 1-1–1-5
  - Number, 2-1–2-12
  - Slider, 3-1–3-3
- ActiveX controls
  - Button, 1-1–1-5
  - Button control
    - description, 1-1
    - toolbar button, 1-1
  - Number control
    - description, 2-1
    - toolbar button, 2-1
  - properties
    - Button control, 1-2–1-6
    - Number control, 2-2–2-13
    - Slider control, 3-2–3-4
  - Slider control
    - description, 3-1
    - toolbar button, 3-2
- Alignment property, A-1
- Appearance property, A-1

## B

- BackColor property, A-2
- BorderStyle property, A-3
- Breakpoints, C-3
- Button control
  - description, 1-1
  - events
    - Change, B-1
    - Click, B-1

- properties, 1-2–1-6
  - Enabled, A-9
  - FalseCaption, A-11
  - FalseColor, A-11
  - Font, A-12
  - TrueCaption, A-25
  - TrueColor, A-25
  - Value, A-27
- toolbar button, 1-1

## C

- CaptionBackColor properties, A-5
- CaptionFont property, A-5
- Caption properties, A-4
- CaptionAlignment properties, A-4
- CaptionForeColor properties, A-6
- CaptionSize property, A-6
- Change event, B-1
- Click event, B-1
- Computing
  - Button control
    - description, 1-1
    - properties, 1-2–1-6
    - toolbar button, 1-1
  - Number control
    - description, 2-1
    - properties, 2-2–2-13
  - Number control object, toolbar button, 2-1
  - properties
    - Button, 1-2–1-6
    - Number, 2-2–2-13
    - Slider, 3-2–3-4
  - SIMATIC controls
    - Button, 1-1–1-5
    - Number, 2-1–2-12
    - Slider, 3-1–3-3
  - Slider control
    - description, 3-1
    - properties, 3-2–3-4
    - toolbar button, 3-2

Control engine, SIMATIC controls  
  Button, 1-1–1-5  
  Number, 2-1–2-12  
  Slider, 3-1–3-3  
ConvertedValue property, A-7

## D

Data  
  ActiveX control objects  
    Button, 1-2–1-6  
    Number, 2-3–2-13  
    Slider, 3-2–3-4  
  SIMATIC controls  
    Button, 1-1–1-5  
    Number, 2-1–2-12  
    Slider, 3-1–3-3  
Data control, containers, C-2  
DataType property, A-8  
DbClick event, B-1  
Direction property, A-9

## E

Emergency stop circuit, 1-2, 2-3, 3-2  
Enabled property, A-9  
Error codes, Number control, 2-12  
Error event, B-2  
Events  
  Change, B-1  
  Click, B-1  
  DbClick, B-1  
  Error, B-2  
  KeyDown, B-3  
  KeyPress, B-4  
  KeyUp, B-4  
  LimitCheckFailed, B-5  
  MouseDown, B-6  
  MouseMove, B-7  
  MouseUp, B-8

## F

Factor property, A-10  
FalseCaption property, A-11  
FalseColor property, A-11  
Font property, A-12  
ForeColor property, A-12

## G

Guidelines  
  breakpoints in Visual Basic, C-3  
  connect/disconnect, C-4  
  containers, C-2  
  Data control for critical data, C-3  
  emergency stop circuit, 1-2, 2-3, 3-2  
  String functions (VB) when using a Number control, 2-5, C-4  
  using Visual Basic timers, C-3

## H

Height (properties), 2-7  
Horizontal labels, 2-7

## K

KeyDown event, B-3  
KeyPress event, B-4  
KeyUp event, B-4

## L

LargeChange property, A-13  
LayoutType property, A-14  
LimitCheck property, A-14  
LimitCheckFailed event, B-5  
Locked property, A-15  
LowerLimit property, A-16

## M

Max property, A-17  
Memory areas of S7 controllers, SIMATIC controls  
  Button, 1-1–1-5  
  Number, 2-1–2-12  
  Slider, 3-1–3-3  
Methods, Write method, A-28  
Min property, A-17  
Monitoring and modifying data, SIMATIC controls  
  Button, 1-1–1-5  
  Number, 2-1–2-12  
  Slider, 3-1–3-3  
MouseDown event, B-6

