The CoDeSys Visualization

Supplement to the User Manual for
PLC Programming
with CoDeSys 2.3

We have gone to great lengths to ensure this documentation is correct and complete. However, since it is not possible to produce an absolutely error-free text, please feel free to send us your hints and suggestions for improving it.

Trademark
Intel is a registered trademark and 80286, 80386, 80486, Pentium are trademarks of Intel Corporation.
Microsoft, MS and MS-DOS are registered trademarks, Windows is a trademark of Microsoft Corporation.

Publisher

3S - Smart Software Solutions GmbH
Fischerstraße 19
D-87435 Kempten
Tel. +49 831 5 40 31 - 0
Fax +49 831 5 40 31 – 50

Last update 20.08.2003
Version 1.0

Table Of Contents

1 Advantages of an integrated visualization 1-1

2 The Visualization Editor in CoDeSys 2-3

2.1 Create a new Visualization 2-4
2.2 Visualization Elements, Insert... 2-4
2.3 Positioning of Visualization Elements... 2-7
2.4 Visualization Configuration 2-10
2.4.1 Visualization Elements, Configure... 2-11
2.4.2 Visualization Object, Configure... 2-39
2.5 Dynamic Texts........ 2-43
2.5.1 Configuration 2-43
2.5.2 XML-File for Dynamic Texts........ 2-44
2.6 Calling up language-dependent Online Help via a visualization element........ 2-46
2.7 Placeholder Concept........ 2-46
2.8 Visualization in Online Mode... 2-48
2.9 Visualizations in libraries... 2-49

3 CoDeSys HMI 3-1

3.1 Installation, Start and Operating 3-1

4 Web-Visualization 4-1

4.1 Preconditions........ 4-1
4.2 Editing the WebVisu.htm file 4-1
4.3 Preparing a Web-Visualization in CoDeSys 4-2
4.4 Configuration and Start of the Web Server........ 4-2
4.5 Calling a Web-Visualization via Internet 4-3

5 Target-Visualization 5-1

5.1.1 Preconditions........ 5-1
5.1.2 Creating a Target-visualization 5-1
5.1.3 Starting the Target-Visualization........2

6 Index

1 Advantages of an integrated visualization

For visualizing, that means watching and operating the data of a controller which has is programmed with CoDeSys, no additional tool is required. The programming system contains an integrated visualization editor (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**), allowing the user to create visualization objects to the parallel to the development of the application in the same user interface.

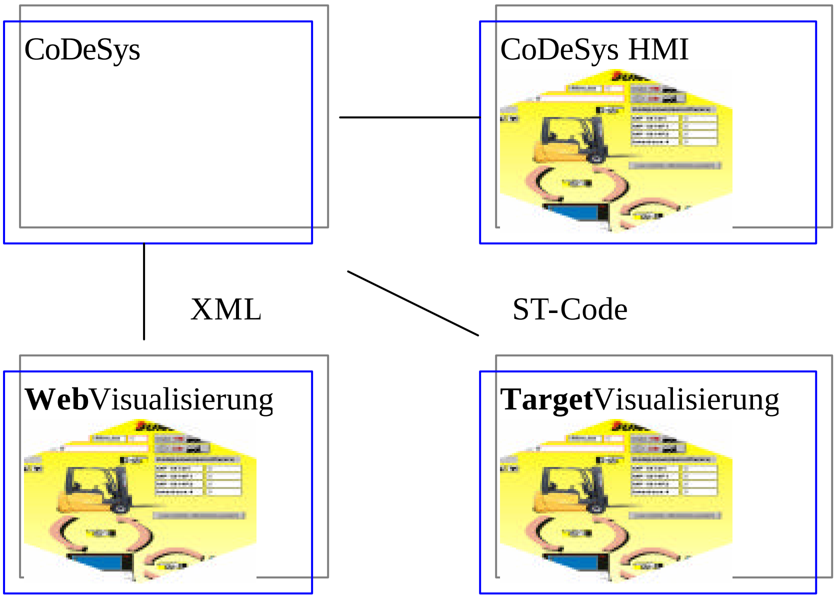
The integration offers a lot of benefits:

The visualization integrated in CoDeSys does not need a tag list and it can directly access the variables in the controller. Any OPC- or DDE-layer, which often is difficult to configure, is also not necessary, because the communication is done by the same mechanism which is used for the programming system. Thus the engineering effort for the realization of visualizations will be clearly reduced. Besides that the connection of controller and visualization allows running modes (see below), in which the conventional display on a PC with a monitor is not needed any longer.

One of the main advantages of the concept: The visualization masks which are created in the programming system can run in four different modes without requiring any modifications, if desired even parallel:

1. Directly in the programming system

For testing the created visualization masks, but also for service or diagnosis purposes in direct connection with the controller you do not need a further tool: In online mode you immediately get a display of the visualizations within the programming system.



2. CoDeSys HMI

The program CoDeSys HMI, which displays visualizations on a PC without the complete development user interface, also communicates via the programming interface. The configuration of OPC (OPC - OLE for Process Control) – resp. DDE (DDE – Dynamic Data Excange) inteface is not necessary. See chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**.

3. Web-Visualization

Optionally CoDeSys generates a description in XML format from the visualization data, which is loaded to the controller together with a Java-Applet (Applet – a program developped in the programming language JAVA, which can be processed by browsers) and which can be displayed via TCP/IP on a browser. Thus the visuaization data are available online on various platforms. See chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**.

4. Target-Visualization

For controllers with integrated display the visualization data can be loaded from the programming system together with the application to the target system. There they get displayed automatically. This solution can be portated with low effort to any devices which are programmable with CoDeSys. See chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**.

Overview on the functions:

- Elements
- Rectangle, Ellipse, Rounded rectangle
- Line, Polygon, Polyline, Curve
- Bitmap, WMF-file
- Button, Table, Histogram, Bar Display, Meter
- Reference to another visualization
- Animations (depending on element type):
- Text display
- Color changes
- Visible/Invisible
- Shift
- Rotation
- Scaling
- Offset on the particular edges of an object (for Bar Display)
- Button activ/inactiv
- Current line (only text display)
- Input possibilities:
- Toggle/tap Boolean value
- Text input
- Change of visualization
- Special actions (Leave visualization, Read/Write receipts, Switch language, call external EXE etc.)
- Choose line (only text display)
- Further properties
- Switching language
- Tooltips for all elements
- ASCII Import/Export
- Background Bitmap
- Automatic Scaling
- Drawing operations: Alignment, Order, Grouping
- Placeholder concept for creating objects with complex graphic elements
- Programmed visualization expressions

2 The Visualization Editor in CoDeSys

A visualization is a **graphical representation** of the project variables which allows inputs to the PLC program in online mode via mouse and keypad. The CoDeSys **visualization editor**, which is part of the programming system provides graphic elements which can be arranged as desired and can be connected with project variables. Thereupon in online mode the look of the graphical elements will change depending on the variables values.
Simple example: In order to represent a fill level, which is calculated by the PLC program, draw a bar and connect it to the corresponding project variable, so that the length and color of the bar will show the current fill level value. Add a text field which will display the current value in a text string and a button for starting and stopping the program.

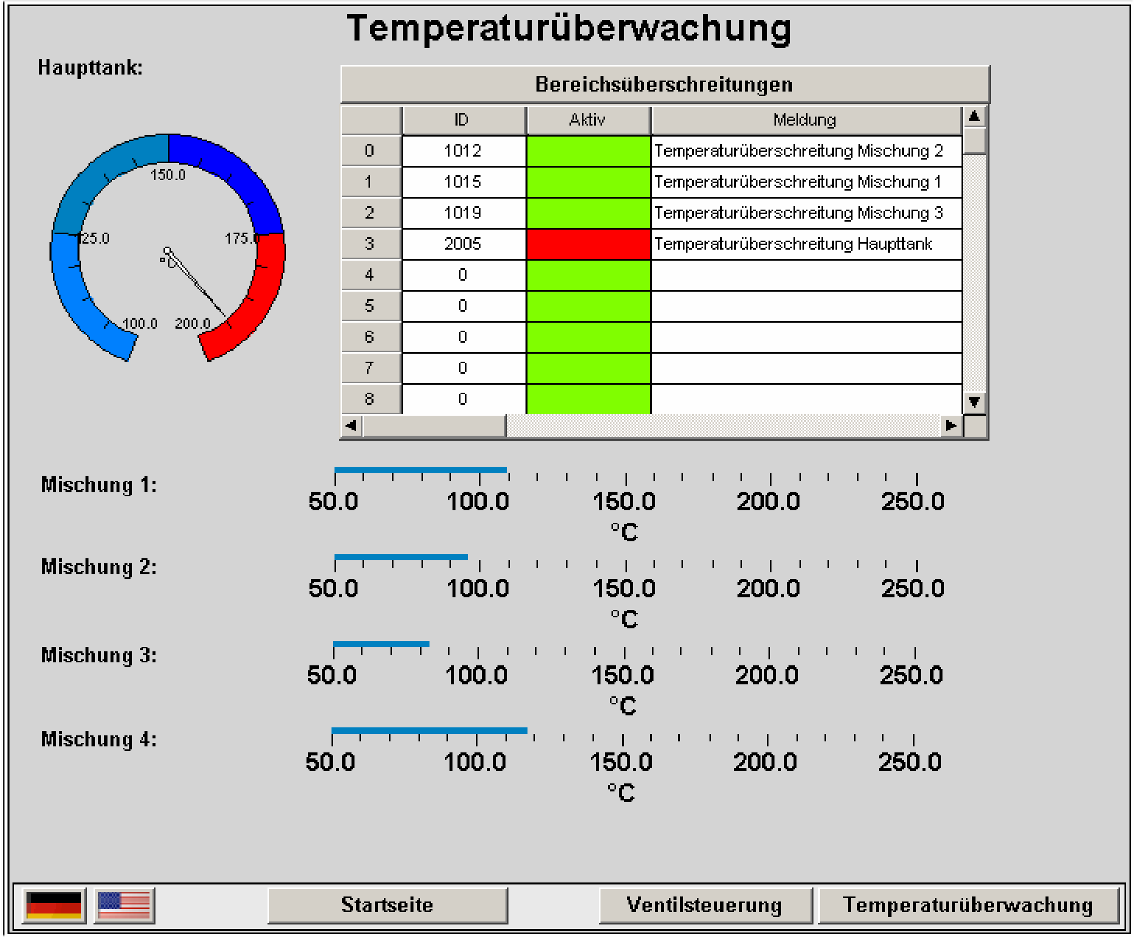
The properties of a single visualization element as well as of the whole visualization object will be defined in appropriate **configuration** dialogs. There it is possible to set basic parameters by activating options as well as to define a dynamic parameterizing by entering project variables.

Additional special possibilities for configuring are given by the programability of element properties via structure variables.

Using placeholders in the configuration dialogs may save a lot of effort in case you want to use the same visualization object several times with different configurations.

Also regard the option to define special hotkeys for each particular visualization.

Example of a visualization



Bereichsüberschreitungen

	ID	Aktiv	Meldung
0	1012	🟩	Temperaturüberschreitung Mischung 2
1	1015	🟩	Temperaturüberschreitung Mischung 1
2	1019	🟩	Temperaturüberschreitung Mischung 3
3	2005	🟥	Temperaturüberschreitung Haupttank
4	0	🟩	
5	0	🟩	
6	0	🟩	
7	0	🟩	
8	0	🟩	

2.1 Create a new Visualization

A **visualization object** is a CoDeSys object which is managed in the 'Visualization' register of the Object Organizer. It contains an arrangement of **visualization elements** and can get certain object properties. One or several visualization objects can be created in a CoDeSys project and might be linked with each other.

In order to create a visualization object in the Object Organizer, you must select the register card for **Visualization** in the Object OrganizerUsing the **'Project' 'Object Add'** command, you can create a new visualization object. Open the 'New visualization' dialog, in which you can enter the name of the new visualization. Once a valid entry is made, that is not a name that is already in use and no special characters used, you can close the dialog with **OK**. A window opens, in which you can edit the new visualization.

Please regard: If you want to use the implicit variable **CurrentVisu** (type STRING) for addressing the currently opened visualization object, you throughout must use capital letters for the names of the visualization objects (e.g. "`PLC_VISU`").

2.2 Visualization Elements, Insert...

A visualization element is a graphical element, which is used to fill a visualization object. The available elements are offered in the CoDeSys menu bar. Each element gets a separate configuration.

You can insert various geometric forms, as well as bitmaps, metafiles, buttons and existing visualizations, into your visualization.

Geometric forms at your disposal *include:* rectangles, rounded rectangles, ellipses/circles, and polygons.

Go to the 'Insert' menu item and select freely from the following commands: **'Rectangle'**, **'Rounded Rectangle'**, **'Ellipse'**, **'Polygon'**, **Polyline'**, **'Curve'**, **'Pie'**, **'Bitmap'**, **'Visualization'**, **'Button'**, **'Table'**, **'Meter'**, **'Bar Display'**, **'Histogram'**, **'WMF file'**. A check appears in front of the selected command. You can also use the tool bar. The selected element appears pushed down(e.g.).

If you now go to the editor window with the mouse, you will see that the mouse pointer is identified with the corresponding symbol (e.g.). Click on the desired starting point of your element and move the pointer with pressed left mouse key until the element has the desired dimensions.

If you want to create a polygon or a line, first click with the mouse on the position of the first corner of the polygon resp. on the starting point of the line, and then click on the further desired corner points. By doubleclicking on the last corner point you will close the polygon and it will be completely drawn respectively the line will be completed. If you want to create a curve (Bezier curves) determine the initial and two other points with mouse clicks to define the circumscribing rectangle. An arc is drawn after the third mouse click. You can then change the position of the end point of the arc by moving the mouse and can then end the process with a double click or add another arc with additional mouse clicks.

Furthermore pay attention, to the status bar and the change from select and insert modes.

'Insert' 'Rectangle'

Symbol:

With the command you can insert a rectangle as an element into your present visualization. (Use, see Visualization Elements, Insert)

'Insert' 'Rounded Rectangle'

Symbol:

With the command you can insert a rectangle with rounded corners as an element in your present visualization. (Use, see Visualization Elements, Insert).

'Insert' 'Ellipse'

Symbol:

With the command you can insert a circle or an ellipse as an element in your present visualization. (Use, see Visualization Elements, Insert).

'Insert' 'Polygon'

Symbol:

With the command you can insert a polygon as an element in your present visualization. (Use, see Visualization Elements, Insert).

'Insert' 'Polyline'

Symbol:

With the command you can insert a line as an element into your current visualization. (Use, see Visualization Elements, Insert).

'Insert' 'Curve'

Symbol:

With the command you can insert a Bezier curve as an element into your current visualization. (Use, see Visualization Elements, Insert).

'Insert' 'Pie'

Symbol:

Use this command to insert a Pie Segment as an element into your current visualization.

While pressing the left mouse button, bring up an area in the desired size. An oval element including a line marking the radius at the 0° position will be displayed. As long as keeping the mouse button pressed you can immediately change size and position of the element by moving the mouse. A little black square is attended to the the element, indicating the corner of a virtual rectangle surronding the element.

In order to define the start and end angles of a Pie, select the end point of the radius line on the circular arc by a mouse-click. As soon as you, keeping the mouse button pressed, move the cursor, two small rectangles will be displayed, indicating the two angle positions. As from now those can be selected and moved seperately. If you want the angle values get defined dynamically by variables, open the configuration dialog category 'Angle' and enter the desired variable names.

You can resize or reshape the element later by either clicking on the centre point, the cursor getting displayed as diagonally crossed arrows, and moving the mouse while keeping the mouse button pressed (or using the arrow keys). Alternatively you can select and move the corner indicating little square outside of the element. In order to move the element to another position, click inside the element to get the cursor beeing displayed as vertically crossed arrows and then move the cursor.

'Insert' 'Bitmap'

Symbol:

With the command you can insert a bitmap as an element in your present visualization. (Use, see Visualization Elements, Insert).

While pressing the left mouse button, bring up an area in the desired size. The dialog box is opened for opening a file. Once you have selected the desired bitmap, it will be inserted into the area brought up. You can define in the bitmap configuration dialog, whether a just link to the bitmap file should be stored or the bitmap should be inserted as an element.

'Insert' 'Visualization'

Symbol:

With the command you can insert an existing visualization as an element in your present visualization. (Use, see Visualization Elements, Insert).

While pressing the left mouse button, bring up an area in the desired size. A selection list of existing visualizations opens. After you have selected the desired visualization, it will be inserted in the defined area.

An inserted visualization will also be named as a reference.

'Insert' 'Button'

Symbol:

This command is used to insert a button into your current visualization. (Use, see Visualization Elements, Insert).

Drag the element to the desired size with the left mouse button held down.

If a toggle variable is configured for the button it displays the state of this variable by visually displaying whether it is pressed or not pressed. Conversely, the variable is toggled by „pressing" the button.

'Insert' 'WMF file'

Symbol:

This command is used to insert a Windows Metafile. The standard dialog for opening a file will appear, where you can select a file (extension *.wmf). After having closed the dialog with OK the file will be inserted as an element in the visualization. Please regard, that no link to a file will be saved, like it is done when you insert a bitmap, but the elements of the metafile will be inserted as a group.

'Insert' 'Table'

Symbol:

Use this command to insert a Table element as an element into your current visualization. It is used to display the current values of the elements of an array.

While pressing the left mouse button, bring up an area in the desired size. Before the element gets displayed the configuration dialog 'Configure table" will be opened. Here you will find additionally to the standard categories Tooltip and Security the categories 'Table', 'Columns', 'Rows' and 'Selection' where you can define contents and appearance of the table.

'Insert' 'Meter'

Symbol:

Use this command to insert a Meter as an element into your current visualization. It provides a scale which is defined as a sector of a circular arc, and a pointer element.

While pressing the left mouse button, bring up an area in the desired size. Before the element gets displayed the configuration dialog Configure Meter will be opened. Here you can define various parameters concerning the display of the element and a preview is available to check the configuration before really inserting the element by confirming the dialog.

'Insert' 'Bar Display'

Symbol:

Use this command to insert a Bar Display element into your current visualization. It is used to visualize the value of the assigned variable by a bar indicating the value by its lenght along a horizontal scale.

While pressing the left mouse button, bring up an area in the desired size. Before the element gets displayed the configuration dialog 'Configure bar display' will be opened. Here you can define various parameters concerning the display of the element and a preview is available to check the configuration before really inserting the element by confirming the dialog.

'Insert' 'Histogram'

Symbol:

Use this command to insert a Histogram element into your current visualization. It is used to visualize the elements of an array by bars which are placed side by side each indicating the value of the element by its length.

While pressing the left mouse button, bring up an area in the desired size. Before the element gets displayed the configuration dialog 'Configure Histogram' will be opened. Here you can define various parameters concerning the display of the element and a preview is available to check the configuration before really inserting the element by confirming the dialog.

2.3 Positioning of Visualization Elements...

Selecting Visualization Elements

The selection mode is activated by default. Um ein Element In order to select an element, click with the mouse on the element. You can also select the first element of the elements list by pressing the <Tab> key and jump to the next by each further keystroke. If you press the <Tab> key while pressing the <Shift> key, you jump backwards in the order of the elements list.

In order to select elements, which are placed one upon the other, first select the top level element by a mouseclick. Then do further mouseclicks while the <Ctrl> button is pressed, to reach the elements in the underlying levels .

In order to mark multiple elements, press and hold the <Shift> key and click the corresponding elements, one after another; or, while holding down the left mouse button, pull a window over the elements to be selected.

In order to select all the elements, use the 'Extras' 'Select All' command.

If you are in the element list (called by **'Extras' 'Element list'**), you can select the concerned element in the visualization by selecting a line.

Changing the Selection and Insert Mode

After the insertion of a visualization element, there is an automatic change back into the selection mode. If you want to insert an additional element the same way, you can once again select the corresponding command in the menu or the symbol in the tool bar.

You can also quickly change between the selection mode and the insert mode by pressing the <Ctrl>-key and the right mouse button simultaneously.

In the insert mode, the corresponding symbol will also appear at the mouse pointer, and the name will also be indicated in black in the status bar.

'Extras' 'Select'

This command is used to switch the selection mode on or off. This can also be achieved using the symbol or by pressing the right mousekey while holding down the key at the same time.

'Extras' 'Select All'

This command allows you to select all visualization elements within the current visualization object.

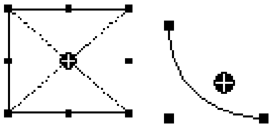
Copying Visual Elements

One or more selected elements can be inserted with the **'Edit' 'Copy'**command, the <Ctrl>+<C> key combination, or the corresponding copy symbol, and with **'Edit' 'Paste'**.

A further possibility is to select the elements and to again click in one of these elements with the **<Ctrl>** key held down. If you now hold the left mouse button down, you can separate the elements thus copied from the original.

Modifying Visualization Elements

You can select an element which has already been inserted by a mouse click on the element or by pressing the <tab> key. A small black square will appear at each corner of each of the elements, (with ellipses at the corners of the surrounding rectangle). Except in the case of polygons, lines or curves further squares appear in the middle of the element edges between the corner points.



With a selected element, the turning point (balance point) is also displayed at the same time. You can then rotate the element around this point with a set motion/angle. The turning point is displayed as a small black circle with a white cross (⊕). You can drag the turning point with a pressed left mouse button.

You can change the size of the element by clicking on one of the black squares and, while keeping the left mouse button pressed, controlling the new outline.

With the selection of a *polygon*, you can drag each individual corner using the same technique. While doing this, if you press the <Ctrl>-key then an additional corner point will be inserted at the corner point, an additional corner point will be inserted, which can be dragged by moving the mouse. By pressing the <Shift>+<Ctrl>-key, you can remove a corner point.

Dragging Visualization Elements

One or more selected elements can be dragged by pressing the left mouse button or the arrow key.

Grouping Elements

Elements can be grouped by selecting all desired elements and performing the command 'Extras' 'Group'. The group will behave like a single element:

- the grouped elements get a collective frame; when dragging the frame all elements will be stretched or compressed; only the group can be moved to another position.
- the grouped elements get collective properties: inputs only can effect the group and not a single element. Thus the elements also get one collective configuration dialog (category 'group'). The property 'Change color' can not be configured for a group !

To redefine a single element of a group, the grouping must be redone by the command 'Extras' 'Break up group'. The configuration of the group will be lost in this case.

Note: As soon as you save the project as CoDeSys Version 2.1 or lower, a group of visualization elements will be resolved automatically; that means that the elements of the group will be shown as single elements in the visualization.

'Extras' 'Send to Front'

Use this command to bring selected visualization elements to the front.

'Extras' 'Send to Back'

Use this command to send selected visualization elements to the back.

'Extras' 'Align'

Use this command to align selected visualization elements.

The following alignment options are available:

- **Left**: the left edge of each of the elements will be aligned to the element that is furthest to the left
- the same is true for **Right** / **Top** / **Bottom**
- **Horizontal Center**: each of the elements will be aligned to the average horizontal center of all elements
- **Vertical Center**: each of the elements will be aligned to the average vertical center of all elements

'Extras' 'Elementlist'

This command opens a dialog box containing a list of all visualization elements including their **number, type** and **position**. The position is given according to the x and y position of the upper left (x1, y1) and the lower right (x2, y2) corner of the element.

When one or more items have been selected, the corresponding elements in the visualization are marked for visual control and if necessary the display will scroll to that section of the visualization that contains the elements.

Use the **To front** button to bring selected visualization elements to the front. Use the **To behind** button to move them to the back.

Below the elements list there you find – depending on which element is currently selected - one of the following combinations of edit fields where you can modify size and position of the element:

- If a rectangle, rounded rectangle, ellipse, bitmap, visualization, button or a meta file is currently selected, then next to the text "**Rectangle (x1, y1, x2, y2)**" there are four edit fields, where the actual x/y positions are shown and can be modified.
- If a line, polygon or a curve is currently selected, a table will be available showing the actual **X-Position** and **Y-Position** of each of the black squares which mark the shape of the element, as soon as it is selected. These values can be edited here.

To set the modified position values in the elements list and in the visualization, press button **Set rectangle** (in case 1.) resp. **Set polygon** (in case 2.).

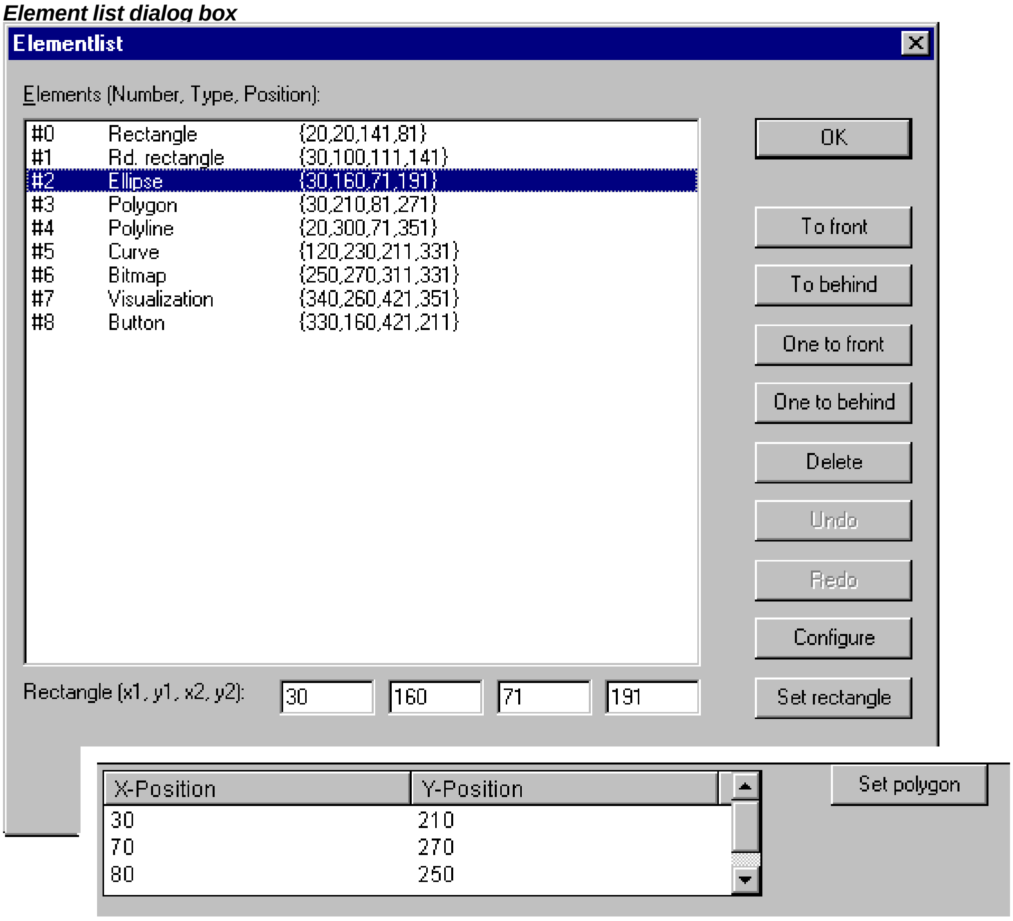
Use the **Delete** button to remove selected visualization elements.

Use the Undo and Redo buttons to undo or restore changes that have been made just as you would do with the commands 'Edit' 'Undo' and 'Edit' 'Redo' . In the dialog box, you can observe the changes that are being made.

Click on **OK** to close the dialog box and confirm the changes.

Use **Configure** to get the configuration dialog for the element.

Element list dialog box



Elements (Number, Type, Position):

#0	Rectangle	{20,20,141,81}
#1	Rd. rectangle	{30,100,111,141}
#2	Ellipse	{30,160,71,191}
#3	Polygon	{30,210,81,271}
#4	Polyline	{20,300,71,351}
#5	Curve	{120,230,211,331}
#6	Bitmap	{250,270,311,331}
#7	Visualization	{340,260,421,351}
#8	Button	{330,160,421,211}

Rectangle (x1, y1, x2, y2): 30 160 71 191

X-Position	Y-Position
30	210
70	270
80	250

Status Bar in the Visualization

If a visualization has the focus, the current **X** and **Y position** of the mouse cursor in pixels relative to the upper left corner of the image is displayed in the status bar. If the mouse pointer is located on an **Element**, or if the element is being processed, then the number of the element will be displayed. If you have selected an element to insert, then this element will also appear (for example, **Rectangle**).

2.4 Visualization Configuration

When configuring a visualization you have to distinguish between the configuration a particular graphic element and the visualization object as a whole. Correspondingly a different selection of configuration dialogs will be available, which can be opened by the command Configure' from Menu 'Extras' resp. from the context menu.

In this dialogs the properties of an element or object are set either by activating options or dynamically by inserting project variables. Besides that the properties can be programmed via the components of a structure variable, which can be defined for each visualization element.

Regard the order of analysis, which will be followed in online mode:

- The values which are given dynamically, i.e. via project variables, will overwrite the fix parameters defined for the same property.
- If an element property is defined by a "normal" project variable as well as by the component of a structure variable, then in online mode primarily the value of the project variable will be regarded.

Please regard the possibility of using Placeholders as well as the special input possibilities which are useful if the visualization should be used in CoDeSys HMI resp. as Target or Web-visualization , that means if the visualization serves as the only user inteface for a PLC programm (see also: INTERN in category 'Input' of the configuration dialogs, Keyboard usage)

Placeholders

At each location in the configuration dialog at which variables or text are entered, a **placeholder** can be set in place of the respective variable or text. This makes sense if the visualization object is not to be used directly in the program, but is created to be inserted in other visualization objects as an "instance". When configuring such an **Instance**, the placeholders can be replaced with variable names or with text.

See chapter 2.7 for the placeholder concept.

2.4.1 Visualization Elements, Configure...

'Extras' 'Configure'

With this command, the 'Configure element' dialog opens for configuring the selected visualization element (see Select visualization element). You are given the dialog box when you doubleclick on the element.

Select a category in the left area of the dialog box (available categories depending on element type) and fill out the requested information in the right area. This has to be done by activating options resp. by inserting the name of valid variables, whose values should define the property of the element.

Note:	There are also configuration dialogs available for a group of elements. Regard that the settings will be valid for the "element" group. If you want to configure the particular elements of the group, you have to resolve the group.
Note:	If you have defined an element property by a "static" setting as well as dynamically by a variable, then in online mode the variable will overwrite the static value (Example: "Alarm color Inside" can be defined statically in category 'Color' and additionally dynamically in category 'Colorvariables' by a variable). If the setting is controlled by a "normal" project variable as well as by a structure variable, then the value of structure variable also will be overwritten by the "normal" project variable.

Please regard: Meter, Bar Display and Histogram must be re-grouped before !

At locations in the element configuration where variables are operative, the following **Entries** are possible:

- Variable names, for which input assistant is available
- Expressions which are assembled from component accesses, field accesses with constant index, variables and direct addresses.
- Operators and constants, which can be combined at will with the aforementioned expressions.
- Placeholders instead of variable names or text strings

Examples of permissible expressions:

```
x + y
100*PLC_PRG.a
TRUE
NOT PLC_PRG.b
9*sin(x + 100)+cos(y+100)
```

Function calls are not possible. Invalid expressions result in an error message on login („Invalid Watch expression..."). Examples of invalid expressions: fun(88), a := 9, RETURN.

There are two possible ways in the configuration dialogs to write **global variables**: „.globvar" and „globvar" are equivalent. The style with a dot (which is that used in the Watch- and Receipt Manager) is not allowed within an assembled expression, however.

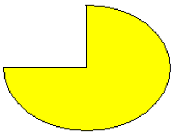
Regard also the possibility of using Placeholders (see chapter 2.7).

Angle

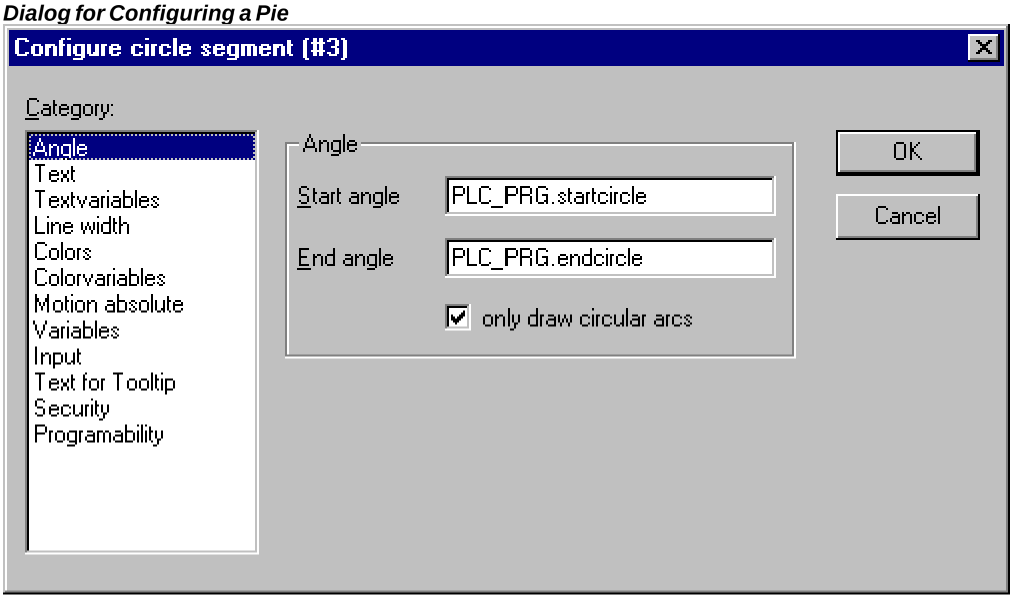
In the configuration dialog 'Configure Pie' in the **Angle** category you can each enter a value or a variable defining the **start angle** and the **end angle** of the sector element in degrees. The sector will be drawn clockwise from the start angle position to the end angle position.

Example:

Enter start angle: "90", end angle: "180"



Dialog for Configuring a Pie



Angle

Start angle: PLC_PRG.startcircle

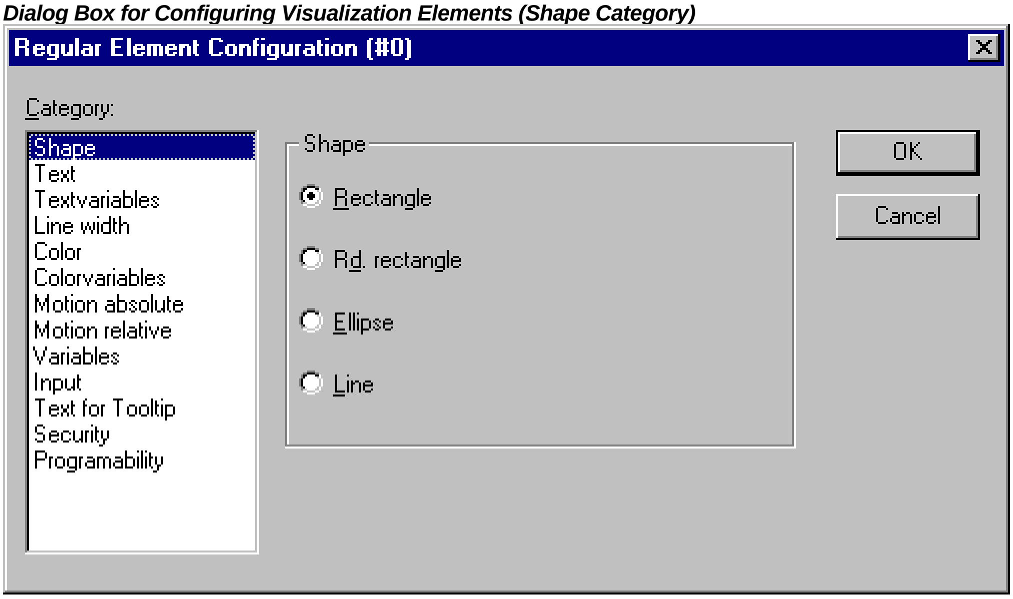
End angle: PLC_PRG.endcircle

☑ only draw circular arcs

Shape

In the visualization element configuration dialog box, you can select in the **Shape** category from among **Rectangle**, **Rounded Rectangle**, **Line** and **Ellipse** respectively **Polygon, Line** and **Curve**. The form will change into the size already set.

Dialog Box for Configuring Visualization Elements (Shape Category)



Shape

● Rectangle

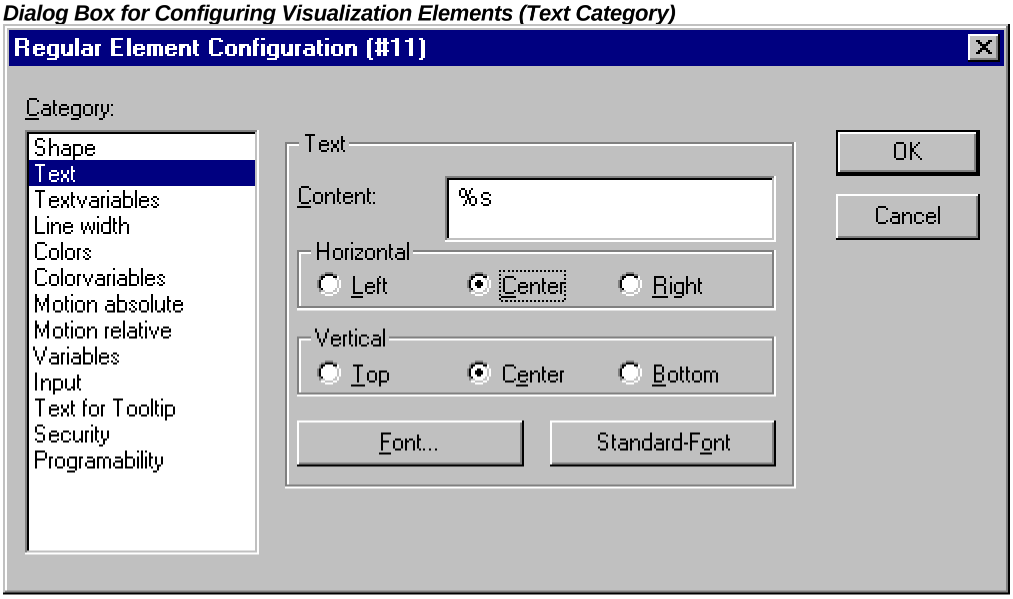
○ Rd. rectangle

○ Ellipse

○ Line

Text

Dialog Box for Configuring Visualization Elements (Text Category)



Content: %s

Horizontal: ○ Left ● Center ○ Right

Vertical: ○ Top ● Center ○ Bottom

In the dialog for configuring visualization elements, you can specify a text for the element in the **Text** category. This can be entered directly or/and a variable can be defined which will determine the text string. The usage of placeholders is possible. Also the default settings for font and alignment are done here.

Note: As soon as text parameters are additionally provided dynamically, which means by a system or structure variable (see also below, categorie 'Text variables' resp. 'Programmability'), the static definitions which are done in the currently opened dialog , will be overwritten !

In case of multiple definition of an element property consider the specific order of precedence concerning according to which a value might be overwritten in online mode by another.

Enter the text in the **Content** field. With the key combination <Crtl>+<Enter> you can insert line breaks, with <Crtl>+<Tab>, tab stops. Besides the input of a pure text string you can use the following formatting sequences:

- If you include "**%s**" into the text, then this location, in Online mode, will be replaced by the value of the variable from the **Text Output** field of the **Variables** category. You also can use a formatting string, which conforms with the standard C-library function 'sprintf': e.g. "%d" for integer types, "%f" for floating types, "%2.5f" for floating types with 2 decimal places before and 5 behind the comma, "%x" for integer types in hexadecimal format. The value of the variable will be displayed correspondingly in online mode. You can enter any IEC-conforming format strings, which fit to the type of the used variable. Attention: It is not checked whether the type which is used in the formatting string matches with the type of the variable which is defined in the 'Text Output' field !

  Example:
  Input in the 'Content' field: `Fill level %2.5f mm`
  Input in the 'Text Output' field e.g.: `fvar1 (REAL variable)`
  -> Output in online mode e.g.: `Fill level 32.8999 mm`

- If you enter "**%t**", followed by a certain sequence of special placeholders, then this location will be replaced in Online mode by the system time. The placeholders define the display format, see the following table. Attention: Do not insert any other characters before %t in the 'Content' field (in contrast this is allowed for e.g. "%s", see above)

%a	Abbreviated weekday name
%A	Full weekday name

%b	Abbreviated month name
%B	Full month name
%c	Date and time representation appropriate for locale
%d	Day of month as decimal number (01 – 31)
%H	Hour in 24-hour format (00 – 23)
%I	Hour in 12-hour format (01 – 12)
%j	Day of year as decimal number (001 – 366)
%m	Month as decimal number (01 – 12)
%M	Minute as decimal number (00 – 59)
%p	Current locale’s A.M./P.M. indicator for 12-hour clock
%S	Second as decimal number (00 – 59)
%U	Week of year as decimal number, with Sunday as first day of week (00 – 53)
%w	Weekday as decimal number (0 – 6; Sunday is 0)
%W	Week of year as decimal number, with Monday as first day of week (00 – 53)
%x	Date representation for current locale
%X	Time representation for current locale
%y	Year without century, as decimal number (00 – 99)
%Y	Year with century, as decimal number
%z, %Z	Time-zone name or abbreviation; no characters if time zone is unknown
%%	Percent sign

Examples:

`%t%a %b %d.%m.%y %H:%M:%S`
-> Display in online mode: `Wed Aug 28.08.02 16:32:45`

Between the placeholders you can insert any text strings:
`%Today is %d.%m.%y`
-> Display in online mode: Today is `28.08.02`

Note: If a text string is to be transferred into a **translation file**, which will then be used in Online mode to enable switching into another national language, it must be delimited at the beginning and end by #.
Examples: "#Pump 1#" or else even "#Pump# 1"
The second case might for example, in the event of multiple occurrences of the text Pump (Pump 1, Pump 2, etc.), prevent multiple appearances in the translation.

- If you include "**%<PREFIX>**" into the text, you can enter instead of "PREFIX" a certain string, which will serve as an identifier concerning the use of dynamic texts. The prefix will be used together with an ID number, which is to be defined in the 'Variables' category of the configuration dialog in field 'Textdisplay'. The combination references to a certain text, which is contained in a xml-file listing all possible dynamic texts.Thus at run time the text which is indicated by the current ID-Prefix-combination will be displayed. For further information see also the description of 'Settings', category Language.

The configured text will appear online in the prescribed alignment within the element: **horizontally left**, **center** or **right** and **vertically top**, **center** or **bottom**.

If you use the **Font** button, a dialog box for selection of the font will appear. Select the desired font and confirm the dialog with **OK**. With the **Standard-Font** button you can set the font that is selected in the project options (Project' 'Options' 'Editor'**).** If the font is changed there, then this font will be displayed in all elements except in those elements for which another font has explicitly been selected by using the **Font** button.

Textvariables

In category **Textvariables** of the dialog for configuring visualization elements you can specify a variable which should dynamically set color and font of that string which is defined in category 'Text'. At best enter the variable name with the aid of the input assistant (<F2>).

You can also use components of the structure *VisualObjectType* to set the text properties. For this see the description of category 'Programability'; there you will find the possible values of the particular structure components and their effect.

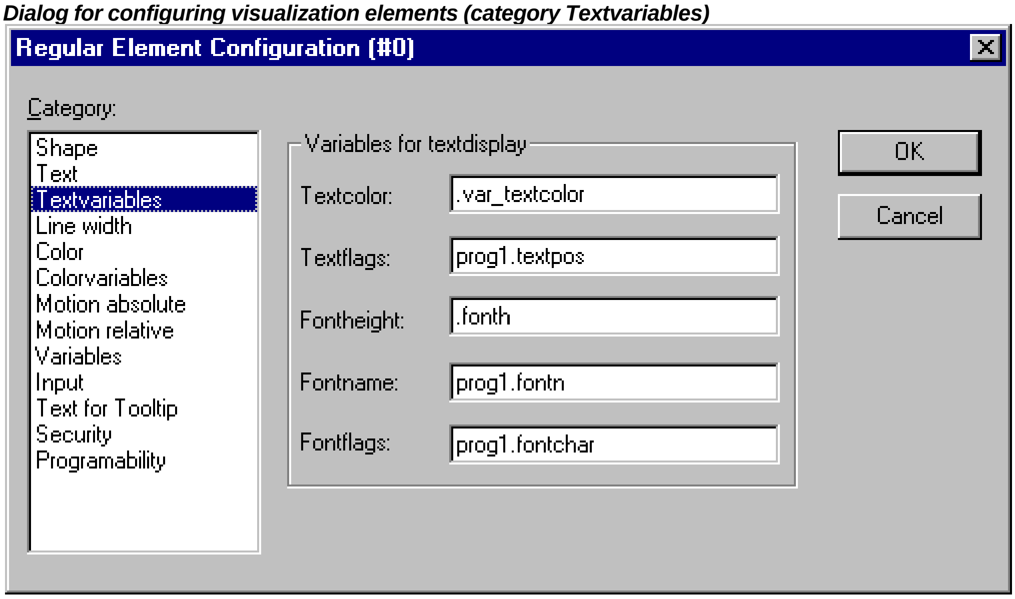
Note: If there are corresponding static definitions in category 'Text', these will be overwritten by the dynamic parameter values.

In case of multiple definition of an element property consider the specific order of precedence concerning according to which a value might be overwritten in online mode by another.

The parameters of the dialog:

Parameter:	Meaning:	Example entry of project variable:	Example Usage of variable in program:	corresponding component of structure VisualObjectType:
Textcolor:	Text color	"plc_prg.var_textcolor"	var_textcolor=16#FF00FF → Farbe	dwTextColor
Textflags:	Alignment (right, left, centered...)	"plc_prg.textpos"	textpos:=2 → Text right justified	dwTextFlags
Fontheight:	Font height in Pixel	".fonth"	fonth:=16; → Font height 16 pt	ntFontHeight
Fontname:	Font name	"vis1.fontn"	fontn:=arial; → Arial is used	stFontName
Fontflags:	Font display (bold, underlined, italic...)	"plc_prg.fontchar"	fontchar:=2 → Text will be displayed bold	dwFontFlags

Dialog for configuring visualization elements (category Textvariables)



Regular Element Configuration (#0)

Category: Textvariables

Variables for textdisplay

Textcolor: .var_textcolor

Textflags: prog1.textpos

Fontheight: .fonth

Fontname: prog1.fontn

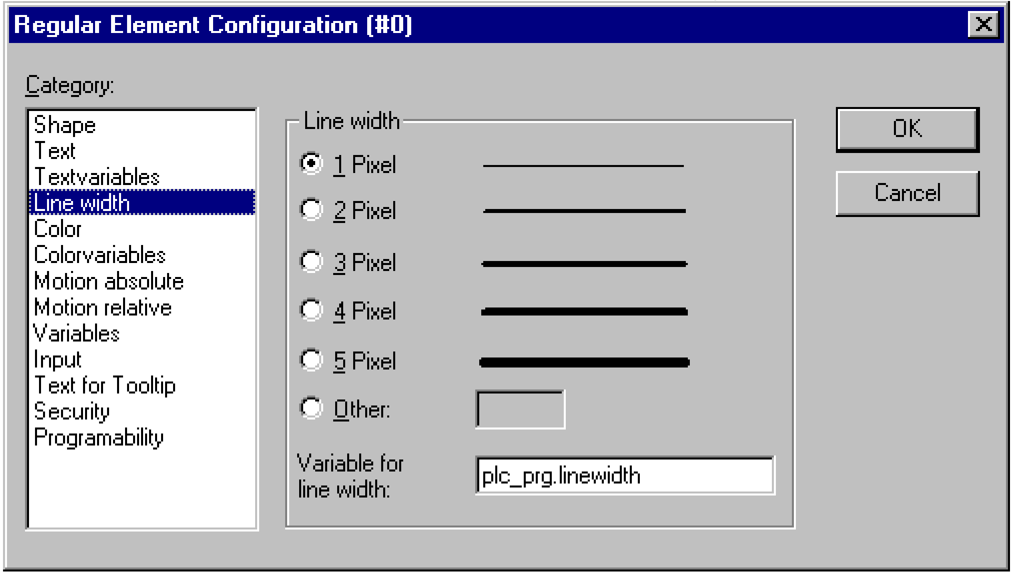
Fontflags: prog1.fontchar

Line width

In the dialog for configuring visualization elements, you can choose the line width for an element. As predefined options you find width settings from 1 to 5 pixel, additionally an other value can be entered manually (**Other:**), or a project variable (**Variable for line width:**) can be inserted. For the latter the input assistance ((<F2>) can be used.

Note: As soon as the parameter is additionally defined dynamically, i.e. by a structure variable (see below, category 'Programmability'), the static setting will be overwritten in online mode.

Dialog Box for Configuring Visualization Elements (Line width category)



Regular Element Configuration (#0)

Category: Line width

Line width

● 1 Pixel

○ 2 Pixel

○ 3 Pixel

○ 4 Pixel

○ 5 Pixel

○ Other:

Variable for line width: plc_prg.linewidth

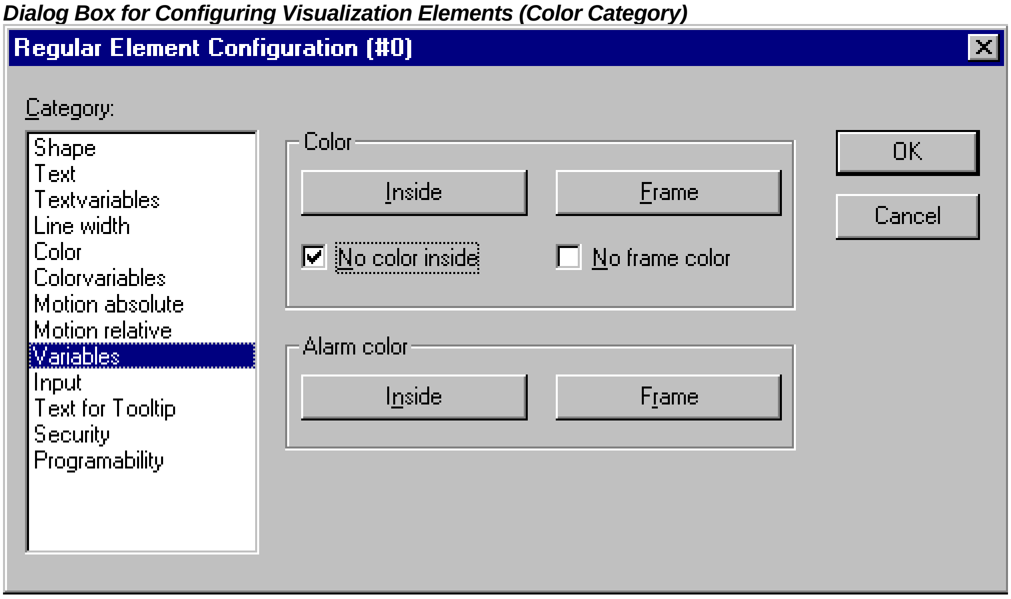
Colors

In the visualization element configuration dialog box, in the **Color** category you can select primary colors and alarm colors for the inside area and for the frame of your element. Chosing the options **no color inside** and **no frame color** you can create transparent elements.

Note: As soon as the parameter is additionally defined dynamically by a variable, the static setting will be overwritten in online mode.

In case of multiple definition of an element property consider the specific order of precedence concerning according to which a value might be overwritten in online mode by another.

Dialog Box for Configuring Visualization Elements (Color Category)



☑ No color inside

☐ No frame color

If you now enter a Boolean variable in the **Variables** category in the **Change Color** field, then the element will be displayed in the **Color** set, as long as the variable is FALSE. If the variable is TRUE, then the element will be displayed in its **Alarm Color**.

Note: The change color function only becomes active, if the PLC is in *Online Mode*!

If you want to change the color of the frame, then press the **Frame** button, instead of the **Inside** button. In either case, the dialog box will open for selection of the color.

Here can to choose the desired hue from the primary colors and the user-defined colors. By pressing the Define Colors you can change the user-defined colors.

Color Variables

Here you can enter project variables (e.g. PLC_PRG.color_inside), which should determine the particular property in online mode: These property definitions also or additionally can be programmed with the aid of components of the structure VisualObjectType. Therefore see the description on the "Programability" of a visualization element. There you will find a list of the possible values and their effects.

Note: The variables, entered in the Color Variables dialog, in online mode will overwrite the static values given in the 'Color' category as well as corresponding values given by a structure variable.

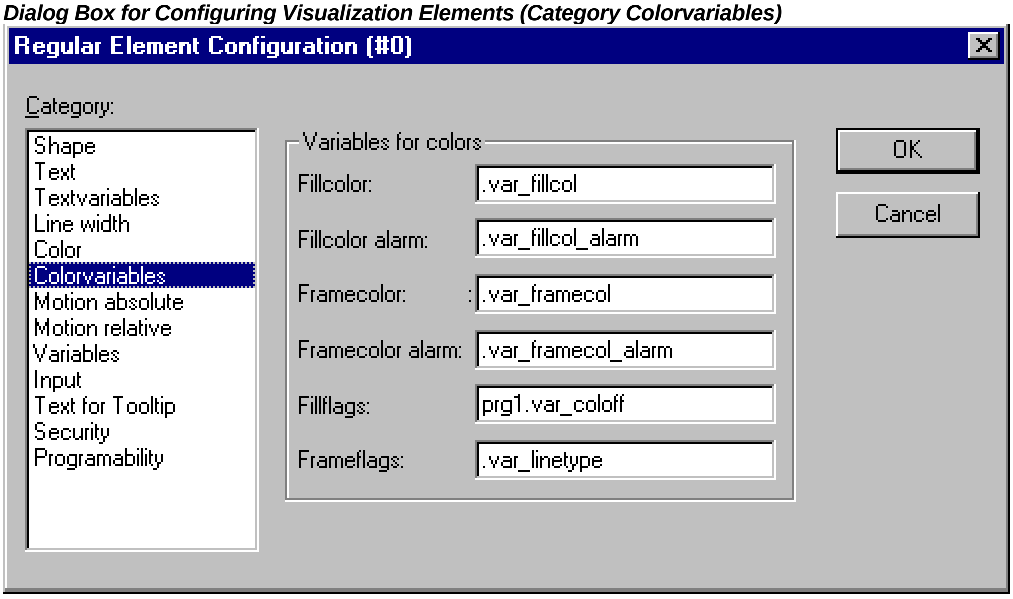
In case of multiple definition of an element property consider the specific order of precedence concerning according to which a value might be overwritten in online mode by another.

The parameters of the dialog::

Parameter:	**Description:**	**Example of an entry:**	**Example for using the variable in the program:**	**corresponding component of structure VisualObjectType:**
Fillcolor:	fill color	"plc_prg.var_fillcol"	var_var_fillcol:=16#FF00FF → fill color pink	dwFillColor
Fillcolor alarm:	fill color if the 'Change color' variable is TRUE	"plc_prg.var_fillcol_a"	var_fillcol_a:=16#FF00FF → alarm fill color pink	dwFillColorAlarm
Framecolor:	frame color	"plc_prg.var_framecol"	var_framecol::=16#FF00FF → frame color pink	dwFrameColor

Framecolor alarm:	frame color if the 'Change color' variable is TRUE	"plc_prg.var_framecol"	var_framecol:=16#FF00FF → alarm frame color farbe pink	dwFrameColorAlarm
Fillflags:	The current inside color configuration can be activated (FALSE) resp. deactivated (TRUE)	"plc_prg.var_col_off"	var_col_off:=1 → the color definition for the fill color will not be regarded, that for the frame remains valid	dwFillFlags
Frameflags:	Display of the frame (solid, dotted etc.)	"plc_prg.var_linetype"	var_linetype:=2; → frame will be displayed as dotted line	dwFrameFlags

Dialog Box for Configuring Visualization Elements (Category Colorvariables)


Regular Element Configuration (#0)

Variables for colors

Fillcolor: .var_fillcol

Fillcolor alarm: .var_fillcol_alarm

Framecolor: .var_framecol

Framecolor alarm: .var_framecol_alarm

Fillflags: prg1.var_coloff

Frameflags: .var_linetype

Motion absolute

In the visualization element configuration dialog box, in the **Motion absolute** category, **X-** or **Y-Offset** fields variables can be entered. These variables can shift the element in the X or the Y direction, depending on the respective variable value. A variable in the **Scale** field will change the size of the element linear to its current value. This value, which is used as scaling factor, will be divided by 1000 implicitely, so that it is not necessary to use REAL-variables in order to get a reduction of the element. The scaling always will refer to the balance point.

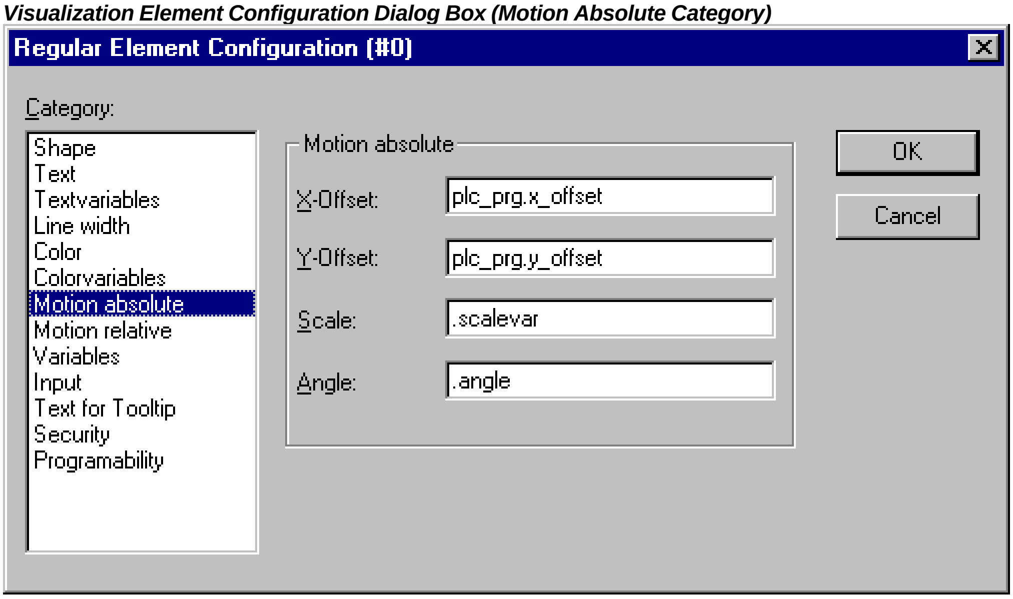
A variable in the **Angle** field causes the element to turn on its turning point, depending on the value of the variable. (Positive Value = Mathematic Positive = Clockwise). The value is evaluated in degrees. With polygons, every point rotates; in other words, the polygon turns. With all other elements, the element rotates, in such a way, that the upper edge always remains on top.

The turning point appears after a single click on the element, and is displayed as a small black circle with a white cross (⊕). You can drag the turning point with a pressed left mouse button.

Note:	In online mode the variables which are set in the 'Motion absolute' dialog will override the values of structure components which additionally might be used to define the same property ('Programability').

In case of multiple definition of an element property consider the specific order of precedence concerning according to which a value might be overwritten in online mode by another.

Visualization Element Configuration Dialog Box (Motion Absolute Category)



X-Offset: plc_prg.x_offset

Y-Offset: plc_prg.y_offset

Scale: .scalevar

Angle: .angle

Motion relative

In the dialog for configuring visualization elements in the **Motion Relative** category, you can assign variables to the individual element edges. Depending on the values of the variables, the corresponding element edges are then moved. The easiest way to enter variables into the fields is to use the Input Assistant (<F2>).

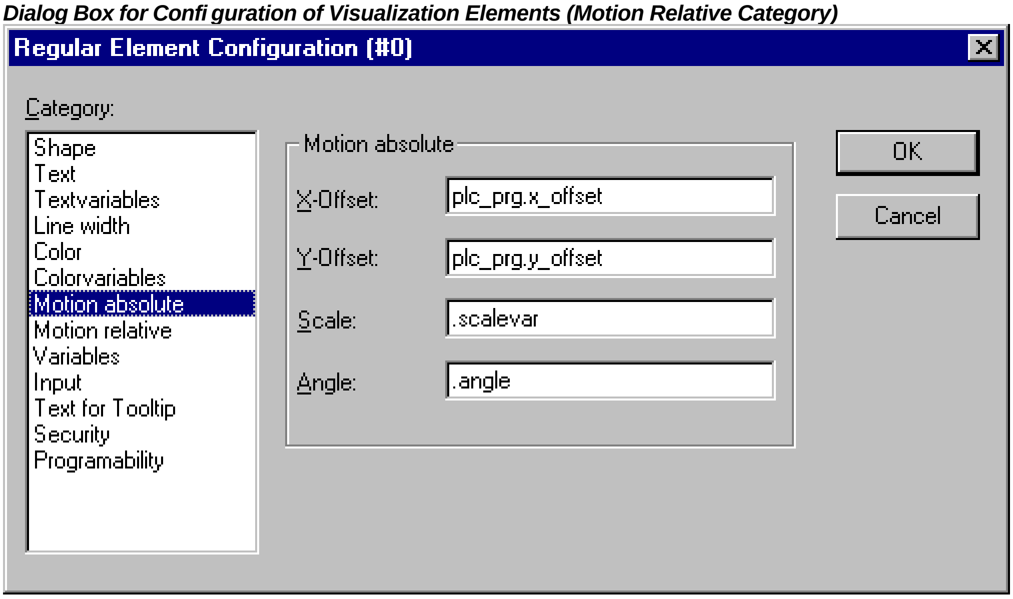
The four entries indicate the four sides of your element. The base position of the corners is always at zero. A new value in the variables, in the corresponding column, shifts the boundary in pixels around this value. Therefore, the variables that are entered ought to be INT variables.

Note: Positive values shift the horizontal edges downward, or, the vertical edges, to the right!

Note: In online mode the variables which are set in the 'Motion absolute' dialog will override the values of structure components which additionally might be used to define the same property ('Programmability').

In case of multiple definition of an element property consider the specific order of precedence concerning according to which a value might be overwritten in online mode by another.

Dialog Box for Configuration of Visualization Elements (Motion Relative Category)



X-Offset: plc_prg.x_offset

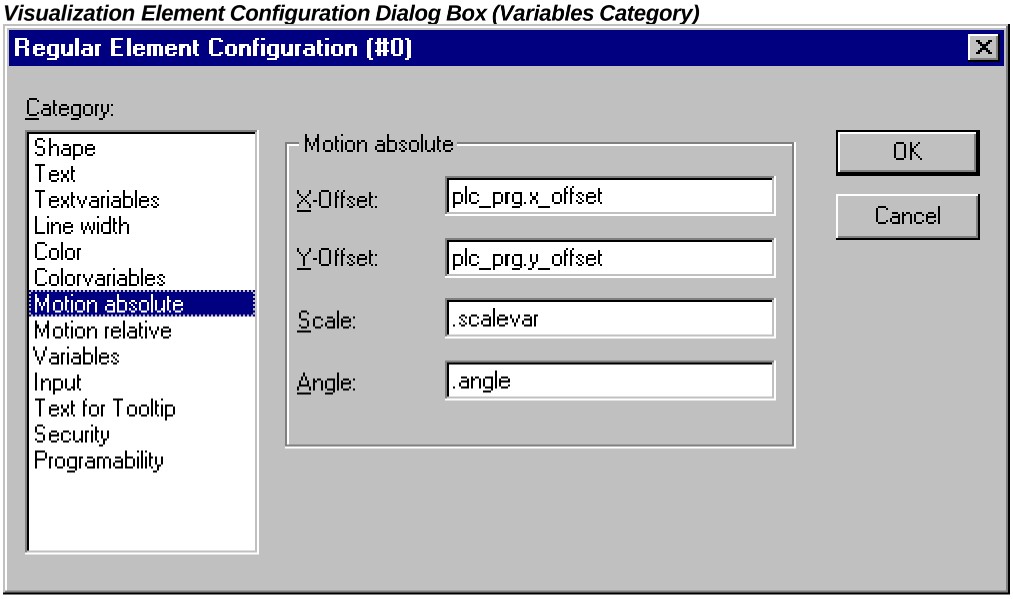
Y-Offset: plc_prg.y_offset

Scale: .scalevar

Angle: .angle

Variables

Visualization Element Configuration Dialog Box (Variables Category)



X-Offset: plc_prg.x_offset
Y-Offset: plc_prg.y_offset
Scale: .scalevar
Angle: .angle

You can enter the variables that describe the status of the visualization elements in the **Variable** category within the dialog box for configuring visualization elements. The simplest way to enter variables in the fields is to use the Input Assistant.

Note: In online mode the variables which are set in the 'Motion absolute' dialog will override the values of structure components which additionally might be used to define the same property ('Programability').

In case of multiple definition of an element property consider the specific order of precedence concerning according to which a value might be overwritten in online mode by another.

You can enter Boolean variables in the **Invisible** and **Change color** fields. The values in the fields determine their actions. If the variable of the **Invisible** field contains the value FALSE, the visualization element will be visible. If the variable contains the value TRUE, the element will be invisible.

Disable input: If the variable entered here is TRUE, all settings of category 'Input' will be ignored.

Change color: If the variable which is defined in this field, has the value FALSE, the visualization element will be displayed in its default color. If the variable is TRUE, the element will be displayed in its alarm color.

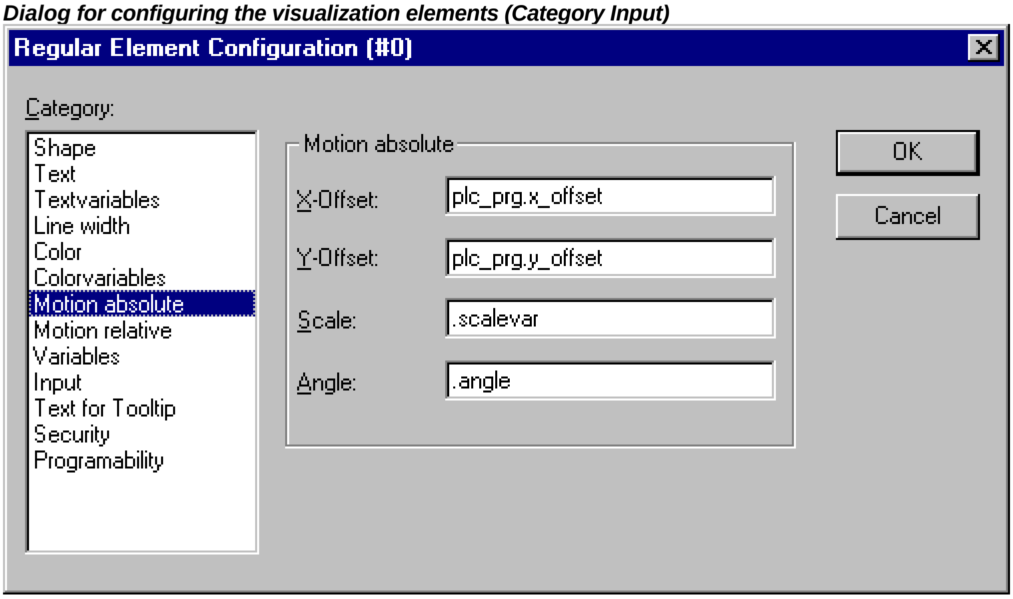
Textdisplay:

- If you have inserted a "%s" in the **Content** field of the **Text** category or if you have included "%s" in the textstring, then the value of the variable which is defined in 'Textdisplay' will be displayed in online mode in the visualization object. "%s" will be replaced by the value.
- If you have inserted resp. included a "%<PREFIX>" in the **Content** field of the Text category ("PREFIX" must be a sequence of letters) eingegeben haben, then the variable resp. the numeric value which is entered here in 'Textdisplay' will be interpreted as an ID, which in combination with the prefix serves as a reference on a text, which is described in a XML-file. This text will be displayed in online mode instead of "%<PREFIX>" in the visualization object. Thus a dynamic modification of the text display is possible. See further information in the description of the dialog 'Settings', category Language.
- If you want to edit the value of the variable in Online mode using the keyboard, you can do this via the 'Text input of variable' 'Textdisplay' in the Input category.

Tooltip-display: Enter here a variable of type STRING whose value should be displayed in a tooltip for the element in online mode.

Input

Dialog for configuring the visualization elements (Category Input)



Motion absolute

X-Offset: plc_prg.x_offset

Y-Offset: plc_prg.y_offset

Scale: .scalevar

Angle: .angle

Toggle variable: If this option is activated, in online mode you will toggle the value of the variables which are located in the input field by each mouse click on the visualization element. You can obtain input assistance for data entry via <F2>. The value of the Boolean variable changes with each mouse click from TRUE to FALSE and then back to TRUE again at the next mouse click, etc.

Tap Variable: If this option is activated, in online mode you can switch the value of the Boolean variable which is located in the input field, between TRUE and FALSE. Place the mouse cursor on the element, press the mousekey and hold it depressed. If option Tap FALSE is activated, the value is set to FALSE as soon as the mouse key is pressed, otherwise it is set to TRUE at this moment. The variable changes back to its initial value as soon as you release the mouse key.

Zoom to Vis...: If this option is activated, you can enter in the edit field the name of a visualization object of the same project to which you want to jump by a mouse-click on the element in online mode. In this case always first the window of the target visualization will be opened before that of the current one will be closed.

The following entries are allowed:

- The name of a visualization object of the current project (see Object Organizer)
- If a visualization reference that contains placeholders is to be jumped to, the placeholders can be directly replaced by variable names or text when called up. For this purpose, conform to the following syntax:
  <Visuname>(<Placeholder1>:=<Text1>, <Placeholder2>:=<Text2>,..., <Placeholder n>:=<Textn>)

Example:
Calling the visualization visu1, whereby the placeholders var_ref1 and var_ref2 used in visu1 are replaced by the variables PLC_PRG.var1 and PROG.var1 respectively:

visu1(var_ref1:=PLC_PRG.var1, var_ref2:=PROG.var1)

- If a program variable of the type STRING (e.g. PLC_PRG.xxx) has been entered instead of a visualization object, then this variable can be used to define the name of the visualization object (e.g. ‚visu1') which the system should change to when a mouse click occurs (e.g. xxx:= ‚visu1).
- If you issue the command „ZOOMTOCALLER" in the **Zoom to vis.** field, a backward jump into the calling visualization is achieved in Online mode by a mouse click on the element, if such a constellation was configured.

Note: The implicit variable **CurrentVisu** (type STRING) describes the name of the currently opened visualization object. For example it can be used in the application to control which visualization should be opened resp. to see which is the currently opened. But this will only work if the names of

the visualization objects are defined in capital letters (see chapter 2.1, Create a visualization object).
Example: `CurrentVisu:='PLC_VISU';`

Execute program: If this option is activated you can enter ASSIGN- or special "INTERN"-commands in the input field, which will be executed in online mode as soon as you perform a mouse-click on the element. Inter alia there is a command "PROGRAM" which can be used to call any executable programs. You get a selection list of possible command entries if you press button "**...**".

Note: This feature especially is important if the visualization will be the only operating interface of a system (pure **operating version**). See a description of the commands: Special input possibilities for the **CoDeSys** operating version.

If you select the **Text input of variable 'Textdisplay', then** in Online mode you will get the possibility to enter an value in this visualization element which will upon pressing **<Enter>** be written to the variable that appears in the **Textdisplay** field of the **Variables** category.

Select in the scroll box which kind of input should be possible later in online mode. **Text:** An edit field will open, where you can enter the value. **Numpad** resp. **Keypad:** A window will open showing an image of the numeric resp. alphabetic keypad, where you can enter a value by activating the appropriate key elements. This might be useful if the visualization must be operatable via a touch screen. The range of valid input values can be restricted by defining a minimum and a maximum value in the edit fields **Min:** and **Max:**.

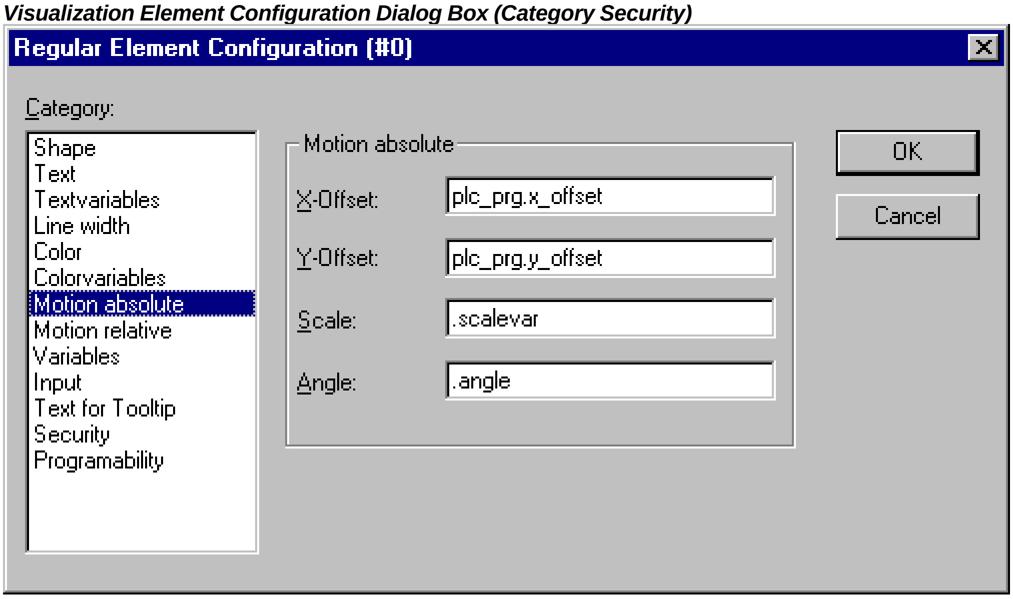
ToolTip

The dialog Text for Tooltip offers an input field for text which appears in a text field as soon as the mouse cursor is passed over the object in online mode. The text can be formatted with line breaks by using the key combination <Ctrl> + <Enter>.

Security

It might be useful that different user groups get different operating possibilities and display of a visualization. This can be reached by assigning different access rights concerning particular visualization elements. You can do this for the eight user groups which are available in CoDeSys (see also 'Project' 'Object' 'Properties' resp. 'Project' 'User Group Passwords'). The access rights can be assigned by activating the appropriate option in the configuration dialog 'Access rights' for a visualization element:

Visualization Element Configuration Dialog Box (Category Security)



X-Offset: plc_prg.x_offset

Y-Offset: plc_prg.y_offset

Scale: .scalevar

Angle: .angle

The access rights for a visualization element and their effect in online mode:

No Access	Element will not be visible

Read Access	Element will be visible but not operatable (no inputs allowed)
Full Access	Element is not visible and not operatable

If you want to assign the access rights also to all other elements of the visualization object, activate option **Apply to all visual elements**.

Note: Please regard, that the access rights which are set for the visualization object in the 'Project''Object' 'Properties' dialog, are independent on those of the particular visualization elements !

Programability

The properties of an visualization element can not only be defined by a static setting or by a "normal" project variable, but also by the components of a structure variable, which is exclusively used for programming visualization elements.

For this purpose the structure **VisualObjectType** is available in the library **SysLibVisu.lib**. Its components can be used to define most of the element properties.

Note: In case of multiple definition of a element property the value of the "normal" project variables will overwrite that of the structure variable and both will overwrite a static definition.

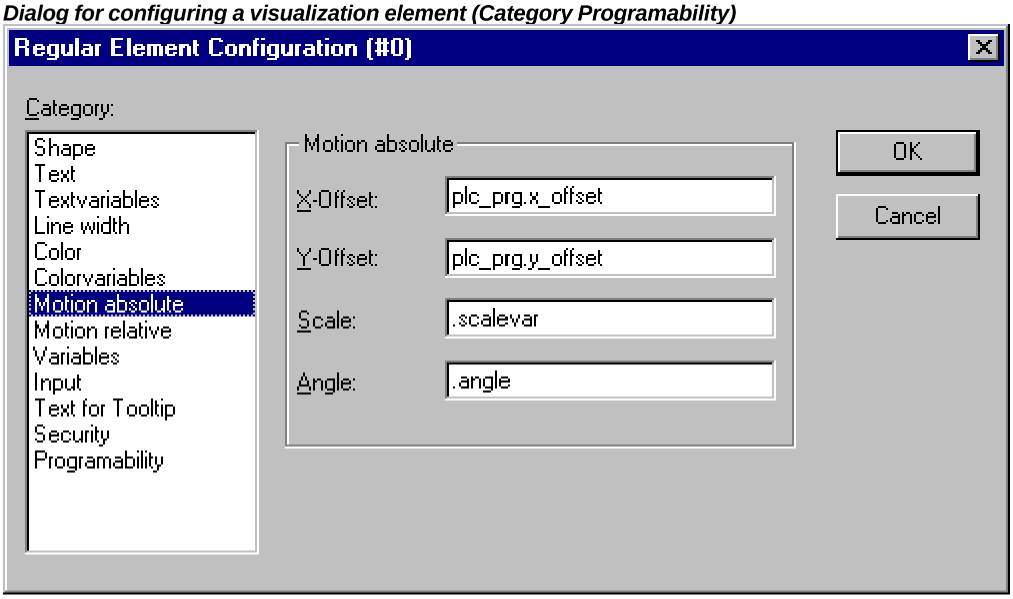
In order to configure the element properties by using a structure variable, do the following:

Open the configuration dialog, category 'Programmability' and enter a new, unique (!) variable name in the field **Object Name:**. For this purpose you must activate the option by a mouse-click in the checkbox. The variable automatically will be declared with type **VisualObjectType**, a structure which is contained in the library **SysLibVisu.Lib**. The declaration is done implicitely and not visible for the user. Make sure that the library is included in the library manager.

After the next compile the newly assigned structure variable will be available in the project. (Hint: Activate the Intellisense functionality 'List components' in the project options, category Editor, in order to get the structure components in a selection list as soon as the variable name followed by a dot is entered).

Example: If you have defined a Object Name "visu1_line" for a visualization element, then you can program the line width of this element by e.g. "visu1_line.nLIneWidth:=4".

Dialog for configuring a visualization element (Category Programability)



Motion absolute

X-Offset: plc_prg.x_offset

Y-Offset: plc_prg.y_offset

Scale: .scalevar

Angle: .angle

The strucure VisualObjectType:

The following table will show you all components of the structure and references to the corresponding items in the different categories of the configuration dialog:

At the beginning of the component name the data type is integrated:
n INT
dw DWORD
b BOOL
st STRING

Component (+Data type)	**Effect**	**Example** **(the Object Name "vis1" has been defined for the element)**	**corresponding entries in configuration dialog:**
nXOffset : INT;	Shift element in X-direction	`vis1.nXOffset:=val2;` (element is set to position X=val2)	- Cat. Motion absolute: X-Offset
nYOffset : INT;	Shift element in Y-direction	`vis1.nYOffset:=22;` (element is set to position Y=val2)	- Cat. Motion absolute: Y-Offset
nScale : INT;	Change of the size	`vis1.nScale:=plc_prg.scale_var;` (element size changes linear with change of value of plc_prg.scale_var)	- Cat. Motion absolute: Scaling
nAngle : INT;	Rotating element around its center	`vis1.anglevar:=15;` (element rotates clockwise by 15)	- Cat. Motion absolute: angle
bInvisible : BOOL;	Element is visible / invisible	`vis1.visible:=TRUE;` (element is invisible)	- Cat. Color: No color inside + No frame color - Cat. Colorvariables: Fillcolor + Framecolor
stTextDisplay : STRING;	Text is displayed in element	`vis1.TextDisplay:='ON / OFF';` element is inscribed with this text	- Cat. Text: entry at 'Content'
bToggleColor : BOOL;	color change when toggling between TRUE and FALSE	`vis1.bToggleColor:=alarm_var;` (As soon as alarm_var gets TRUE, the element gets the color defined via the components dwFillColorAlarm, dwFrameColorAlarm resp. via the settings in category 'Colorvariables' or 'Color'.	- Cat. Input: Toggle variable + - Cat. Variables: Change color
bInputDisabled: BOOL;	if FALSE: Inputs in category 'Input' are ignored	`vis1.bInputDisabled:=FALSE;` (no input is possible for this elementt)	- Cat. Variables: 'Disable Input'
stTooltipDisplay:STRING;	Text of the tooltip	`vis1.stTooltipDisplay:='Switch for .....';`	- Cat. Text for Tooltip: Entry in 'Content:'
dwTextFlags:DWORD;	Text position: 1 left justified 2 right justified: 4 centered horiz. 8 top 10 bottom 20 centered vertically **Note:** Always set a horizontal <u>and</u> a vertical position (addition of values)!	`vis1.dwTextFlags:=24;` (Text will be placed in the center of the element (4 + 20)	- Cat. Text: Horizontal and Vertical options - Cat. Textvariables: Textflags
dwTextColor : DWORD;	Text color (definition of colors see	`vis1.dwTextColor := 16#00FF0000;`	- Cat. Text: Font | Color

	subsequent to this table)	(Text is blue-colored)	- Cat. Textvariables: Textcolor
nFontHeight : INT;	Font height in Pixel. should be in range 10-96	`vis1.nFontHeight:=16;` (Text height is 16 pt)	- Cat. Text: Font | Grad' - Cat. Textvariables: Font heigth
dwFontFlags : DWORD;	Font display. Available flags: 1 italic 2 fett 4 underlined 8 canceled + combinations by adding values	`vis1.dwFontFlags:=10;` (Text is displayed blue and canceled)	- Cat. Text: Schrift | Schriftschnitt - Cat. Textvariables: Fontflags
stFontName : STRING;	Change font	`vis1.stFontName:=’Arial’;` (Arial is used)	- Cat. Text: Schrift | Schriftart - Cat. Textvariables: Fontname
nLineWidth : INT;	Line width of the frame (pixels)	`vis1.nLWidth:=3;` (Frame width is 3 Pixels)	- Cat. Line width
dwFillColor : DWORD;	Fill color (definition of colors see subsequent to this table)	`vis1.dwFillColor":=16#00FF0000;` (Element ist im "Normalzustand" blau)	- Cat. Color: Color | nside - Cat. Colorvariables: Inside
dwFillColorAlarm : DWORD;	Fill color as soon as bToggleColor gets TRUE, see above) (definition of colors see subsequent to this table)	`vis1.dwFillColorAlarm:=16#00808080;` (as soon as Variable togglevar gets TRUE, the element will be displayed grey-colored)	- Cat. Color: Alarm color | Inside - Cat. Colorvariables: Inside Alarm
dwFrameColor: DWORD;	Frame color (definition of colors see subsequent to this table)	`vis1.dwFrameColor:=16#00FF0000`; (Frame is blue-colored)	- Cat. Color: Color | Frame - Cat. Colorvariables: Frame
dwFrameColorAlarm: DWORD;	Fill color as soon as bFrameColor gets TRUE, see above (definition of colors see subsequent to this table)	`vis1.dwFrameColorAlarm:=16#00808080`; (as soon as Variable vis1.bToggleColor gets TRUE, the frame will be displayed grey-colored)	- Cat. Color: Alarm color | Frame - Cat. Colorvariables: Frame Alarm
dwFillFlags: DWORD;	Color, as defined by the color variables, can be displayed or ignored 0 = show color >0 = ignore setting	`vis1.dwFillFlags:=1;` (element gets invisible)	- Cat. Color: No color inside + No frame color - Cat. Colorvariables: Fillflags
dwFrameFlags: DWORD;	Display of frame: 0 full 1 dashed (---) 2 dotted ( ) 3 dash-point (_._._)	`vis1.FrameFlags:=1;` (Frame will be displayed as dashed line)	- Cat. Colorvariables: Frameflags

	4 dash-point-point (_.._..) 8 blind out line		

Defining color values:

Example: e1.dwFillColor := 16#00FF00FF;

A color is entered as a hex number which is composed of the Blue/Green/Red (RGB) components. The first two zeros after "16#" should be set to in each case, to fill the DWORD size. For each color value 256 (0-255) colors are available.

FF Blue component
00 Green component
FF Red component

Example for a blinking visualization element:

Define a global variable 'blink1' of type VisualObjectType in the configuration of a rectangle. In a program of function block the value of a component of the structure can be modified.

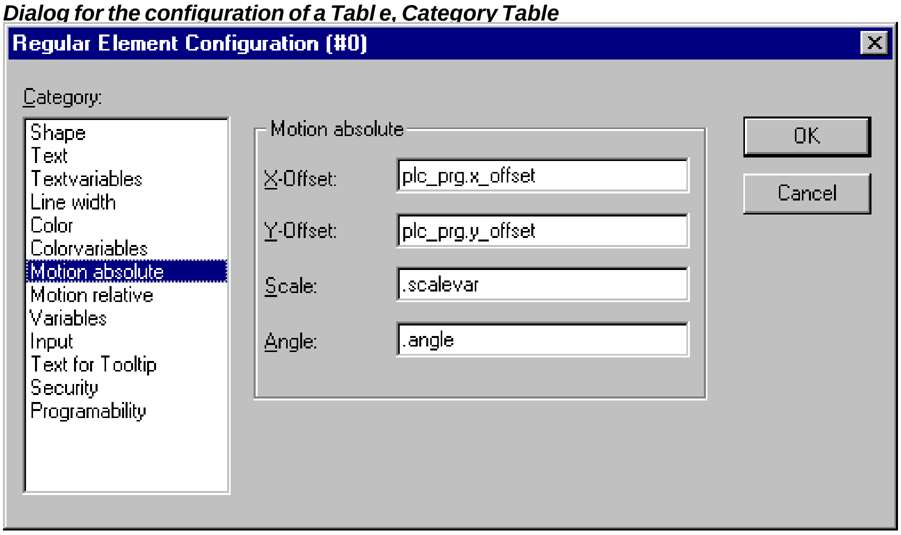
```
PROGRAM PLC_PRG
VAR
 n:INT:=0;
 bMod:BOOL:=TRUE;
END_VAR

(* Blinking element *)
n:=n+1;
bMod:= (n MOD 20) > 10;
IF bMod THEN
 blinker.nFillColor := 16#00808080;  (* Grau *)
ELSE
 blinker.nFillColor := 16#00FF0000;  (* Blau *)
END_IF
```

Table

As soon as a table is inserted for the purpose of visualization of an array, the dialog **Configure Table** will be opend. Besides the categories 'Tooltip' and 'Security' which are also available for other visualization elements, the following categories will be available for configuring display and contents of the table:

Category Table:

Dialog for the configuration of a Table, Category Table

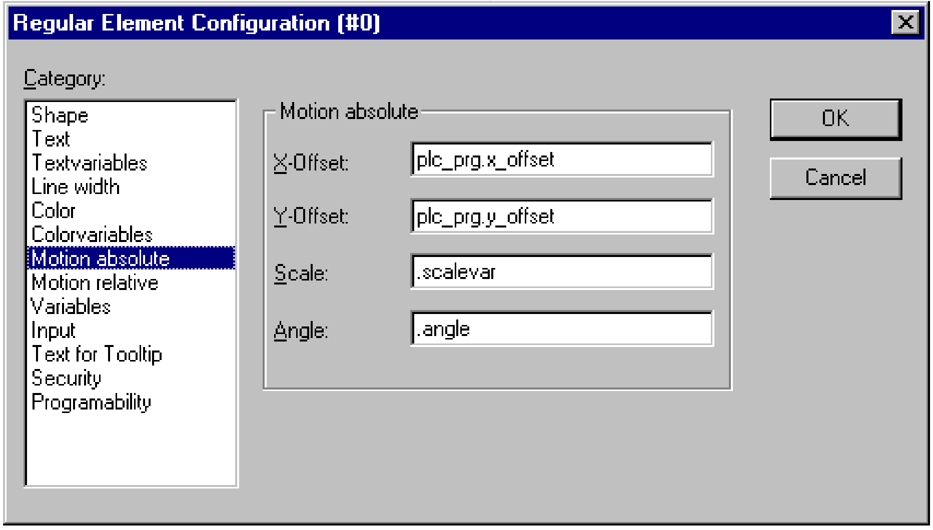


X-Offset: plc_prg.x_offset
Y-Offset: plc_prg.y_offset
Scale: .scalevar
Angle: .angle

Do the following table settings:
Data array: Insert the name of an array which should be visualized in the table. It is recommended to use the input assistant (<F2>) resp. the Intellisense function.
Column header, Line header: Activate these options if you want to get displayed the titles in the table. The line title reflects the array index (first column of the table), the column title can be defined in category 'Columns'.

Category Columns:

Dialog for the configuration of a Table, Category Columns



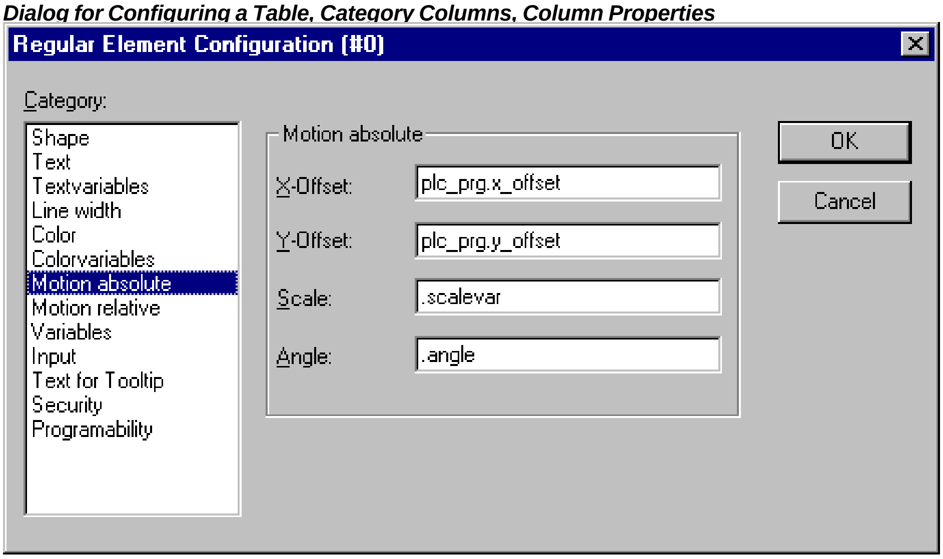
Category: Motion absolute

X-Offset: plc_prg.x_offset
Y-Offset: plc_prg.y_offset
Scale: .scalevar
Angle: .angle

Here you define the **table elements**. In the left window you get al list of all elements, which are handled in the array per index. In case of an array of a structure these would be the structure components.

Using the arrow button **>** you can transfer a selected component from the left window to the right window where you define the set of elements to be displayed in the table. Pressing button **>>** all elements will be transferred at a single blow. In the same manner you can remove elements from an already defined set (**<**, **<<**). In order to modify the default settings concering the display of the table column for one of the elements, perform a double-click on the desired entry in the right part of the window, or press button '**...**' to open the dialog 'Configure columns':

Dialog for Configuring a Table, Category Columns, Column Properties



Category: Motion absolute

X-Offset: plc_prg.x_offset
Y-Offset: plc_prg.y_offset
Scale: .scalevar
Angle: .angle

Editing the column header and the column width:
Initially the edit field **Column header** will contain an automatically created title (e.g. "PLC_PRG.arr1[INDEX].iNo" in case of an array of structure. for the column representing the

structure component "iNo"). which you can change Furtheron the **Column width** (number of characters) can be set.

Editing configuration parameters for all elements of a column:

By default the table fields are displayed as simple **rectangles** and the entries are not editable. If you however activate the option **Use template**, a pre-defined set of parameters can be used, e.g. for the line width, the text input possibilities etc. To edit these parameters the common configuration dialogs as know for rectangles, bitmaps, buttons etc. are available, which you can access by a mouse-click on the button **Edit template**.

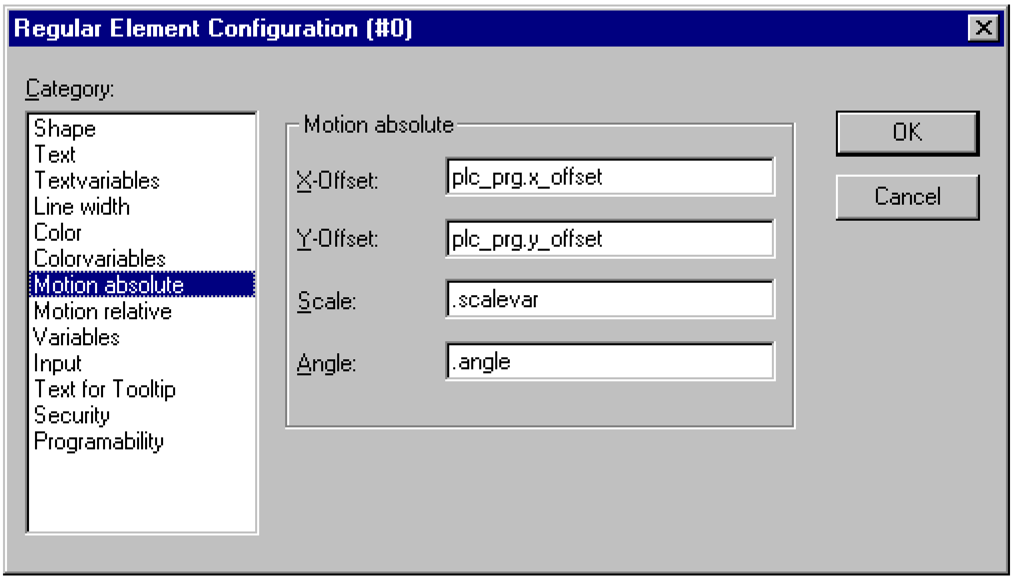
In order to define online actions for all entries of a column, i.e. for all array-elements of one array dimension, use the **placeholder [INDEX]** when entering the array name in the configuration dialogs (e.g. PLC_PRG.arr1[INDEX] like used in the default column header).

Example:

You are visualizing an array "arr1 [0..2] of BOOL" (table with 1 column) and you want, that in online mode by a mouse-click on a table cell the cell gets red-colored and the corresponding array element will be toggled and vice versa.

To reach this activate 'Use template' in the configuration dialog for the column and define the template as follows:
Category 'Input', Action 'Toggle variable': "PLC_PRG.arr1[INDEX].
Category 'Colors': Alarm color red.
Category 'Variables', Action 'Change color': "PLC_PRG.arr1[INDEX].

Category Rows:



X-Offset: plc_prg.x_offset
Y-Offset: plc_prg.y_offset
Scale: .scalevar
Angle: .angle

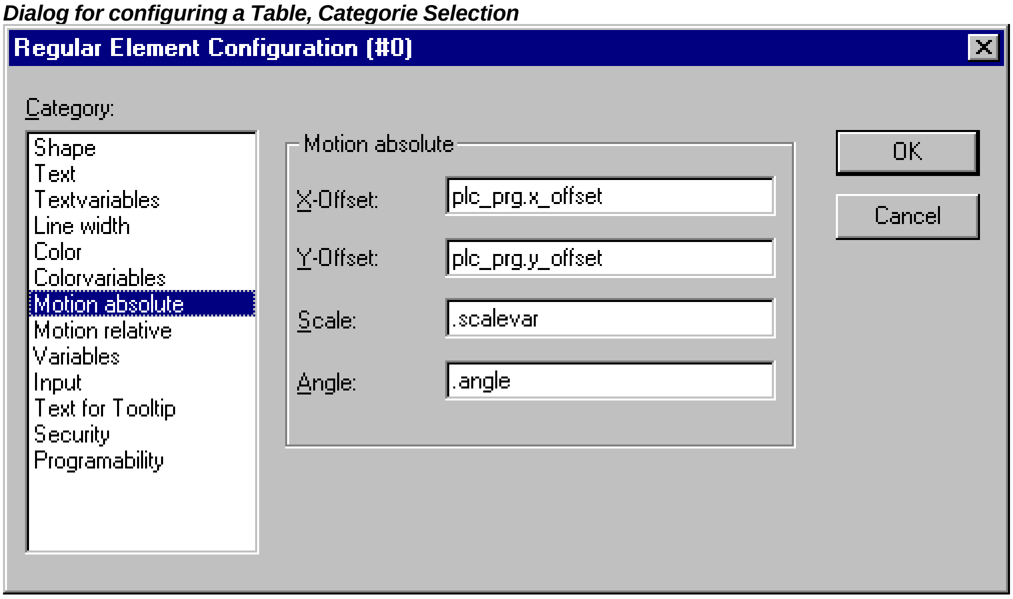
Row height: Insert the desired height in number of pixels.

Category Selection:

Here you can set the following parameters concerning the selection behaviour within the table:

Selection color: Press this button to get the standard color dialog for choosing a color for selected cells.

Dialog for configuring a Table, Categorie Selection



X-Offset: plc_prg.x_offset

Y-Offset: plc_prg.y_offset

Scale: .scalevar

Angle: .angle

Selection type: Define which part of the table will be selected when you perform a mouse-click on one of the table fields in online mode:

Select single cells: Only the cell will be selected.

Select only rows: The whole line will be selected.

Select only columns: The whole column will be selected.

Select rows and columns: The whole column and line will be selected.

Frame around selected cells: A selected cell gets surrounded by a frame.

Variable for selection X, Variable for selection Y: Here you each can enter a project variable, which will indicate the X- resp. Y-Index of the selected table cell.

Example: Create a table element visualizing the array of a structure:

Define the following structure:

```
TYPE strucTab :
STRUCT
iNo: INT;
bDigi : BOOL;
sText:STRING;
byDummy: BYTE;
END_STRUCT
END_TYPE
```

In PLC_PRG define the following array:

```
arr1:ARRAY [1..5] OF strucTab;
```

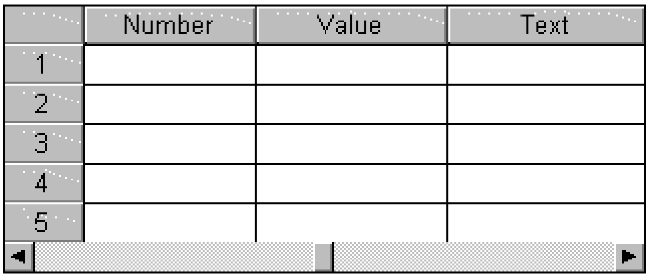
and the following variables:

```
selX:INT;
selY:INT;
```

Create a visualization object and insert a table element. Configure like follows::

Cat. Table: data array: "PLC_PRG.arr1"

Cat. Columns: (Close the dialog which will open with YES) – Transfer the components iNo, bDigi, sText to the right window - In the right window perform a double-click on the first entry (PLC_PRG.arr1[INDEX].iNo) and in the dialog which will open, replace the default title by "Number". Confirm with OK and also define new column titles for the other two entries (e.g. "Value" and "Text"). – In category 'Spec.Table' enter at 'Variable Selection X': "PLC_PRG.selX" and at Variable Selection "Y: PLC_PRG.selY". Activate option 'Frame around selected cells'. Press button 'Selection color' and choose color 'yellow'. Close the configuration dialog with OK. The table element now should be displayed as shown in the following:

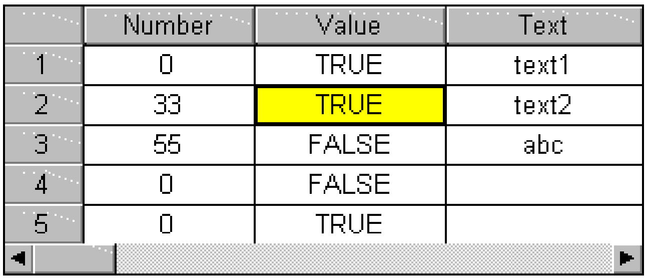


	Number	Value	Text
1			
2			
3			
4			
5			

At the left border the numbers of the array index, at the top the titles of the selected structure components. You can modify the column widths by placing the cursor on the separator between two columns and moving the mouse as soon as the cursor appears as a horizontal double-arrow. .

In online mode the current values of the array elements will be displayed in the table cells. As soon as you select a table cell by a mouse-click, it will get yellow-colored and surrounded by a frame.

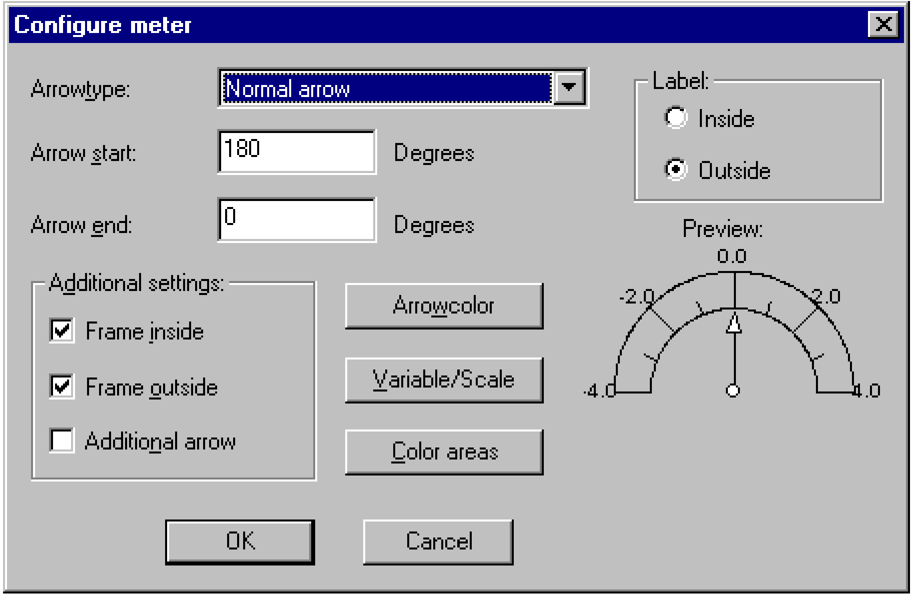
Example:



	Number	Value	Text
1	0	TRUE	text1
2	33	TRUE	text2
3	55	FALSE	abc
4	0	FALSE	
5	0	TRUE	

Meter

Dialog for the configuration of a Meter element



Configure meter

Arrowtype: Normal arrow

Arrow start: 180 Degrees

Arrow end: 0 Degrees

Label:
○ Inside
● Outside

Preview:

Additional settings:
☑ Frame inside
☑ Frame outside
☐ Additional arrow

This dialog will open automatically as soon as you insert a Meter into a visualization object. A **Preview** is part of the dialog, immediately showing how the element will look as a result of the currently set parameters:

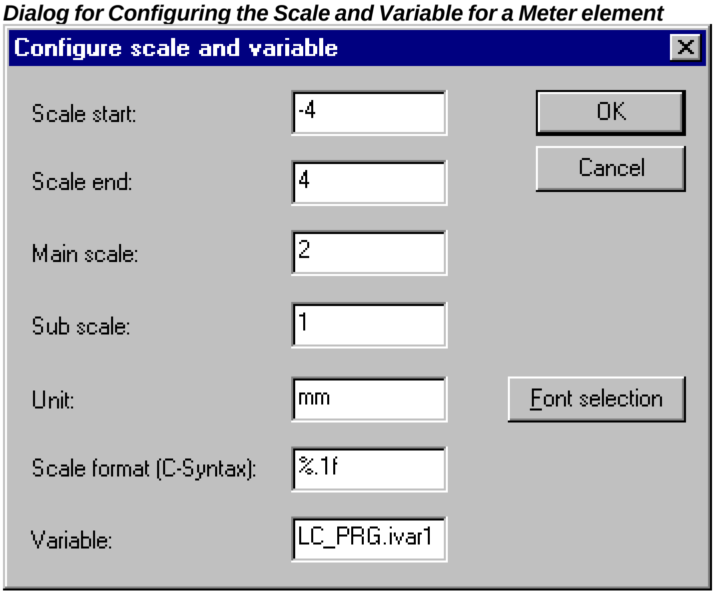
Arrowtype: Define the type of the arrow which will point at the current value on the Meter. Possible types: Normal arrow, Thin arrow, Wide arrow, Thin needle.

Arrow start, **Arrow end**: Here you define the start and the end positions of the scale on a virtual circular arc in ° **Degrees** (angle). (Example: a Start angle of 180° and an End angle of 0° will define a upturned semicircle).

Arrow color: This button opens the standard dialog for choosing a color. Define the color of the pointer.

Variable/Scale: This button opens the dialog **Configure scale and variable**:

Dialog for Configuring the Scale and Variable for a Meter element



Configure scale and variable

Scale start: -4

Scale end: 4

Main scale: 2

Sub scale: 1

Unit: mm

Scale format (C-Syntax): %.1f

Variable: LC_PRG.ivar1

Scale start, **Scale end:** lowest and highest value on the scale, e.g. "-4" and "4".

Main scale: Define which intervals on the scale should be marked "with all", that means which should get a scale pitch and a label. If you insert e.g. "2", each second integer value will be indicated.

Sub scale: In addtion to the main scale (Label + long pitch lines) here you can define a sub-scale which will be displayed as short pitch lines without any labels.

Unit: Define here the scale unit, e.g. "cm" or "sec". The unit is indicated by a label at the origin of the pointer.

Scale format (C-Syntax): According to the C-syntax you can define the display format of the scale labels; see the description concerning Category 'Text'.. Example: If you insert "%1.1f" the scale values will be indicated by a floating point number with one decimal place before and one after the comma (e.g. "12.0")

Variable: Here you can define a variable which is assigned to the pointer position. (e.g. "PLC_PRG.posvar")

Font selection: This button will open the standard dialog for defining the font used in the Meter element.

Color areas: This button opens the dialog **Configure color areas**: Here you can define a separate color for each partition of the scale:

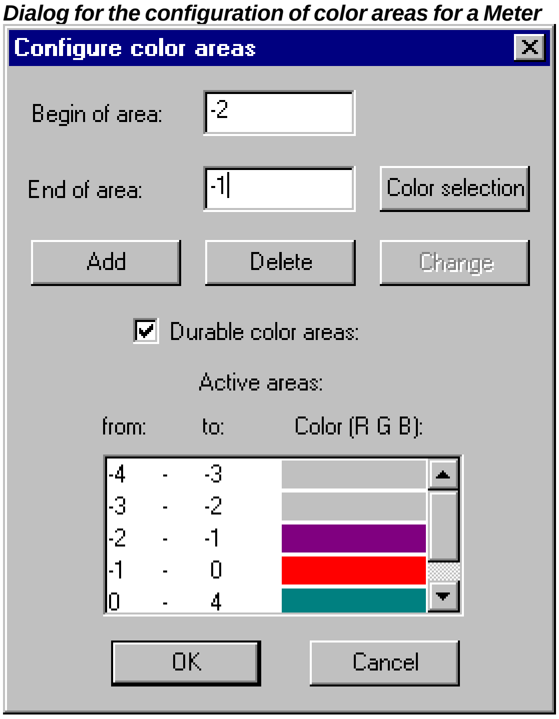
Begin of area, End of area: Insert here the start and end values of the scale partition which should get the color defined in the following:

Color selection: This button opens the standard dialog for choosing a color. Confirm your selection with OK, which will close the dialog, and press button **Add**, whereupon the color and the assigned partition of the scale will be added to the window 'Active areas'. In order to remove an already defined area, select the entry and press **Delete**.

If the option **Durable color areas** is activated, the defined color ranges will be displayed permanently, otherwise in online mode just that partition of the scale will be colored which contains the current value of the respective value.

Label: Depending on which of the options is activated (**inside** or **outside**), the scale labels are placed at the inside or the outside of the circular arc of the scale.

Dialog for the configuration of color areas for a Meter



Begin of area: -2

End of area: -1

☑ Durable color areas:

Active areas:

from:		to:	Color (R G B):
-4	-	-3	⬜
-3	-	-2	⬜
-2	-	-1	🟪
-1	-	0	🟥
0	-	4	🟦

Additional settings:

Frame inside, Frame outside: If one or both of these options is/are activated, an inner or outer frame will be added to the scale arc.

Additional arrow: In addition to the main pointer a little arrow will indicate the current value directly on the scale.

Bar Display

Dialog for the configuration of a Bar Display element

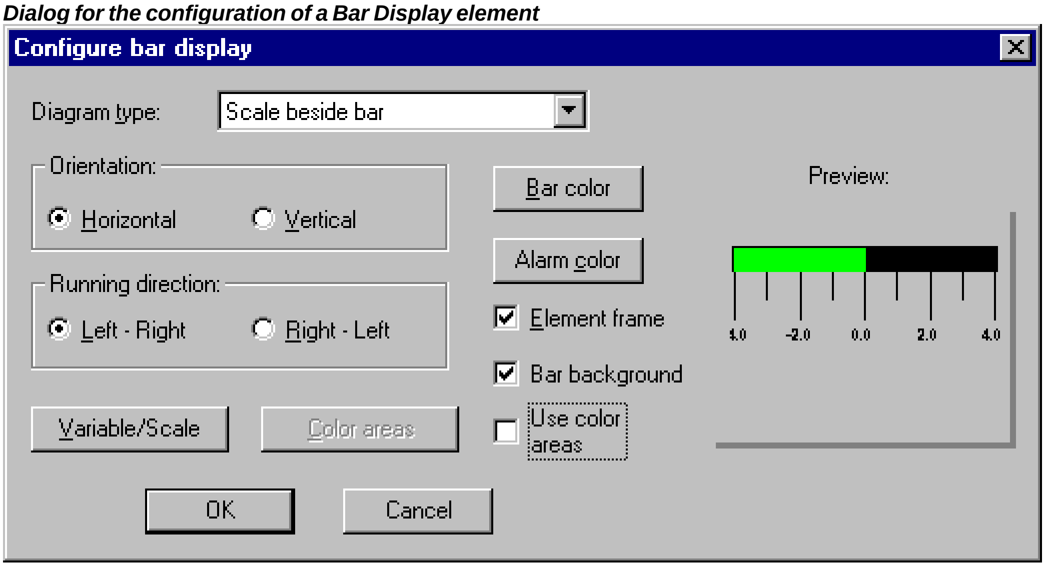


Diagram type: Scale beside bar

Orientation: ● Horizontal ○ Vertical

Running direction: ● Left - Right ○ Right - Left

☑ Element frame

☑ Bar background

☐ Use color areas

This dialog will be opened as soon as you insert a Meter element into a visualization object. A **Preview** is part of the dialog, immediately showing how the element will look as a result of the currently set parameters:

Diagram type: Choose one of the options: 'Scale beside bar', 'Scale inside bar' und 'Bar inside scale'.

Orientation: Define one of the options: **Horizontal** or **Vertical** bar.

Running direction: Choose whether the bar should be elongated corresponding to a growing value of the assigned variable in Left – Right or in Right – Left direction.

Bar color: This button opens the standard dialog for choosing a color. Define a color for the bar in normal state (no alarm). If option 'Use color areas' (see below) is activated, no entries are possible.

Alarm color: This button opens the dialog **Configure alarm**, where you define at which value the bar will be displayed in alarm color and which is the alarm color: Insert the desired limit value in the edit field and activate one of the **Condition**s **greater than** or **lower than**, in order to define whether values higher or lower than the limit value should set off an alarm. Press button **Alarm color** to open the standard color dialog for choosing an alarm color. Close both dialogs with OK in order to confirm the settings and to return to the main dialog for configuring the bar display. If the option 'Use color ranges' (see below) is activated, no entries are possible.

Variable/Scale: This button opens the dialog **Configure scale and variable**, which corresponds to that used for the Meter element.

Element frame: If this option is activated a frame will enclose the bar display.

Bar background: If this option is activated, the whole display range will be indicated by a black bar in the background of the current values' bar, otherwise only the current values' bar will be displayed.

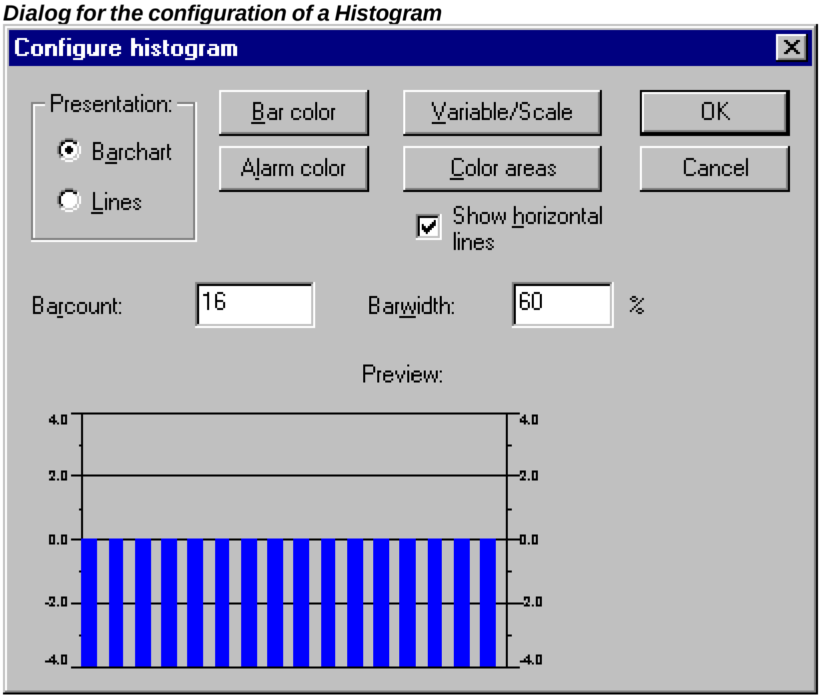
Use color areas: If this option is activated, any settings defined in the dialogs for 'Bar color' and 'Alarm color' (see above) will not be valid. In this case the color area definitions will be used, which have been made in the dialog 'Configure color areas'. This dialog can be opened by pressing button 'Color areas' (see below)

Color areas: This button opens the dialog **Configure color areas** where you can define a separate color for each partition of the scale. These definitions will only be valid if the option 'Use color areas' (see above) is activated. Use the dialog as described for the Meter element.

Histogram

A histogram element can be used to visualize an array. The values of the array elements will be represented by bars or lines side by side, indicating the current values of the element by their heigth.

Dialog for the configuration of a Histogram



Presentation:
● Barchart
○ Lines

☑ Show horizontal lines

Barcount: 16

Barwidth: 60 %

The configuration dialog will be opened as soon as you insert a histogram element into a visualization object. A **Preview** is part of the dialog, immediately showing how the element will look as a result of the currently set parameters:

Presentation: Activate one of the options **Barchart** or **Lines**.

Show horizontal lines: If this option is activated, horizontal lines spanning the diagram will additionally display the scale gradation.

Alarm color: This button opens the dialog **Configure alarm**, where you define at which value the bar will be displayed in alarm color and which is the alarm color: Insert the desired threshold value in the edit field and activate one of the **Condition**s **greater than** or **less than**, in order to define whether values higher or lower than the limit value should set off an alarm. Press button **Alarm color** to open the standard color dialog for choosing an alarm color. Close both dialogs with OK in order to confirm the settings and to return to the main dialog for configuring the histogram.

Variable/Scale: This button opens the dialog **Configure scale and variable**, which can be filled like described for the Meter element.

Color areas: This button opens the dialog **Configure color areas**: Here you can define a separate color for each partition of the scale. See the description of the Meter where the same dialog is available.

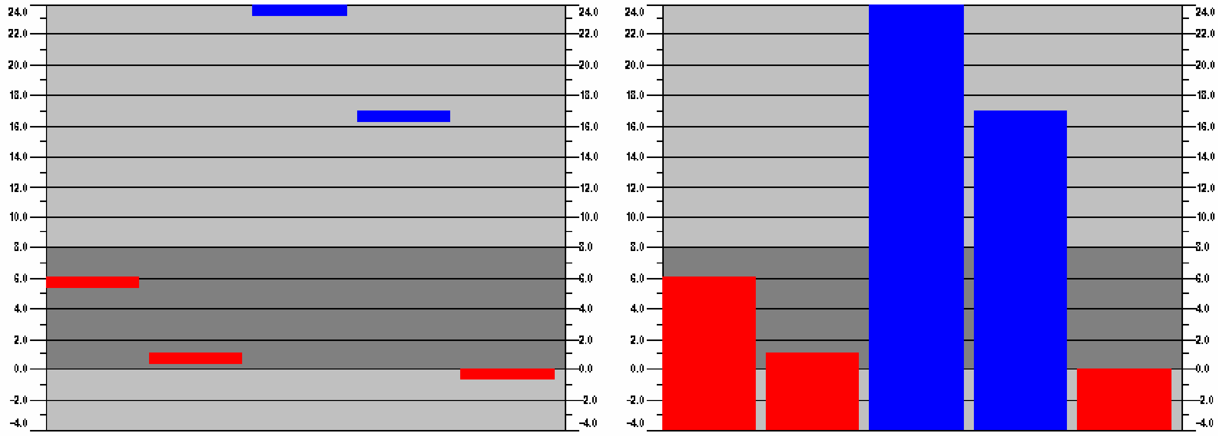
Bar color: This button opens the standard dialog for choosing a color. Define a color for the bar in normal state (no alarm).

Barcount: Insert the desired number of bars (e.g. corresonding to the number of elements of the array).

Barwidth: Define the width of the bars in percent by the total width available for one bar.

Example:

See in the following picture an example of the online display (bars resp. lines) of a histogram which represents an array arr1 [0..4] of INT. The number of bars appropriately was set to "5", the scale start to "–4", the scale end to "24", the main gradation was set to "2", the sub-gradation to "1" and the scale range 0 – 8 has got assigned another color (dark grey) than the rest of the scale range. Furtheron the bars should be displayed alarm-colored (blue) as soon as the value of the corresponding array element exceeds "8". You see the array elements arr1[2] and arr1[3] currently beeing in alarm state:



Bitmap

You can enter the options for a bitmap in the **Bitmap** category within the visualization element configuration dialog box.

Enter the bitmap file and its path in the **Bitmap** field. You can use the **...** button to open the standard Windows Browse dialog box from which you can select the desired bitmap.

All other entries affect the **frame** of the bitmap.

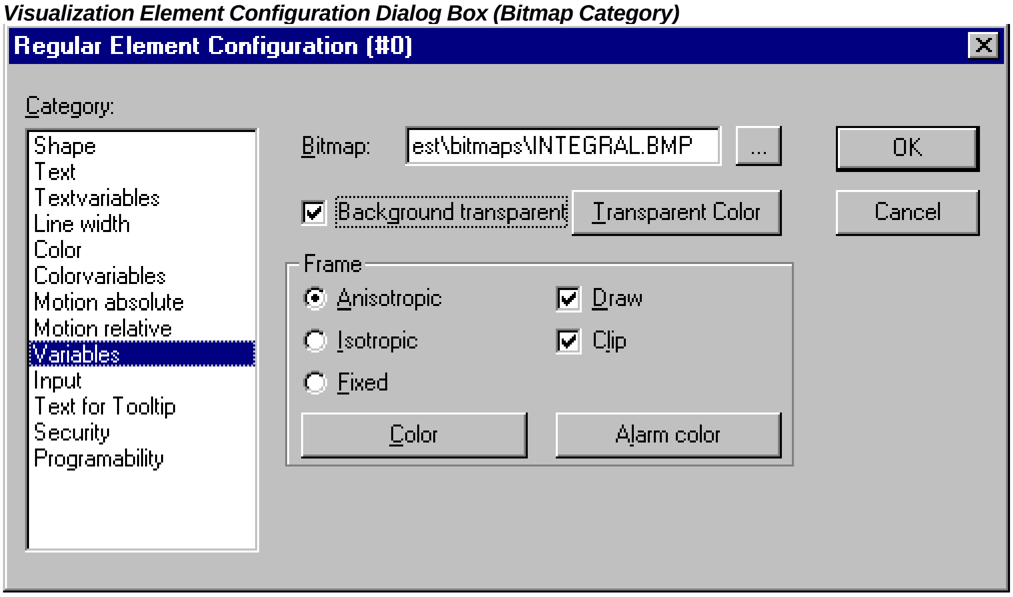
By selecting **Anisotropic**, **Isotropic** or **Fixed** you specify how the bitmap should react to changes in the size of the frame. **Anisotropic** means that the bitmap remains the same size as the frame which allows you to change the height and width of the bitmap independently. **Isotropic** means that the bitmap retains the same proportions even if the overall size is changed (i.e., the relationship between height and width is maintained). If **Fixed** is selected, the original size of the bitmap will be maintained regardless of the size of the frame.

If the **Clip** option is selected together with the **Fixed** setting, only that portion of the bitmap that is contained within the frame will be displayed.

If you select the **Draw** option, the frame will be displayed in the color selected in the **Color** and **Alarm color** buttons in the color dialog boxes. The alarm color will only be used if the variable in the **Change Color** field in the **Variable** category is TRUE.

In the selection list in the lower part of the dialog you can define whether the bitmap should be inserted in the project (Insert) or whether just a link to an external bitmap-file (path as entered above in the 'Bitmap' field) should be created (**Link to file**). It is reasonable to keep the bitmap file in the project directory, because then you can enter a relative path. Otherwise you would enter an absolute path and this might cause problems in case you want to transfer the project to another working environment.

Visualization Element Configuration Dialog Box (Bitmap Category)



Regular Element Configuration (#0)

Bitmap: est\bitmaps\INTEGRAL.BMP

☑ Background transparent

Frame

● Anisotropic

○ Isotropic

○ Fixed

☑ Draw

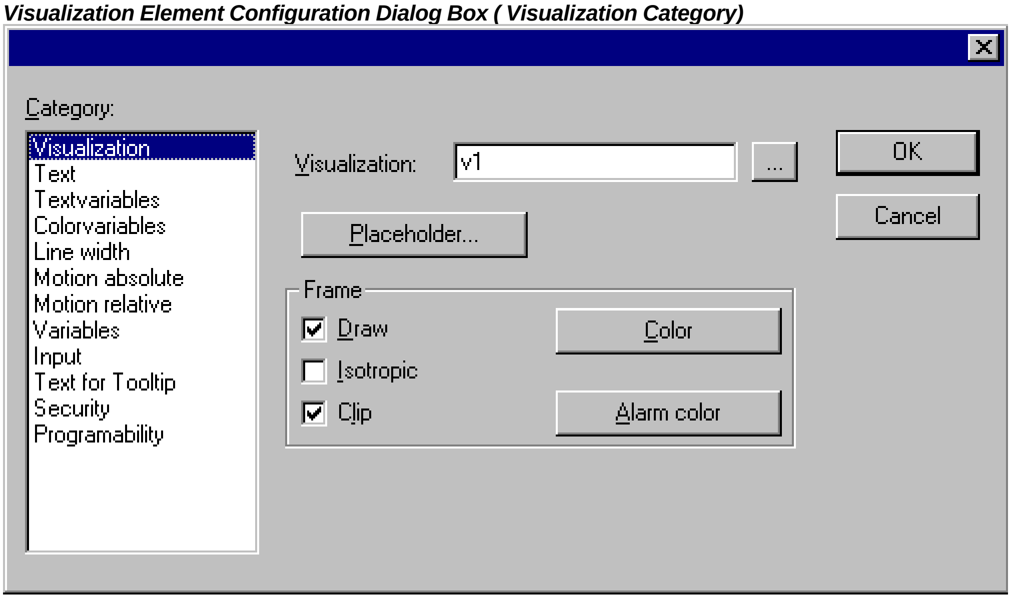
☑ Clip

Visualization

When you insert a visualization as an element in another visualization, you are creating a "reference" of the visualization.

The configuration of this reference can be done in the **Visualization** category within the visualization element configuration dialog box.

Visualization Element Configuration Dialog Box (Visualization Category)



Visualization: v1

Frame

☑ Draw

☐ Isotropic

☑ Clip

Enter the object name for the visualization, which should be inserted, in the **Visualization** field. Use the **...** button to open a dialog box containing the visualizations available in this project. Any visualization may be used with the exception of the current one.

The following entries affect the visualization **Frame**:

- If you select the **Draw** option, the frame will be displayed in the color selected in the **Color** and **Alarm color** buttons in the color dialog boxes. The alarm color will only be used if the variable in the **Change Color** field in the **Variables** category is TRUE.
- If **Isotropic** is selected, the proportions of the visualization will be maintained even if the size changes (i.e., the relationship between height and width will remain the same). Otherwise the proportions can be changed.
- If the **Clip** option is selected in Online mode, only the original portion of the visualization will be displayed. For example, if an object extends beyond the original display area, it will be clipped and may disappear from view completely in the visualization.

The **Placeholder** button leads to the 'Replace placeholder' dialog. It lists in the 'Placeholder' column all the placeholders which had been inserted in the configuration dialogs of the "mother"-visualization and offers in the 'Replacements' column the possibility of replacing these for the current reference with a definite value. Which replacements are possible in a given case depends on whether a value group was predefined in the 'Extras' 'Placeholder list' dialog in the "mother"-visualization. If this is the case, it will be displayed in a combo box for selection. If nothing was pre-defined, double clicking on the corresponding field in the Replacements column opens an editing field which can be filled in as desired.

A further possibility for replacing placeholders in references occurs directly when you define the call of a visualization by an entry into the **Zoom to vis.** option field in the configuration dialog ('Input' category).

Note:	No control of the chronological sequence of replacements is possible! Therefore no placeholders should be replaced with text that also contains placeholders!
Note:	When using placeholders it is no longer possible to check for invalid entries in the configuration of the visualization element immediately upon compilation of the project. Hence the appropriate error messages are first issued in Online mode (...Invalid Watch expression..).
Attention:	**Online behaviour of a visualization reference:** If you insert a visualization and then select and configurate this reference, it will be regarded as a single object and in online mode will react to inputs correspondingly to its configuration. In contrast: if you do not configurate the reference, then in online mode its particular visualization elements will react exactly like those of the original visualization.

Example of an application of the placeholder concept (also see chapter 2.7) :

Instances of a function block can easily be displayed with the help of references of the same visualization. For example, in configuring the visualization visu, which visualizes the variables of function block, one could begin each variable entry with the placeholder FUB (e.g. FUB.a). If a reference from visu is then used (by inserting visu in another visualization or by calling via 'Zoom to vis.'), then in the configuration of this reference the placeholder FUB can then be replaced with the name of the function block instance to be visualized.

This might look like shown in the following:

In the project define a function block containing the following declarations:

```
FUNCTION_BLOCK fu
VAR_INPUT
  changecol : BOOL; (* should cause a color change in the visualization *`)
END_VAR
```

In PLC_PRG define two instances of 'fu':

```
inst1_fu : fu;
inst2_fu : fu;
```

Create a visualization object 'visu'. Insert an element and open the configuration dialog, category 'Variables'. Enter in field 'Change color' the following: "FUB.changecol". Open category 'Input' and enter in field 'Tap Variable' "FUB.changecol". Open category 'Text' and enter"FUB - change color ".

Create another visualization object 'visu1'.

Insert visualization 'visu' twice in 'visu1' (two references of 'visu').

Mark the first reference of 'visu' and open the configuration dialog of category 'Visualization'. Press button 'Placeholder', so that the placeholder list will be displayed. There replace entry 'FUB' by 'PLC_PRG.inst_1'.

Now mark the second reference of 'visu' and (like described for the first one) replace 'FUB' by 'PLC_PRG.inst_2'.

Now in online mode the values of the variables which are used to configure the two instances of 'fu' will be visualized in the corresponding reference of 'visu'.

Of course the placeholder FUB can be used at all places in the configuration of 'visu' where variables or text strings are entered.

Special input possibilities for operating versions

The CoDeSys visualization can target specifically be used with CoDeSys HMI or as Web- or Target-Visualization as a mere operating interface. Then no menus and status and tool bars will be available to the user and no possibility to modify the code.

Thus, when a visualization is created with CoDeSys for the purpose of beeing used as a 'operating version' the principal control and monitoring functions in a project must be assigned to visualization elements thus making them accessible via mouse click or keyboard in Online mode.

See in the following some special input possibilities to configure visualization elements for the purpose of beeing used in CoDeSys HMI. They are available in the configuration dialog for a visualization element:

In category **Input** enter internal commands in the field **Execute program** according to the following syntax (The dialog 'Configure Programs' is available for this purpose):

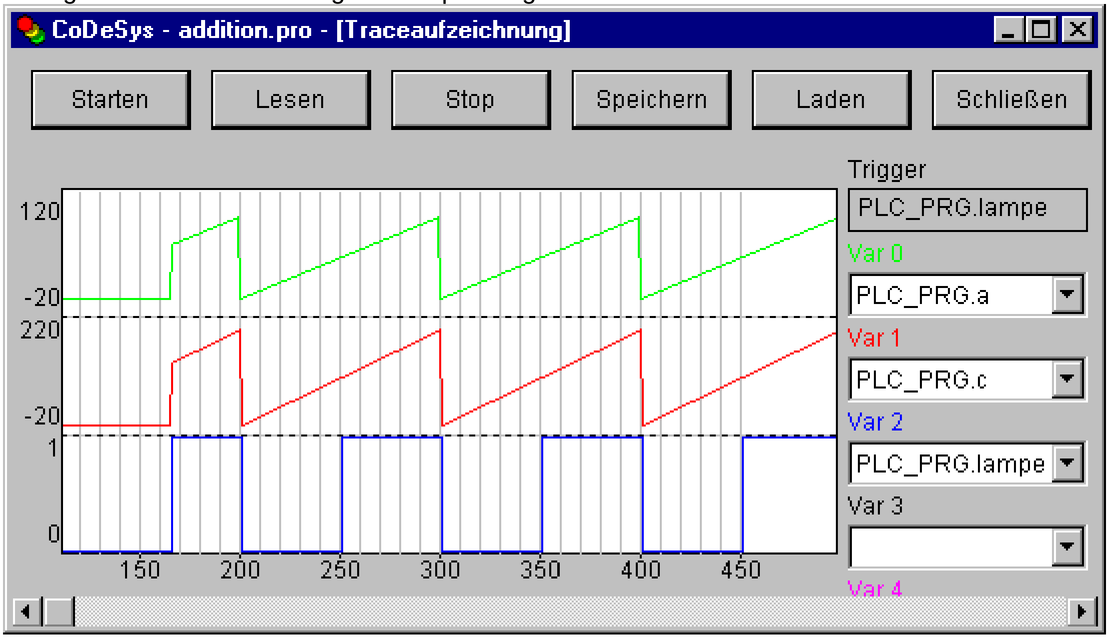
INTERN <COMMAND> [PARAMETER]*

The following table shows the available internal commands. Some of them expect to receive several parameters, which are then entered separated by spaces. Optional parameters are enclosed in square brackets. For those commands which require that a Watch list be specified, a placeholder can be used instead of the direct name. If you enter several commands for one element, these are separated by commas.

Command	**The equivalent in the programming version of CoDeSys**	**Explanation**
ASSIGN <Variable>:=<Expression>	Assignment	A variable or expression gets assigned to another variable. Example: INTERN ASSIGN PLC_PRG.ivar1:=PROG1.ivar+12;
PROGRAM <Pfad executable program> [Path of the file to be opened]	Program call	The program will be executed. Example: INTERN PROGRAM C:\programms\notepad.exe text.txt
LANGUAGEDIALOG	visualization settings	The dialog for visualization settings which includes the category language gets openend.
LANGUAGE <language identifier as used in the currently set language file>	visualization settings, Language	The desired language is set without using the dialog for visualization settings.
DEFINERECEIPT name	Select watch lists	A watch list is selected from the receipt manager which enters your name (name) when the command is given. The variables in this watch list are registered and displayed.
READRECEIPT name	'Read receipt'	In the defined watch list the pre-definition of the variables will be replaced by the current values.
WRITERECEIPT name	'Write receipts'	The name of a watch list of the receipt manager

		is expected. The receipt of this watch list will be written. A previous execution of DEFINERECEIPT is not necessary.
SAVEWATCH	'Save watch list'	The receipt will be read into the current watch list which will be stored in a file. Important: call a previous DEFINERECEIPT to define the current receipt !
LOADWATCH	'Load watch list'+ 'Write receipt'	The standard window ‚File open' appears, from which a previously stored receipt can be selected. This receipt will be immediately written into the controller system.
CHANGEUSERLEVEL	-	A dialog for setting the user group level will open. The eight CoDeSys user group levels are offered for selection.
CHANGEPASSWORD	cp. 'Project' 'User Group Passwords...'	A dialog for changing the user group password will appear.
SAVEPROJECT	‚File' 'Save'	The project will be saved.
EXITPROGRAM	'File' 'Close'	The program will be exited.
PRINT	',File' 'Print'	The current visualization will be printed out online.
HELP <name of help file>	Call of a help file	Depending on which language is set for the visualization, a help file will be called which is entered for that language in the **CoDeSys**.ini file. (see 'Extras' 'settings')
TRACE	Open object trace recording	The window for trace recording (Sampling Trace) will be opened. The menu commands Trace Start, Read, Stop, Save, Load which are available in the full version of **CoDeSys** are available in this window.

Dialog for the trace recording in the operating version



Trigger
PLC_PRG.lampe
Var 0
PLC_PRG.a
Var 1
PLC_PRG.c
Var 2
PLC_PRG.lampe
Var 3
Var 4

2.4.2 Visualization Object, Configure...

Besides the configuration of the individual visualization elements also the visualization object on the whole can get configured. This is possible concerning the settings for frame, language, grid, background etc. as well as the assignment of special hotkey definitions (keyboard usage), which should be valid for exactly one visualization object.

'Extras' 'Settings' / Display, Frame, Grid, Language

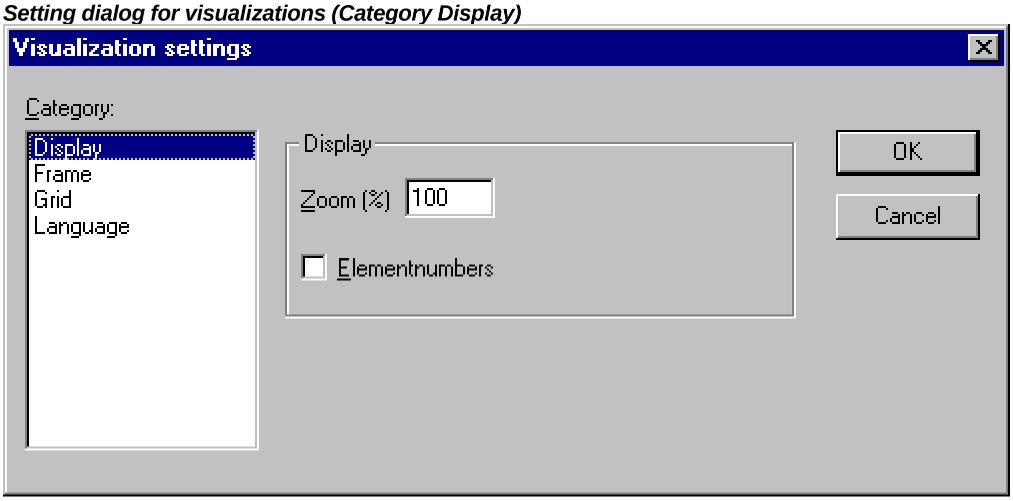
When this command is used, a dialog box will open in which you can make certain settings that affect the visualization.

Note: The categories **Display, Frame** and **Language** also can be edited in the online mode.

Category Display :

Enter a zoom factor into the field Zoom of between 10 and 500 % in order to increase or decrease the size of the visualization display.

Setting dialog for visualizations (Category Display)



Zoom (%) 100

☐ Elementnumbers

Category Frame: If **Auto-scrolling** is selected, the visible portion of the visualization window will move automatically when you reach the edge while drawing or moving a visualization element. If **Best fit in Online mode** is selected, the entire visualization including all elements will be shown in the window in Online mode regardless of the size of the window. When **Include Background Bitmap** is selected, the background bitmap will be fitted into the window as well, otherwise only the elements will be considered.

Category Grid: Define here whether the grid points are **visible** in the offline mode, whereby the spacing between the visible points is at least 10 even if the entered size is smaller than that. In this case the grid points only appear with a spacing which is a multiple of the entered size. Selecting **Active** causes the elements to be placed on the snap grid points when they are drawn and moved. The spacing of the grid points is set in the field **Size**.

Category Language : Here you can specify in which national language the text that you assigned to an element in the **Text** and **Text for Tooltip** options should be displayed. Additionally the option 'Dynamic Texts' allows a dynamic change of the displayed text.

Note: The text display changes only in Online mode!

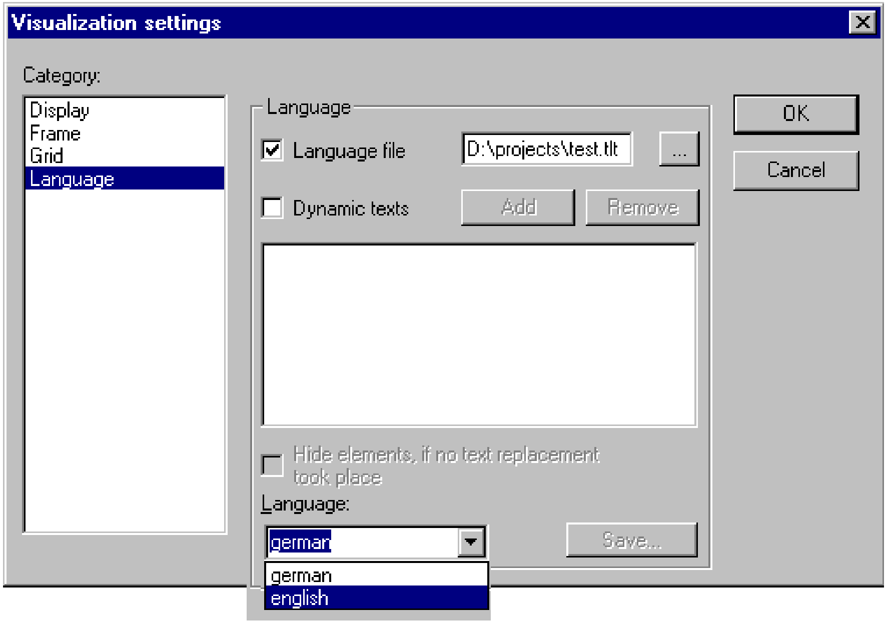
For a non-dynamic switch of the used language a language file can be used (project translation file *.tlt or *.txt, resp. a special visualization language file *vis).

Alternatively **dynamic texts** can be used. Those allow to switch between different text version in the display of a visualization element, inclusively the change of language.

Language file:

In order to select a translation or language file, activate the **Language file** option in the dialog and enter in the input field next to it the appropriate file path, or obtain the help of the standard file opening dialog via the button.

Dialog for selection of a language file for a visualization



☑ Language file D:\projects\test.tlt ...

☐ Dynamic texts

Language: german

german
english

In the selection window under **Language** you can now select among the options German and English in the example shown here.

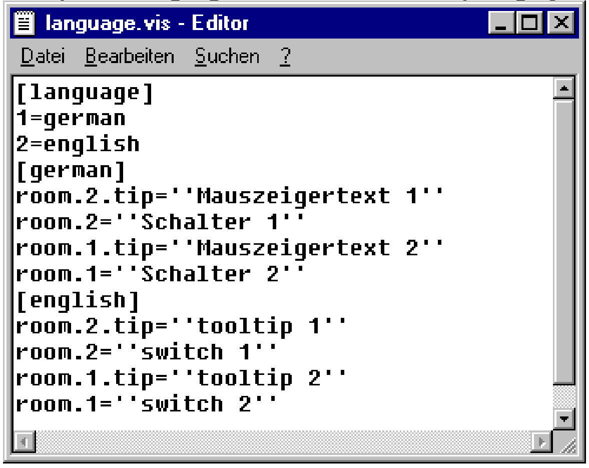
Regarding creating a **translation file *.tlt**, please see 'Project' 'Translate into other anguages'.

For creating a special **language file *.vis** for the visualization, see here (This option is retained for reasons of compatibility with projects created under Version 2.1.):

Creating a *.vis language file:
In order to set up a new *.vis language file for the visualization, proceed as follows:

> Open likewise the Settings Visualization dialog, Language category. Choose option language file. In the associate input field enter where you want to store the file. The extension is .vis. You also can use the dialog 'Open file' by pressing the button. If a language file with the extension .vis is already present, it will be offered to you here.
>
> In the input field next to Language you fill in a keyword for the language which is currently used in the visualization, i.e."german" (or "D"). then press the button Save. A file with the extension .vis will be created, which now can be edited by a normal text editor. For example you can open the file by NOTEPAD:

Example of a language file for a visualisation (Category Language)



```
[language]
1=german
2=english
[german]
room.2.tip=''Mauszeigertext 1''
room.2=''Schalter 1''
room.1.tip=''Mauszeigertext 2''
room.1=''Schalter 2''
[english]
room.2.tip=''tooltip 1''
room.2=''switch 1''
room.1.tip=''tooltip 2''
room.1=''switch 2''
```

You get a list of the text variables for the language currently used in the visualization. It includes a reference to the title of this list, for example "1=german" as reference to the title [german]. You can extend the list by copying all lines, then replacing the German by English text and setting a new title [english]. Beyond the line 1=german you accordingly have to add 2=english.

To view the visualization in one of the prepared languages, open the dialog Language again. In the option field beyond Language now you can choose between german and english (for the example described above).

Dynamic texts:

Dynamic texts allow to switch between different text versions for a visualization elements. The text version is controlled by a Prefix-ID-combination, which is assigned to the text in a XML-file and which can be defined by a project variable (see chapter 2.5.2)

'Extras' 'Select Background Bitmap'

Use this command to open the dialog box for selecting files. Select a file with the extension "*.bmp". The selected bitmap will then appear as the background in your visualization.

The bitmap can be removed with the command 'Extras' 'Clear Background Bitmap'.

'Extras' 'Clear Background Bitmap'

Use this command to remove the bitmap as the background for the current visualization.

You can use the command 'Extras' 'Select Background Bitmap' to select a bitmap for the current visualization.

'Extras' 'Keyboard usage'

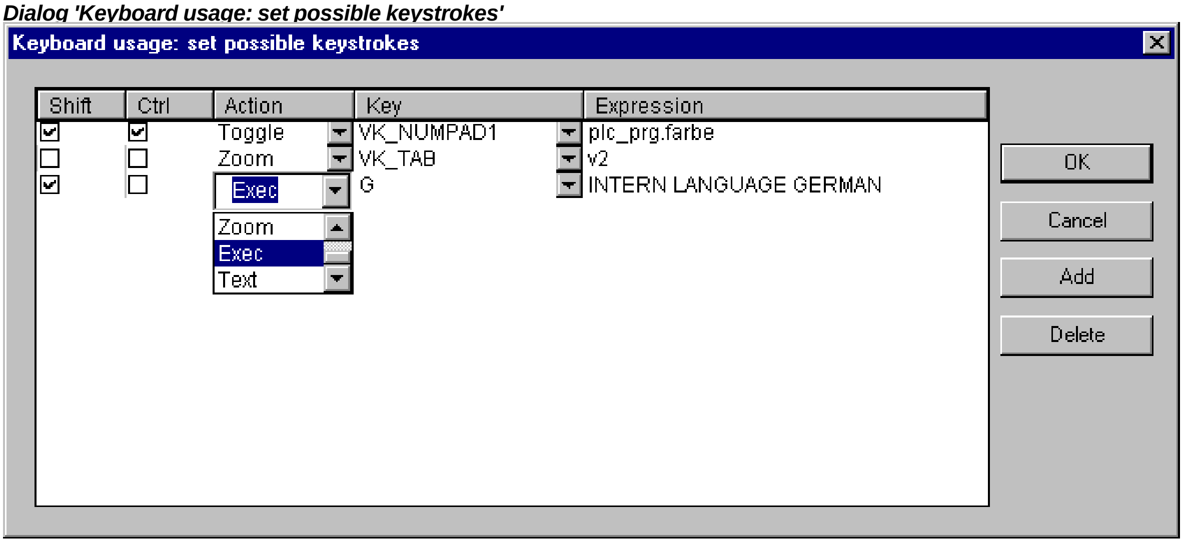
The use of hotkeys can optimize the pure keyboard operation of a visualization.

In the configuration of a visualization object you can define hotkeys which will cause actions like visualization elements do. For example you could define that – if visualization 'xy' is active – in online mode the hotkey <Strg><F2> will stop the program, which also will happen as soon as element 'z' of visu 'xy' gets an input (by mouseclick or via touch screen).

Anyway per default the keys <Tabulator> <Space> <Enter> will work in that way that in online mode each element of a visualization can be selected and activated.

The dialog 'Keyboard usage: set possible keystrokes' can be called in the menu 'Extras' or in the context menu:

Dialog 'Keyboard usage: set possible keystrokes'



Keyboard usage: set possible keystrokes

Shift	Ctrl	Action	Key	Expression
☑	☑	Toggle	VK_NUMPAD1	plc_prg.farbe
☐	☐	Zoom	VK_TAB	v2
☑	☐	Exec	G	INTERN LANGUAGE GERMAN

In column **Key** a selection list offers the following keys to which an action can get assigned:

VK_TAB	Tab-Key
VK_RETURN	Enter-Key
VK_SPACE	Space-Key
VK_ESCAPE	Esc-Key
VK_INSERT	Insert-Key
VK_DELETE	Delete-Key
VK_HOME	Pos1-Key
VK_END	Ende-Key
VK_PRIOR	Bild ()-Key
VK_NEXT	Bild (↓)-Key
VK_LEFT	Arrow-Key (←)
VK_RIGHT	Arrow-Key (→)
VK_UP	Arrow-Key ()
VK_DOWN	Arrow-Key (↓)
VK_F1-VK_F12	Function keys F1 to F12
0-9	Keys 0 to 9
A-Z	Keys A to Z
VK_NUMPAD0 - VK_NUMPAD9	Keys 0 to 9 of the numeric keypad
VK_MULTIPLY	Key* of the numeric keypad
VK_ADD	Key+ of the numeric keypad
VK_SUBTRACT	Key- of the numeric keypad
VK_DIVIDE	Key÷ of the numeric keypad

In the columns **Shift** and **Ctrl** you can add the <Shift>- and/or the <Ctrl>-key to the already chosen key, so that a key combination will result.

In column **Action** you define what should happen as soon as the key (combination) will be pressed. Select the desired action from the list and insert an appropriate expression. See in the following the available actions and valid expressions, corresponding to those which can be set in the configuration dialog of category 'Input':

Action	**Meaning**	**Expression**
Toggle	Toggle variable	Variable, e.g."plc_prg.tvar"
Tap true	Tap variable (set to TRUE)	Program variable, e.g. "plc_prg.svar"
Tap false	Tap variable (set to FALSE)	Program variable, e.g. "plc_prg.xvar"
Zoom	Zoom to Vis.	Name of the visualization object to which you want to jump, e.g. "Visu1"

Exec	Execute program	Name of the executable file, e.g. "notepad C:\help.txt" (Notepad will start and open the file help.txt)
Text	Text input of variable 'Textdisplay'	Number of the element for which the text input is to be configured, e.g. "#2" (Display of element numbers can be switched on in 'Extras' 'Settings'; also see 'Elementlist...')

In column **Expression** you must enter – depending on the type of action – either a variable name, a INTERN-command, a visualization name of a text string, exactly like you would do in the configuration dialog of category 'Input' for the corresponding visualization element.

Use button **Add** to add another empty line at the end of the table. Use the **Delete** button to remove the line where the cursor is positioned currently. **OK** resp. **Cancel** will save resp. not save the done settings and close the dialog.

The keyboard usage can be configured separately for each visualization object. Thus the same key (combination) can start different actions in different visualization.

Example:

The following key configurations have been done for the visualizations VIS_1 and VIS_2:

VIS_1:

Shift	Ctrl	Action	Key	Expression
x		Toggle	A	PLC_PRG.automatic
	x	Zoom	Z	VIS_2

VIS_2:

Shift	**Ctrl**	**Action**	**Key**	**Expression**
		Exec	E	INTERN LANGUAGE DEUTSCH
	x	Zoom	Z	PLC_VISU

If you now go online and set the focus to VIS_1, then pressing <Shift><A> will cause that variable PLC_PRG.automatic will be toggled. <Ctrl><Z> will cause a jump from Visu1 to VIS_2.

If VIS_2 is the active window, pressing key <E> will cause that the language within the visualizatoin will switch to German. <Ctrl><Z> here will cause a jump to visualization PLC_VISU.

2.5 Dynamic Texts

2.5.1 Configuration

You can control dynamically which text will be displayed in a visualization element in online mode by using prefix-ID-combinations, each pointing to another text defined in a XML-file (see chapter 2.5.2). Prefix and ID must be defined in the configuration dialogs of the visualization element, where the ID can be dynamically defined by using a project variable. The XML-file describing the assignment to a text, must be linked to the project. This is done in the visualization settings dialog.

The XML-file must have a certain format, the particular texts are listed below the according Prefix-ID-combination and a language code is added. Thus the text display not only can switch between different contents but also between different languages. The desired language can be selected as described above for the Language file in the 'Settings' dialog, category 'Language':

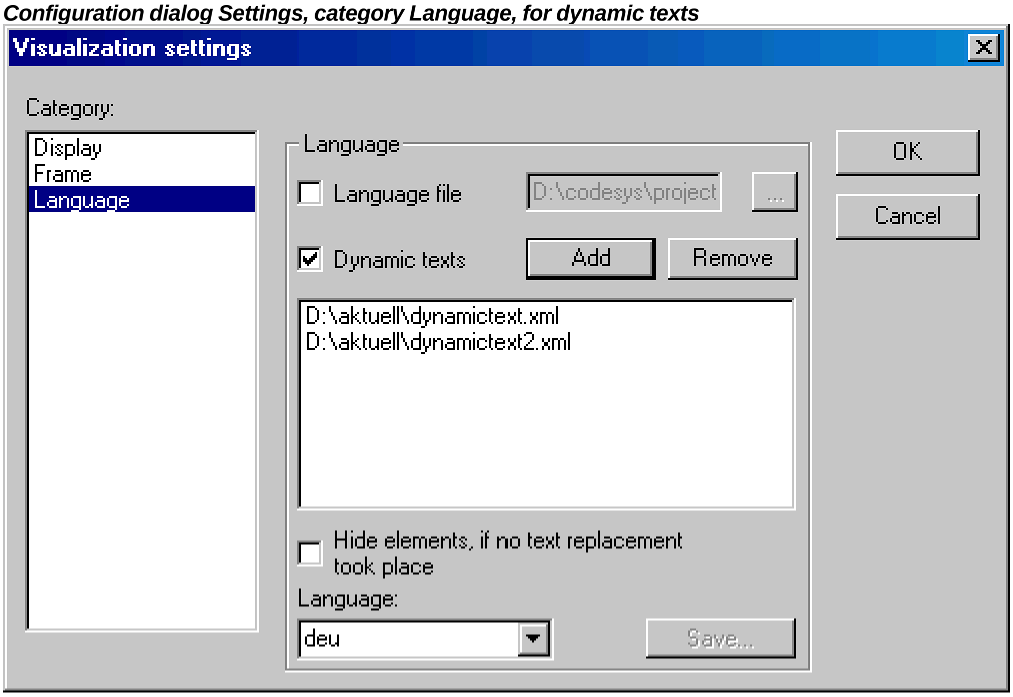
So in order to get a dynamic text display, the following entries have to be made in the different configuration dialogs of a visualization:

1. <u>Link the XML-file(s): Dialog 'Settings' category Language:</u>

   Activate option **Dynamic texts** and press button **Add**, in order to link one or several XML-files, which are available on your system, to the project. The selected files will be listed in the window below the button. Press button **Delete** if you want to remove a selected file from the list. If you want to get displayed just those visualization elements, for which a dynamic text replacement is done, then activate option **Hide elements, if no text replacement took place**.

Selecting one of the language identifiers offered in the selection list at field Language will cause the display of that text versions (for the corresponding prefix-ID-combination) which are marked with that language identifier in the XML.-file.

Configuration dialog Settings, category Language, for dynamic texts



☐ Language file D:\codesys\project

☑ Dynamic texts

D:\aktuell\dynamictext.xml
D:\aktuell\dynamictext2.xml

☐ Hide elements, if no text replacement took place

Language: deu

2. <u>Define the ID in configuration dialog Variables' in field 'Textdisplay'</u>:

   Enter here a value (number) resp. a project variable which should define the ID of a text (as used in the XML-file).

3. <u>Define the text format in configuration dialog 'Text':</u>

   In the Content field, insert a placeholder "%<PREFIX>" at that position of the text, where you want to get displayed a dynamic text in online mode. Instead of "PREFIX" you can enter any desired string. See the description for the 'Text' configuration dialog.

For each prefix-ID-combination, which is found in a linked XML-file, the assigned text will be displayed in the visualization element in online mode. If no appropriate entry is found, no replacement will be done.

2.5.2 XML-File for Dynamic Texts

For use of dynamic texts please see in chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** the descripton of dialog 'Settings', category For a description how to use dynamic texts in the visualization see 'Settings', category Language. The basic file must be available in XML format (<file name>.xml). In this file texts are assigned to identifiers (which are a combination of a prefix and a ID), which can be entered in the configuration of a visualization element.

The file lists all assignments in a list, which must follow the following format: The list must start with <text list> and end with </text list>.

A text entry which is assigned to a Prefix-ID-combination must contain the following lines:

<text prefix>= "<PREFIX> id="<ID>"	"PREFIX" corresponds to the <PREFIX> used in the visualization element configuration (category Text); "ID" corresponds to the entry in category 'Variables', Textdisplay
<Sprachkürzel> <!CDATA[<TEXT>] </Sprachkürzel>	Use any string as an 'language' identifier (e.g. "english"). This identifier will be displayed in the 'Settings' dialog , category Language of the visualization element in the selection list at 'Language'; instead of "TEXT" insert any text which then will be displayed instead of the above defined ID-prefix-combination in the visualization element.
</text>	Abschluss der Definition

For each prefix-ID-combination at least for 1 language a text entry must be available.

e.g. <deu> in the file example shown below will indicate the start of the german version of a text, </deu> terminates the text.

Dynamc texts on the one hand can serve to display Texts in different languages, but of course on the other hand they also can be used to change the content of a text (same language) display dynamically.

Example:

Define in PLC_PRG a variable ivar of type INT.

Configure a visualization element as follows:

Enter in category Text in the Content field: "Achtung, %<ABC>"

Enter in category Variables at Textdisplay: PLC_PRG.ivar

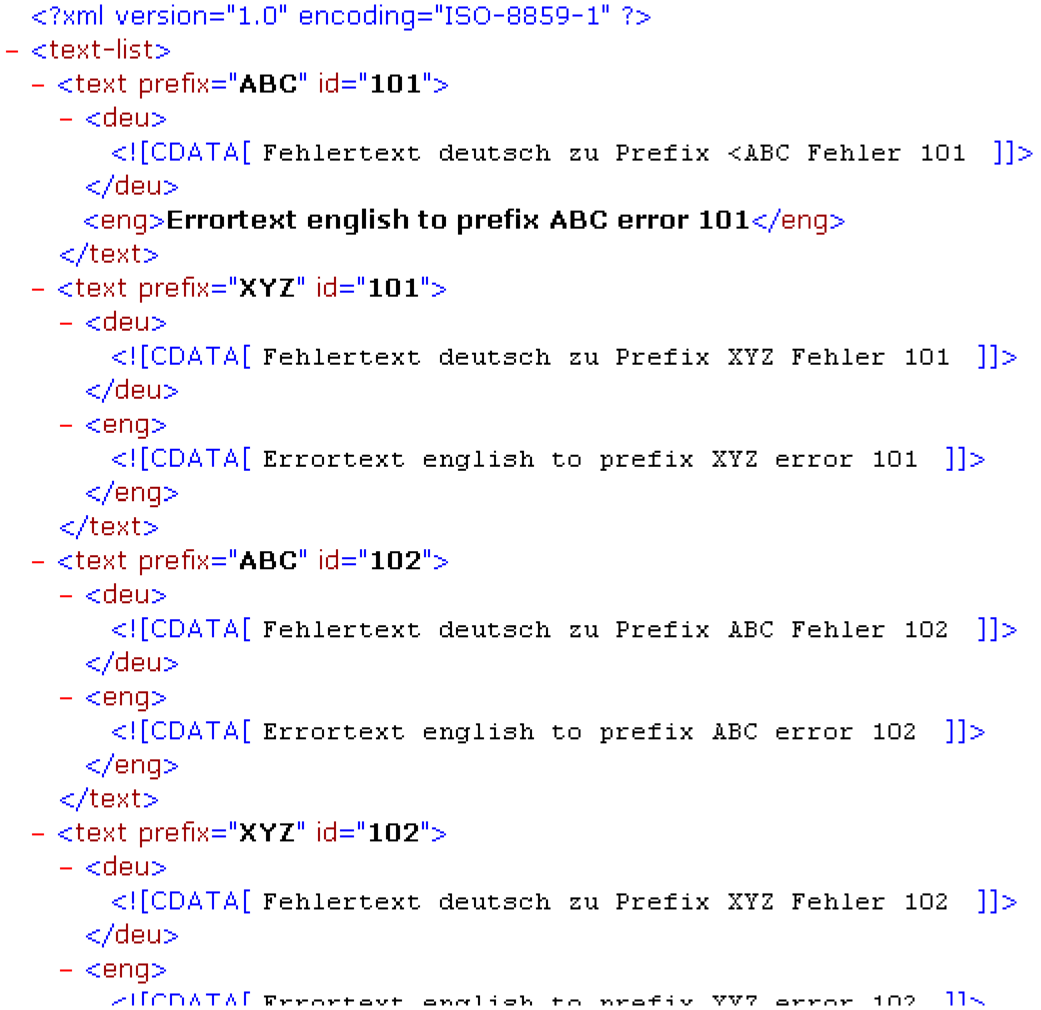
Open dialog Settings, category Language: Activate option Dynamic texts, add the above shown file dynamictext1.xml, which you have available on your computer, to the file list.

Go online with the project. Set the language in the visualization settings to "deu".

Set PLC_PRG.ivar to 102. In the visualization element the following text should be displayed: "Fehlertext deutsch zu Prefix ABC Fehler 102". Set PLC_PRG.ivar to 101. The text should change to "Fehlertext deutsch zu Prefix ABC Fehler 102".

(The change of language also could be managed by using the INTERN command 'LANGUAGE' by another visualization element.)

Example: Part of an XML-Datei for dynamic texts:



```
  <?xml version="1.0" encoding="ISO-8859-1" ?>
- <text-list>
  - <text prefix="ABC" id="101">
    - <deu>
        <![CDATA[ Fehlertext deutsch zu Prefix <ABC Fehler 101  ]]>
      </deu>
      <eng>Errortext english to prefix ABC error 101</eng>
    </text>
  - <text prefix="XYZ" id="101">
    - <deu>
        <![CDATA[ Fehlertext deutsch zu Prefix XYZ Fehler 101  ]]>
      </deu>
    - <eng>
        <![CDATA[ Errortext english to prefix XYZ error 101  ]]>
      </eng>
    </text>
  - <text prefix="ABC" id="102">
    - <deu>
        <![CDATA[ Fehlertext deutsch zu Prefix ABC Fehler 102  ]]>
      </deu>
    - <eng>
        <![CDATA[ Errortext english to prefix ABC error 102  ]]>
      </eng>
    </text>
  - <text prefix="XYZ" id="102">
    - <deu>
        <![CDATA[ Fehlertext deutsch zu Prefix XYZ Fehler 102  ]]>
      </deu>
    - <eng>
        <![CDATA[ Errortext english to prefix XYZ error 102  ]]>
```

2.6 Calling up language-dependent Online Help via a visualization element

The calling of a different Help file with a visualization element can be tied in with the language currently entered for the visualization. For this purpose, the command INTERN HELP must be entered for this element in the 'Configure element' dialog at the location 'Execute program', and a [Visu-Helpfiles] section must be present in the **CoDeSys** .ini file. Below this, the corresponding help files must be assigned to the languages available for selection in the visualization: e.g.:

```
[Visu-Helpfiles]
German=C:\PROGRAMME\HELP_D.HLP
English=C:\PROGRAMME\HELP_E.HLP
```

2.7 Placeholder Concept

At each location in the configuration dialog at which variables or text are entered, a **placeholder** can be set in place of the respective variable or text. This makes sense if the visualization object is not to be used directly in the program, but is created to be inserted in other visualization objects as an "instance". When configuring such an **Instance**, the placeholders can be replaced with variable names or with text.

See in chapter 2.4.2 the paragraph describing the configuration of a visualization which has been inserted in another visualization. There you also find an example of how to use placeholders.

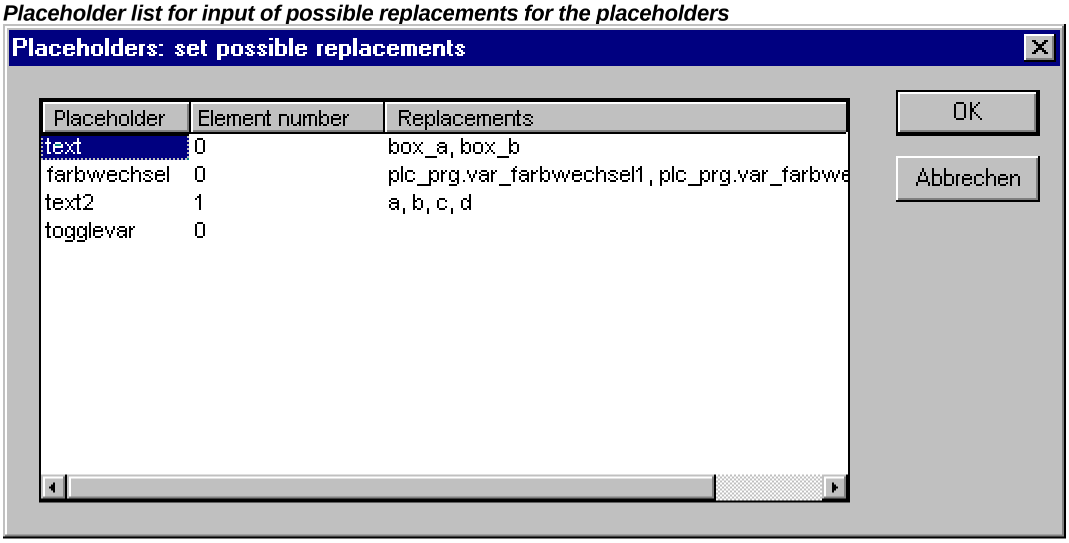
Any string enclosed in two dollar signs ($) is a valid placeholder (e.g. $variable1$, variable$x$). For each placeholder a „value group" can be defined as an input specification in the 'Placeholder list' dialog (called from 'Extras' 'Placeholder list'). With one of these values you can replace the placeholder when configuring an instance of the visualization object. A placeholder list will be available in the instance to do this replacements.

'Extras' 'List of Placeholders...'

This list is used at two places in CoDeSys: to manage placeholders and to configure them:

- primarily you use the list when configuring a visualization object, which later should be inserted, which means instanced, in other visualization(s). For this reason you will use placeholders instead of or additonally to variables and strings in the configuration dialogs. You can open the dialog 'Placeholders' by the command 'List of Placeholders' in the 'Extras' menu or in the context menu. The list shows three columns:

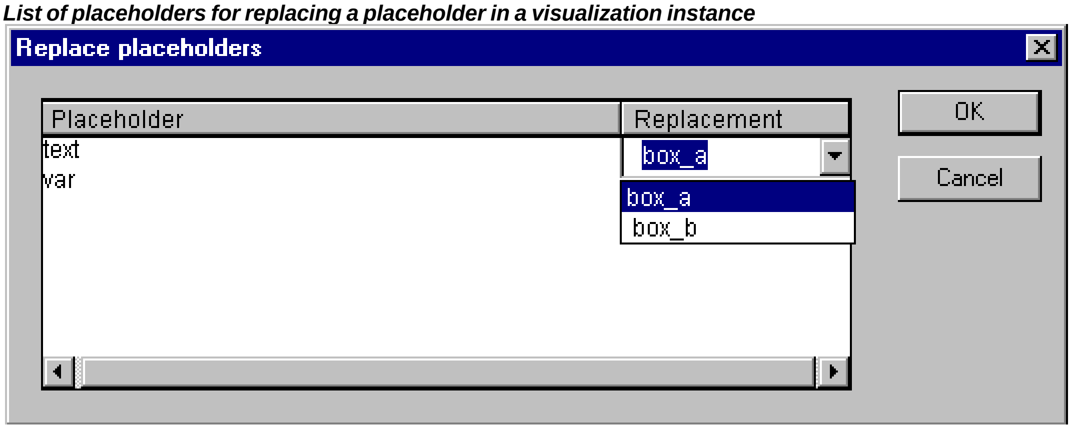
Placeholder list for input of possible replacements for the placeholders



Placeholder	Element number	Replacements
text	0	box_a, box_b
farbwechsel	0	plc_prg.var_farbwechsel1, plc_prg.var_farbwe
text2	1	a, b, c, d
togglevar	0	

- Column **Placeholder** will list all placeholders, which are currently used in the configuration of the visualization object. Column **Element number** shows the elements which contain a placeholder. In column **Replacements** you can enter one or several strings (text, variable, expression) which you want to get available later when replacing a placeholder during the configuration of an instance of the visualization object. The elements of the selection must be entered separated by commas. If no or an impossible replacement string is specified, then the placeholder can be replaced with any desired text later during the configuration of the visualization's reference.
- later you use the list of placeholders when **configuring an instance** of the above mentioned visualization object, that means after this object has been inserted (as a 'reference') in another visualization by the command 'Insert' 'Visualization'. For this purpose do the following to open the dialog: Select the inserted visualization, execute command 'Extras' 'Configure' and press button 'Placeholders' in Category 'Visualization'. In this case the dialog will only contain two columns:

List of placeholders for replacing a placeholder in a visualization instance



- Column **Placeholder** – like described above – shows all placeholders which have been defined for the primary visualization object. If additionally a selection of possible replacements had been defined, this list will now be available in column 'Replacement'. Select one of the entries to replace the placeholder in the present instance. If no replacements have been pre-defined then you can manually enter an expression or variable. For this purpose perform a mouse-click on the field in column **Replacement** to open an editor field.

2.8 Visualization in Online Mode...

Regard the following items concerning a visualization in online mode:

- Order of evaluation:
  - Dynamically defined element properties (by variables) will overwrite the (static) base settings defined by options in the configuration dialogs.
  - If an element property is defined by a "normal" project variable as well as by the component of a structure variable (Programability), then primarily the value of the project variable will be regarded.
  - A visualization can be configured in that way that in online mode it can be operated solely by inputs via keyboard. This is an important feature especially for using the visualization with CoDeSys HMI, as Target- or as Web-Visualization.
- The configuration settings for Display, Frame and Language can also be edited in online mode.
- As long as a visualization "reference" is not configured explicitely, the particular elements of the reference in online mode will react on inputs like those of the original visualization ("mother" of the references).
- When you switch the language ('Extras' 'Settings') this will only effect the display in online mode.
- A visualization can be printed in online mode.

Operation over the keyboard - in online mode

In order to get independent from the mouse or a touch screen, it is useful to configure a visualization in a way that allows pure keyboard operation:

Per default the following key (combinations) will work in online mode anyway (no special configuration necessary):

Pressing the <Tabulator> key selects the first element in the element list for which an input is configured. Each subsequent pressing of the key moves one to the next element in the list. Pressing the key while keeping the <Shift> key depressed selects the previous element.

The arrow keys can be used to change from a selected element to a neighbouring one in any direction.

The <**Space bar**> is used to execute an activity on the selected visualization element. If the element is one which has a text output variable, a text input field will be opened which displays the text contents of the variable. Pressing the <**Enter**> key writes in this value.

Additional key (combinations) for the online operation can be defined in the configuration dialog 'Keyboard usage'. There also the keys <Tab>, <Space> and <Enter> can get assigned another functions than the above described standards.

The individual elements of references behave in Online mode identically to the corresponding ones in the visualization that is referenced. They will therefore react the same way as individual elements to inputs and operation by mouse and keyboard; the display of tooltips in references is also element-dependent. When processing the element list, as for instance when jumping from one input element to the next using the tabulator, the processing of all individual elements of a reference proceeds from the location of the reference in the element list before jumping to the next element of the list.

Note: Operation over the keyboard in online mode is of greatest significance, if the visualization should be used with CoDeSys HMI resp. as Target-Visualization or Web-Visualization !

'File' 'Print' in online mode

'File' 'Print'is used to print out the contents of the visualization window in online mode. Visualizations which stretch over the border of the window can lead to inconsistencies particularly when there are moving elements in the visualization.

2.9 Visualizations in libraries...

Visualizations can also be stored in libraries and thus be made available to projects in the form of library POUs. They can be inserted as references or they can be called up via the command „Zoom to vis." in the input configuration of another visualization which is part of the project.

Note: Visualizations used in a project must have unique names. It can be problematic if for instance a visualization from a library is called or referenced which has the same name as one present in the project. Because, in processing references or visualization calls in the program, first the visualizations in the project, and only thereafter the ones in the loaded libraries will be implemented.

3 CoDeSys HMI

CoDeSys HMI is the run time system needed for the execution of the well known CoDeSys visualisation. This system enables the user to create an inexpensive software for operating machines and plants using the visualisations generated in CoDeSys.

CoDeSys HMI uses the already existent project file for the visualisation. Of course it is possible to protect the application program with a read-only protection. The masks then appear in the full-picture mode and merely the visualisation starting with the picture “PLC_VISU” is shown. All the other operational elements of CoDeSys like menus or the object organiser are not visible.

By seamlessly integrating the development platform for the visualisation masks into the PLC programming system CoDeSys HMI offers advantages other visualisations simply cannot come up to:

- A tag list of the variables to be used is not needed. It is possible to work directly with the CoDeSys variables.
- Variable values in input fields can be modified by expressions (e.g. “Variable1+ Variable2 * 12 + 5”)
- A mighty place-holder concept allows the user to create object-oriented masks. This means the user can build modules similar to the CoDeSys function blocks in the visualisation.

The CoDeSys functions “Sampling Trace“ and “read/write recipe“ are also available in CoDeSys HMI.

3.1 Installation, Start and Operating

A setup is available for the installation of the full version of CoDeSys HMI. A time limited demo version can be installed with the setup of the CoDeSys programming system.

CoDeSys HMI (**CoDeSysHMI.exe**) is started by a **command line**:

In each case at least the desired CoDeSys project has to be given in the command line. If no further parameters are set there, CoDeSys HMI automatically will start with a visualization POU named PLC_VISU (if existant in the project) and on that target or simulation mode, which was set when the project had been saved last.

Additionally as well the known command line and command file commands (see User Manual for CoDeSys) as the following special parameters can be used:

"/simulation" resp. "/target"	Per default it will be started in that mode which was set when the project was saved last. Using the parameter "/simulation" resp. "/target" in the command line it can be set explicitely whether the project should run in simulation mode or on the target.
/visu <visualization POU>	If the project contains a visualization POU named PLC_VISU, it will start automatically with this one. If another POU should be the entrance, it has to be set in the command line with "/visu <name of visualization POU>".
/visudownload	Overrides the download lock: If the user tries to log in with a project, which is different to that on the PLC, per default a download of the new project (dialog) can be initiated. If however there is an entry "visudownload=no" in the codesys.ini file, then no download is possible. This lock can be overridden by the command line parameter "/visudownload".

Example for a command line:

```
D:\PROGRAMME\CoDeSysHMI /simulation D:\PROJECTS\PROJECT.PRO /visu overview
```

The project project.pro will start in simulation mode and with the visualization POU 'overview'.

Please regard: Paths containing spaces must be bordered by quotation marks (").

The project will start in **full screen mode** with the entrance POU.

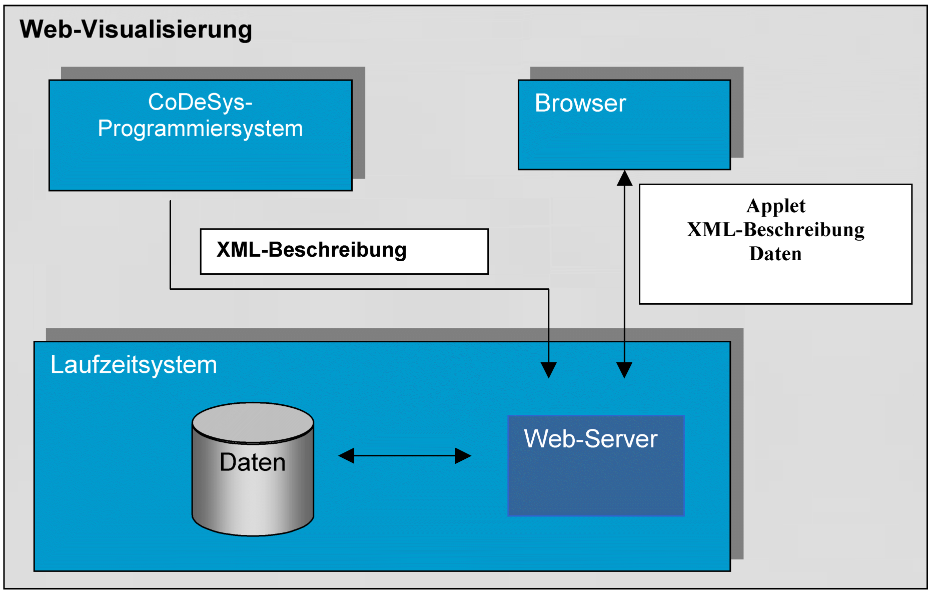
CoDeSys HMI can be operated corresponding to the functions of the visualization elements via keyboard and mouse.

If there is no visualization element configured with a corresponding function, CoDeSys HMI at any time can be terminated by pressing **<Alt><F4>**.

4 Web-Visualization

The Web-Visualization is a target specific application of a CoDeSys visualization.

CoDeSys can create XML descriptions of the visualization objects and download them to the PLC. There a Web-Server will provide the PLC data in XML format too and thus can create a continuosly updated visualization which can be opened in the Web Browser of any computer which is connected via Internet, independently from the target platform (e.g. useful for remote maintenance purposes).



Please regard: Due to the low tranfer rates of the Internet only about 100 variables per visualization window can be monitored.

4.1 Preconditions

- the target system must support the functionality; that means that in the target settings the option 'Web-Visualization' must be activated. If defined in the target file, this can be done by the user in the 'General' target settings.
- A **WebServer** must be started. (see below)
- In order to get the Web-Visualization displayed on a computer, a **Web-Browser** (IE-Explorer oder Netscape) is required.
- **Operating system**: Windows NT/2000, Windows CE, Linux, RTE

4.2 Editing the WebVisu.htm file

During CoDeSys installation the file WebVisu.htm (base-HTML-page for the Web-Visualization) gets copied to the subdirectory "visu" in the installation directory. The page can get edited there, before it gets downloaded to the target system together with the Web-Visualization project:

If webvisu.htm is opened in a text editor, it shows the following structure::

```
<HTML>
<HEAD>
   <TITLE>Applet HTML Page</TITLE>
</HEAD>
<BODY>
```

```
<APPLET   CODEBASE=.   CODE=webvisu/WebVisu.class   archive="webvisu.jar,minml.jar"
name="WebVisu" width="1600" height="1200">
<param name="STARTVISU" value="PLC_VISU">
<param name="UPDATETIME" value="100">
<param name="USECURRENTVISU" value="FALSE">
</APPLET>
</BODY>
</HTML>
```

The following parameters can be set to control the behaviour of the Web-Visualization:

- STARTVISU - Definition of the start POU (Default: PLC_VISU)
- UPDATETIME - Definition of the monitoring interval (msec)
- USECURRENTVISU - Definition whether an automatic change to another visualization will be done, as soon as the system variable 'CurrentVisu' is changed by the PLC program.
- The parameters „width“ and „heigth“ define the size of the screen (Regard the possibility to make visible this size already during creating a visualization in CoDeSys).

4.3 Preparing a Web-Visualization in CoDeSys

- Create the visualization(s) for your PLC program as usual in the CoDeSys programming system. If you want a certain visualization to be called as starting object, then name it '**PLC_VISU**'. It will be loaded automatically as soon as the visualization is called via the Internet.
- For visualization objects, which should not be part of the web version of the visualization, deactivate option Web-Visualization' in the dialog **'Object' 'Properties'** in category 'Visualization'.
- Perform command 'Project' 'Clean all', then 'Project' 'Build'.
- Log in to the target system ('Project' **'Online' 'Login'**) and start the project on the PLC.

4.4 Configuration and Start of the Web Server

- The WebServer must be available as an appropriate executable file (*.exe) for the used target system.
- The configuration of the server can be done by a configuration file or by parameters added in the command line when calling the server-exe. The definitions given by a configuration file will overwrite those of the command line.

  The possible parameters:

webserver-port-nr	Port, on which the WebServer expects requests of the client (Web-Browser)	Default: 80
target-port-nr	Port of the runtime system	Default: 1200
target-ip-address	IP-Address of the runtime system	Default: localhost
use-file-upload-dir	If this flag is set to TRUE, additionally the file-upload-directory (see below, file-upload-dir) must be defined, where the *.xml, *.bmp, *.jpg, etc. files of the Web-Visualzation are stored on the target system and from where they are uploaded to the browser.	Default:false
file-upload-dir	Directory for the Web-Visualization files	Default:“”

Please regard: If you define a "file-upload" directory, where the controller can store the visualization download files, then the visualization files will be automatically updated at each download. The advantage of this upload-directory is that the controller is not involved and thus not strained. The Web Server gets the files directly from the directory and by this the data transfer is much quicker. This is especially of impact in case of a big amount of data.

A configuration file for the Web Server must be available in XML-format and it must be named "**webserver_conf.xml**". It must be found in the directory where the webserver-exe is. If no configuration file is available, the above mentioned default settings will be used (if not changed by parameters added to the call of the server in a command line.)

An example configuration in the webserver_conf.xml file could look as follows:

```
<webserver-configuration>
<webserver-port-nr> 8080 </webserver-port-nr>
<target-port-nr> 1200 </target-port-nr>
<target-ip-address> localhost </target-ip-address>
<use-file-upload-dir> true </use-file-upload-dir>
<file-upload-dir> C:\Programme\CoDeSysV23 </file-upload-dir>
</webserver-configuration>
```

A call in a command line must use the following syntax:

```
WebServer [webserver-Port-nr] [target-port-nr] [target-IP-address] | [file-upload-dir]
```

Thus a call corresponding to the above shown configuration example would look like follows:

```
> webserver 8080 1200 localhost c:\Programme\CoDeSysV23
```

Please regard: The parameter settings in the command line are without any effect, if there is a valid configuration file available.

4.5 Calling a Web-Visualization via Internet

- Insert the following address in the Browser:
  http://<IP-Adresse of the WebServer>:<Port of the WebServer>/webvisu.htm
- **WebVisu** is the default HTML-file. It contains a <applet> tag which will start the WebVisu-Applet.

5 Target-Visualization

The Target-Visualization is one of the possible ways to use a CoDeSys Visualization. CoDeSys can create ST code (Structured Text) for the visualization objects of a project. This code can be downloaded to the PLC together with the normal project code.

Thus, if it is supported by the target system and if an appropriate monitor is available, the visualization can be started directly on the PLC computer. It is not necessary any longer that the programming system is running in order to run the visualization. This means a considerable reduction of storage use.

5.1.1 Preconditions

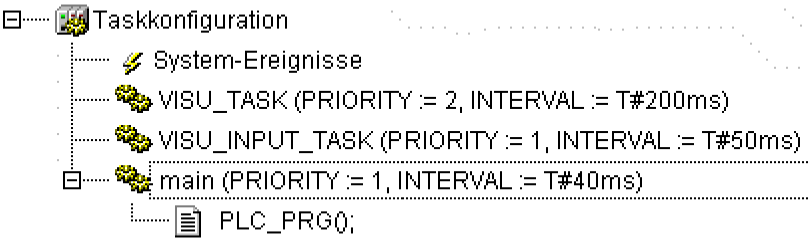
- The target system must support the functionality; that means that in the target settings the option 'Target-Visualization' must be activated. If defined in the target file, this can be done by the user in the 'General' target settings.
- The library **SysLibTargetVisu.lib** (external) is required for implementing the visualization functions in the runtime system. It will be linked to the project as soon as the option 'Target-Visualization' is activated in the target settings. In the runtime system the SysLibTargetVisu.lib also must be implemented.
- Operating system of the PLC computer: Windows NT/2000, Windows CE
- The PLC computer needs devices for displaying and operating the visualization (monitor, keyboard etc.)

5.1.2 Creating a Target-visualization

1. Create a visualization in CoDeSys. In order to optimize the performance of the visualization, put as many elements, which are static (no movement, no dynamic texts, no dynamic color changes), to the back. Hint: Using 'Extras' 'Elementlist' several or all elements can be "Sent to back" or "Sent to front" in a single blow. (Explanation: All static elements only once will be drawn to a background bitmap. Thus the cyclic repaint time will be reduced. This will be profitable especially for complex polygons or bitmaps.)
   If there is a visualization object PLC_VISU, the Target-Visualization later will be started with this object. Otherwise it will start with that object which is the first one in the list of visualization objects in the Object Organizer.

Please regard: For target computers running with Windows NT/2000, Windows CE the bitmaps of the visualization will be transferred as files.

2. For each visualization object, which should not be part of the Target-Visualization, deactivate the option 'Target-Visualization' in the Object Properties dialog ('Project' 'Object' 'Properties') in category 'Visualization'.
3. Make sure that the option 'Optimized jumps' is deactivated in the target settings (Target Platform).
4. A project which is intended to be used with the Target-Visualization, must be controlled by a task configuration. If the option 'Target-Visualization' is activated in the target settings, then you there also can define whether two or only one task should be created automatically in the task configuration. It is recommended to use two tasks (see below), but this is not possible with all types of target systems. The tasks: VISU_INPUT_TASK will do the processing of the user inputs, VISU_TASK will do the redrawing (update) of the visualization elements. See the default settings of the tasks in the example which is shown at the end of the paragraph. Of course the parameters can be modified. But: VISU_INPUT_TASK always should be processed before VISU_TASK in order to guarantee a useful interaction of user inputs and update of the visualization. That task, which calls the main program (e.g. PLC_PRG), should at least be processed as often as VISU_INPUT_TASK, ideally even with a higher priority, but it also could be added directly to VISU_INPUT_TASK.



```
Taskkonfiguration
    System-Ereignisse
    VISU_TASK (PRIORITY := 2, INTERVAL := T#200ms)
    VISU_INPUT_TASK (PRIORITY := 1, INTERVAL := T#50ms)
    main (PRIORITY := 1, INTERVAL := T#40ms)
        PLC_PRG();
```

5. Load the project into the controller ('Online' 'Login')

5.1.3 Starting the Target-Visualization

Start the loaded project on the PLC. Hereupon the visualization will start with the PLC_VISU object resp. – if no PLC_VISU is available – with that visualization object which is the last one in the list of visualization objects in the Object Organizer.

6 Index

A

Access rights in Visualization 2-22
Align 2-9
Angle 2-12

B

Bar Display 2-7
Bar Display in Visualization 2-32
Bitmap 2-6
Bitmap in Visualization 2-34
Button 2-6

C

Clear Background Bitmap 2-41
CoDeSys HMI
    Special configuration possibilities in Visualisation 2-37
CoDeSys HMI 2-37
Color Variables in Visualization 2-17
Colors in Visualization 2-16
Configuration of a Web-Server 4-2
Create Visualization 2-4
Curve 2-5

D

DEFINERECEIPT 2-37
Display in visualization 2-39
Dynamic texts 2-39

E

Element list 2-9
Ellipse 2-5
EXITPROGRAM 2-37
Extras
    Select 2-8
Extras Align 2-9
Extras Clear Background Bitmap 2-41
Extras Configure
    Extras Menu\Keyboard Usage 2-41
Extras Configure 2-11
Extras Element list 2-9
Extras Placeholder list 2-47
Extras Select All 2-8
Extras Select Background Bitmap 2-41
Extras Send to Back 2-9
Extras Send to Front 2-9
Extras Settings 2-39

F

File Print in online mode 2-49
Frame 2-39

G

Grid 2-39
Grouping elements in Visualization 2-8

H

HELP 2-37
Histogram 2-7
Histogram in Visualization 2-33

I

Input 2-21
Insert Bar Display 2-7
Insert 'Bitmap' 2-6
Insert Button 2-6
Insert Curve 2-5
Insert Ellipse 2-5
Insert Histogram 2-7
Insert Menu
    Rounded Rectangle 2-4
    Table 2-6
Insert Meter 2-7
Insert Pie 2-5
Insert Polygon 2-5
Insert Polyline 2-5
Insert Rectangle 2-4
Insert Rounded Rectangle 2-5
Insert Table 2-6
Insert Visualization 2-6
Insert visualization elements
    Rectangle 2-4
Insert WMF file 2-6

K

Keyboard usage in Visualization 2-41

L

LANGUAGE 2-37
Language in visualization 2-39
Line width in Visualization 2-16
LOADWATCH 2-37

M

Menu Extras
    Align 2-9
    Clear Background Bitmap 2-41
    Configure 2-11
    Element list 2-9
    Placeholder list 2-47
    Select All 2-8
    Select Background Bitmap 2-41
    Send to Back 2-9
    Send to Front 2-9
    Settings 2-39
Menu Insert
    Bar Display 2-7
    Bitmap 2-4, 2-6
    Button 2-4, 2-6
    Curve 2-5
    Ellipse 2-4, 2-5
    Histogram 2-7
    Meta file 2-6
    Meter 2-7
    Pie 2-5

  - Polygon 2-4, 2-5
  - Polyline 2-5
  - Rectangle 2-4
  - Rounded Rectangle 2-5
  - Visualization 2-4, 2-6
  - WMF file 2-6
- Meter 2-7
- Meter element in Visualization 2-30
- Motion absolute in Visualization 2-18
- Motion relative in Visualization 2-19

O

- Online Mode
  - Visualization 2-48
- Online Mode 2-48
- Operating version 2-37

P

- Pie 2-5
- Placeholder 2-11
- Placeholder List 2-47
- PLC_VISU 4-2
- Polygon 2-5
- Polyline 2-5
- PRINT 2-37
- Programmability 2-23

R

- Rectangle 2-4
- Reference 2-11
- Rounded Rectangle 2-5

S

- SAVEWATCH 2-37
- Security in Visualiszation 2-22
- Select Background Bitmap 2-41
- Selecting elements in Visualiziation 2-8
- Send to Back 2-9
- Send to Front 2-9
- Settings in visualization 2-39
- Shape 2-12
- Status bar 2-10
- SysLibTargetVisu.lib 5-1

T

- Table 2-6
- Table element configuration 2-26
- Target-Visualization 5-1
- Text 2-13
- Text in Visualization 2-13
- Textvariables in Visualization 2-15
- ToolTip in Visualization 2-22
- TRACE 2-37
- Translation file
  - Visualization texts 2-13

V

- Variables in Visualization 2-20
- Visualisation
  - Input possibilities for CoDeSys HMI 2-37
- Visualisierung
  - Access rights 2-22
  - Angle 2-12
  - Security 2-22
- Visualization
  - Align 2-9
  - Bar Display element 2-32
  - Bitmap 2-34
  - Changing the Selection and Insert Mode 2-8
  - Clear Background Bitmap 2-41
  - Color Variables 2-17
  - Configure Colors 2-16
  - Configure Input 2-21
  - Configure Line width 2-16
  - Configure Meter 2-30
  - Configure Motion absolute 2-18
  - Configure Motion relative 2-19
  - Configure Shape 2-12
  - Configure Text 2-13
  - Configuring a Visualization - Overview 2-10
  - Copying Visualization Elements 2-8
  - Create 2-4
  - Display 2-39
  - Dragging Visualization Elements 2-8
  - Editor 2-3
  - Element list 2-9
  - Execute program 2-21
  - File – Print in online mode 2-49
  - Frame 2-39
  - Grid 2-39
  - Grouping elements 2-8
  - Histogram 2-33
  - in libraries 2-49
  - Insert Bar Display 2-7
  - Insert Histogram 2-7
  - Insert Meter 2-7
  - Insert Pie 2-5
  - Insert Table 2-6
  - Insert WMF file 2-6
  - Keyboard usage 2-41
  - Language 2-39
  - Modifying Visualization Elements 2-8
  - Online Mode 2-48
  - Operation over the keyboard in online mode 2-48
  - Placeholder 2-11
  - Programmability 2-23
  - Reference 2-11
  - Select 2-8
  - Select All 2-8
  - Select Background Bitmap 2-41
  - Selecting Visualization Elements 2-7
  - Send to Back 2-9
  - Send to Front 2-9
  - Status Bar 2-10
  - Table 2-26
  - Text input of variable 'Textdisplay' 2-21
  - Textvariables 2-15
  - Tip FALSE 2-21
  - Tip variable 2-21
  - Toggle variable 2-21
  - Tooltip 2-22
  - Variables 2-20
  - Visualization Object
    - Configure... 2-39
  - Visualization\Settings 2-39
  - Zoom to Vis. 2-21
- Visualization 2-10
- Visualization 2-21
- Visualization 2-41

Visualization as Visualization element 2-35
Visualization Elements
    Configure 2-11
Visualization Elements 2-4

W

Web Server
    Configuration and Start 4-2
    configuration file 4-2
    File upload directory 4-2
WebVisu.htm 4-1
Web-Visualization
    PLC_VISU 4-2
    WebVisu.htm 4-1
Web-Visualization 4-1
WMF file 2-6
WRITERECEIPT 2-37