MouseMove event, B-7  
 MouseUp event, B-8

## N

Number control  
   description, 2-1  
   error codes, 2-12  
   events  
     Change, B-1  
     Click, B-1  
     DbClick, B-1  
     Error, B-2  
     KeyDown, B-3  
     KeyPress, B-4  
     KeyUp, B-4  
     LimitCheckFailed, B-5  
     MouseDown, B-6  
     MouseMove, B-7  
     MouseUp, B-8  
   properties, 2-2–2-13  
     Alignment, A-1  
     Appearance, A-1  
     BackColor, A-2  
     BorderStyle, A-3  
     Caption, A-4  
     CaptionAlignment, A-4  
     CaptionBackColor, A-5  
     CaptionFont, A-5  
     CaptionForeColor, A-6  
     CaptionSize, A-6  
     ConvertedValue, A-7  
     DataType, A-8  
     Enabled, A-9  
     Factor, A-10  
     Font, A-12  
     ForeColor, A-12  
     LayoutType, A-14  
     LimitCheck, A-14  
     Locked, A-15  
     LowerLimit, A-16  
     Offset, A-18  
     Precision, A-18  
     Radix, A-19  
     RangeRawMax, A-20  
     RangeRawMin, A-20  
     RangeScaledMax, A-20  
     RangeScaledMin, A-20  
     ScaleMode, A-21  
     ShowButtons, A-22  
     ShowErrorBoxes, A-23  
     TagServerName, A-24

UpperLimit, A-26  
 Value, A-27  
 Write method, A-28  
 WriteMode, A-28  
 ZeroPad, A-29  
 toolbar button, 2-1

## O

Offset property, A-18  
 OLE, SIMATIC controls  
   Button, 1-1  
   Number, 2-1–2-12  
   Slider, 3-1–3-3

## P

Panel control  
   events  
     MouseDown, B-6  
     MouseMove, B-7  
     MouseUp, B-8  
   properties, ShowErrorBoxes, A-23  
 Performance  
   connect/disconnect, C-4  
   Data control for critical data, C-3  
 Precision property, A-18  
 Process data, SIMATIC controls  
   Button, 1-1–1-5  
   Number, 2-1–2-12  
   Slider, 3-1–3-3  
 Programming  
   breakpoints, C-3  
   connect/disconnect, C-4  
   container guidelines, C-2  
   Data control for critical data, C-3  
   String functions when using a Number  
     control, 2-5, C-4  
   timers, C-3  
 Properties  
   Alignment, A-1  
   Appearance, A-1  
   BackColor, A-2  
   BorderStyle, A-3  
   Caption, A-4  
   CaptionAlignment, A-4  
   CaptionBackColor, A-5  
   CaptionFont, A-5  
   CaptionForeColor, A-6  
   CaptionSize, A-6  
   ConvertedValue, A-7

## Properties, continued

- DataType, A-8
- Direction, A-9
- Enabled, A-9
- Factor, A-10
- FalseCaption, A-11
- FalseColor, A-11
- Font, A-12
- ForeColor, A-12
- LargeChange, A-13
- LayoutType, A-14
- LimitCheck, A-14
- Locked, A-15
- LowerLimit, A-16
- Max, A-17
- Min, A-17
- Offset, A-18
- Precision, A-18
- Radix, A-19
- RangeRawMax, A-20
- RangeRawMin, A-20
- RangeScaledMax, A-20
- RangeScaledMin, A-20
- ScaleMode, A-21
- ShowButtons, A-22
- ShowErrorBoxes, A-23
- SIMATIC control properties
  - Button, 1-2-1-6
  - Number, 2-2-2-13
  - Slider, 3-2-3-4
- SmallChange, A-24
- TagServerName, A-24
- Ticks, A-24
- TrueCaption, A-25
- TrueColor, A-25
- UpperLimit, A-26
- Value, A-27
- Write method, A-28
- WriteMode, A-28
- ZeroPad, A-29

**R**

- Radix property, A-19
- RangeRawMax property, A-20
- RangeRawMin property, A-20
- RangeScaledMax property, A-20
- RangeScaledMin property, A-20
- Right string function when using a Number control, 2-5, C-4

**S**

- S7 controllers, SIMATIC controls
  - Button, 1-1-1-5
  - Number, 2-1-2-12
  - Slider, 3-1-3-3
- Slider control, properties
  - Direction, A-9
  - Enabled, A-9
  - LargeChange, A-13
  - Max, A-17
  - Min, A-17
  - SmallChange, A-24
  - Ticks, A-24
  - Value, A-27
- ScaleMode property, A-21
- ShowButtons property, A-22
- ShowErrorBoxes property, A-23
- SIMATIC controls
  - Button, 1-1-1-5
  - Button control
    - description, 1-1
    - toolbar button, 1-1
  - Number, 2-1-2-12
  - Number control
    - description, 2-1
    - toolbar button, 2-1
  - Slider, 3-1-3-3
  - Slider control
    - description, 3-1
    - toolbar button, 3-2
- Slider control
  - description, 3-1
  - events
    - Change, B-1
    - Click, B-1
    - DbClick, B-1
    - KeyDown, B-3
    - KeyPress, B-4
    - KeyUp, B-4
    - MouseDown, B-6
    - MouseMove, B-7
    - MouseUp, B-8
  - properties, 3-2-3-4
  - toolbar button, 3-2
- SmallChange property, A-24
- String function Right (VB), 2-5, C-4

**T**

TagServerName property, A-24  
Testing, VB string function (Right) when using  
    a Number control, 2-5, C-4  
Testing (breakpoints), C-3  
Third-party containers, C-2–C-4  
Ticks property, A-24  
TrueCaption property, A-25  
TrueColor property, A-25  
TWIP (measurement of labels), 2-7

**U**

UpperLimit property, A-26

**V**

Value property, A-27  
Vertical labels, 2-7  
Visual Basic  
    breakpoints, C-3  
    connect/disconnect, C-4  
    container guidelines, C-2  
    Data control for critical data, C-3  
    Right string function when using a Number  
        control, 2-5, C-4  
    timers (guidelines), C-3

**W**

Warnings  
    breakpoints in Visual Basic, C-3  
    emergency stop circuit, 1-2, 2-3, 3-2  
    Visual Basic timers, C-3  
Width (properties), 2-7  
Write method, A-28  
WriteMode property, A-28

**Z**

ZeroPad property, A-29





**To**

SIEMENS ENERGY & AUTOMATION INC  
ATTN: TECHNICAL COMMUNICATIONS M/S 519  
3000 BILL GARLAND ROAD  
PO BOX 1255  
JOHNSON CITY TN USA 37605-1255

**From**

Name: \_\_\_\_\_  
Job Title: \_\_\_\_\_  
Company Name: \_\_\_\_\_  
Street: \_\_\_\_\_  
City and State: \_\_\_\_\_  
Country: \_\_\_\_\_  
Telephone: \_\_\_\_\_

Please check any industry that applies to you:

- |   |   |
|---|---|
| <input type="checkbox"/> Automotive               | <input type="checkbox"/> Pharmaceutical |
| <input type="checkbox"/> Chemical                 | <input type="checkbox"/> Plastic        |
| <input type="checkbox"/> Electrical Machinery     | <input type="checkbox"/> Pulp and Paper |
| <input type="checkbox"/> Food                     | <input type="checkbox"/> Textiles       |
| <input type="checkbox"/> Instrument and Control   | <input type="checkbox"/> Transportation |
| <input type="checkbox"/> Non-electrical Machinery | <input type="checkbox"/> Other _____    |
| <input type="checkbox"/> Petrochemical            |   |



Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please give each of the following questions your own personal mark within a range from 1 (very good) to 5 (very poor).

1. Do the contents meet your requirements?

2. Is the information you need easy to find?

3. Is the text easy to understand?

4. Does the level of technical detail meet your requirements?

5. Please rate the quality of the graphics and tables.
- ☐

☐

☐

☐

☐

Additional comments: