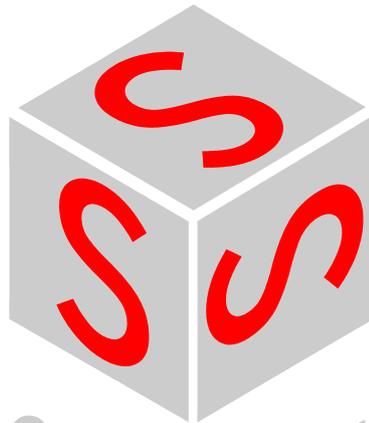


# CoDeSys Visualization

Supplement to the User Manual for  
PLC Programming  
with CoDeSys 2.3



S m a r t  
S o f t w a r e  
S o l u t i o n s

Copyright © 2003, 2004, 2005, 2006, 2007 by 3S - Smart Software Solutions GmbH  
All rights reserved.

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, MS-DOS and Intellisense are registered trademarks, Windows is a  
trademark of Microsoft Corporation.

Publisher

3S - Smart Software Solutions GmbH  
Memminger Straße 151  
DE-87439 Kempten  
Tel. +49 831 5 40 31 - 0  
Fax +49 831 5 40 31 – 50

**Last update 27.04.2007**

**Document Version 2.0, CoDeSys V2.3.8.0**

## Table of Content

<b>1</b>	<b><u>Advantages of an integrated visualization</u></b>	<b>1-1</b>
<b>2</b>	<b><u>The Visualization Editor in CoDeSys</u></b>	<b>2-1</b>
2.1	Create a new Visualization .....	2-2
2.2	Visualization Elements, Insert.....	2-2
2.3	Positioning of Visualization Elements.....	2-6
2.4	Visualization, Configure - Overview .....	2-9
2.4.1	Visualization Elements, Configure... ..	2-10
2.4.2	Visualization Object, Configure.....	2-53
2.4.3	Image files in the visualization .....	2-57
<b>3</b>	<b><u>Language switching in the Visualization</u></b>	<b>3-1</b>
3.1	Static language switching .....	3-1
3.2	Dynamic Language switching .....	3-3
3.2.1	Configuration.....	3-3
3.2.2	XML-File for Dynamic Texts.....	3-4
3.3	Calling up language-dependent Online Help via a visualization element .....	3-8
<b>4</b>	<b><u>Placeholder Concept</u></b>	<b>4-1</b>
<b>5</b>	<b><u>Visualization in Online Mode...</u></b>	<b>5-1</b>
5.1	Overview .....	5-1
5.2	Operation over the keyboard - in online mode.....	5-1
5.3	'File' 'Print' in online mode.....	5-2
5.4	Access Protection in multi-client operation .....	5-2
<b>6</b>	<b><u>Visualizations in libraries...</u></b>	<b>6-1</b>
<b>7</b>	<b><u>CoDeSys HMI</u></b>	<b>7-3</b>
7.1	Installation, Start and Operating .....	7-3
<b>8</b>	<b><u>Web-Visualization</u></b>	<b>8-1</b>
8.1	Preconditions .....	8-1
8.2	Status check, Auto-Reload, File error_ini.xml.....	8-1
8.3	Editing the WebVisu.htm file .....	8-2
8.4	Preparing a Web-Visualization in CoDeSys.....	8-4
8.5	Configuration and Start of the Web Server .....	8-4
8.6	Calling a Web-Visualization via Internet .....	8-6
8.7	Restrictions and Special Features .....	8-7
<b>9</b>	<b><u>Target-Visualization</u></b>	<b>9-1</b>
9.1	Preconditions .....	9-1
9.2	Creating a Target-Visualization.....	9-1
9.3	Starting the Target-Visualization.....	9-4
9.4	Scan of mouse-clicks and dynamic texts .....	9-4
9.5	Restrictions .....	9-5
<b>10</b>	<b><u>APPENDIX</u></b>	<b>10-1</b>

<b><u>Appendix A</u></b>	<b><u>Implicit Variables in the Visualization</u></b>	<b><u>10-1</u></b>
<b><u>Appendix B</u></b>	<b><u>Example Dll for ActiveX Control</u></b>	<b><u>10-3</u></b>
<b><u>Appendix C</u></b>	<b><u>Possible Key Combinations</u></b>	<b><u>10-5</u></b>
<b><u>Index</u></b>		<b><u>9</u></b>

# 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 CoDeSys User Manual), allowing the user to create visualization objects to the parallel to the development of the application in the same user interface.

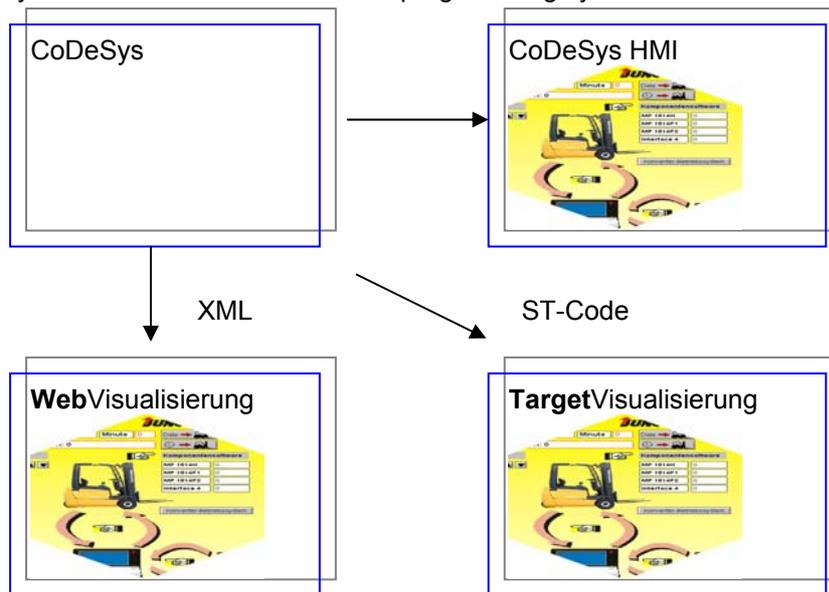
## The integration offers a lot of benefits:

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 Exchange) interface is not necessary. See chapter 7.

### 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 developed in the programming language JAVA, which can be processed by browsers) and which can be displayed via TCP/IP on a browser. Thus the visualization data are available online on various platforms. See chapter 8.

#### 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 ported with low effort to any devices which are programmable with CoDeSys. See chapter 9.

##### Overview on the functions:

- Elements
  - Rectangle, Ellipse, Rounded rectangle
  - Line, Polygon, Polyline, Curve
  - Bitmap, WMF-file
  - ActiveX Control
  - Button, Table, Histogram, Bar Display, Meter
  - Reference to another visualization (instance)
- 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
  - Master layout
  - 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

The CoDeSys 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 and in the object **properties** dialog. 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 programmability of element properties via structure variables.

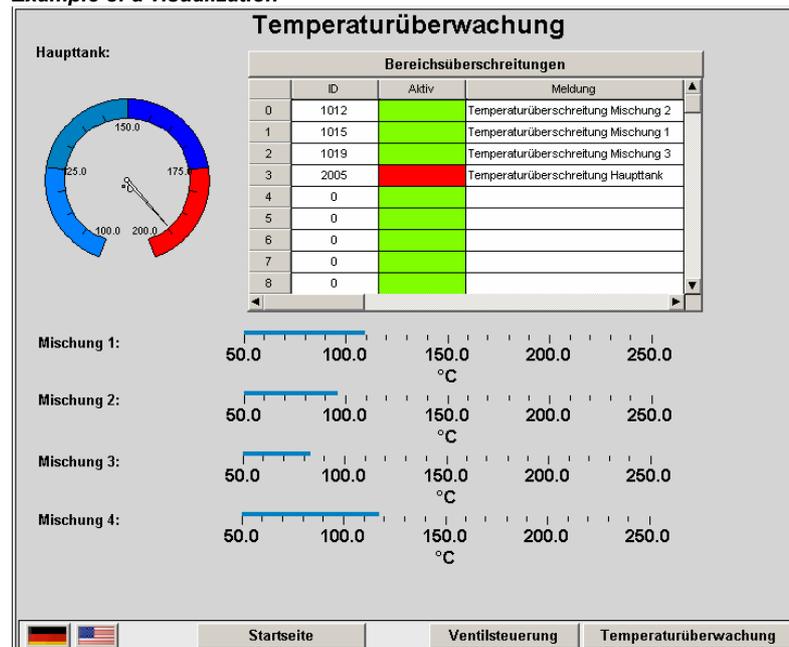
Using placeholders in the configuration dialogs may save a lot of effort in case you want to use the same visualization object several times with different configurations.

The visualization which is created in the programming system will in many cases be used as the only user interface available for controlling and watching the associated PLC program in online mode. For this purpose it must be possible to give inputs to the program solely by activating visualization elements. To reach this you can use special input possibilities during the configuration of the visualization and you have the option to define special hotkeys for each particular visualization.

A visualization created in CoDeSys can later be used in different ways:

- It can be made available on **CoDeSys HMI**, a special runtime system for operating the visualization in full screen mode on a PLC computer.
- It can be made available as a **Web-Visualization**, which allows to call and operate it via the Internet (useful for remote maintenance purposes)
- It can be made available as a **Target-Visualization**, which can be started directly on the PLC.

### Example of a visualization



## 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 Organizer.

Using 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.

When the visualization object is marked in the Object Organizer, via command 'Project' 'Object' 'Properties' (see CoDeSys User Manual) the Properties dialog can be opened, where you can make settings concerning the usage of the object in a Web- or Target-Visualization as well as concerning a Master layout.

When defining the name of the visualization object, please regard the following:

1. A visualization named "**PLC\_VISU**" per default automatically will be used as start visualization in a Target- or Web-Visualization resp. in CoDeSys HMI, if there not explicitly another visualization is configured for this purpose.
2. A visualization may **not get the same name as another object** within the project because this would result in problems when changing between visualizations.

**Please regard:** If you want to use the implicit variable **CurrentVisu** (type STRING) for addressing the currently opened visualization object, for compiler versions < V2.3.0.7, and if the library SysLibStr.lib is not included in the project, you **must** use capital letters for the names of the visualization objects (e.g. "PLC\_VISU"). (For information on implicit variables see 10)

## 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'**,  **'ActiveX-Element'**,  **'Scrollbar'**,  **'Meter'**,  **'Bar Display'**,  **'Histogram'**,  **'Alarm table'**,  **'Trend'**,  **'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.

Regard also the possibility of using placeholders and Master layouts.

### 'Insert' 'Rectangle'

Symbol: 

With the command you can insert a rectangle as an element into your present visualization (see chapter 2.2).

### 'Insert' 'Rounded Rectangle'

Symbol: 

With the command you can insert a rectangle with rounded corners as an element in your present visualization (see chapter 2.2).

### 'Insert' 'Ellipse'

Symbol: 

With the command you can insert a circle or an ellipse as an element in your present visualization (see chapter 2.2).

### 'Insert' 'Polygon'

Symbol: 

With the command you can insert a polygon as an element in your present visualization (see chapter 2.2).

### 'Insert' 'Polyline'

Symbol: 

With the command you can insert a line as an element into your current visualization (see chapter 2.2).

### 'Insert' 'Curve'

Symbol: 

With the command you can insert a Bezier curve as an element into your current visualization (see chapter 2.2).

### '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 attached to the element, indicating the corner of a virtual rectangle surrounding 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 separately. 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 being 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 standard dialog box for browsing for a file will be opened. You can use this dialog to select and enter a bitmap file from the local file system. In the configuration dialog of the inserted bitmap you then can define, whether a link to the bitmap file should be stored or the bitmap should be inserted as an element.

Alternatively you can specify a project variable to define which bitmap should be used. This allows a dynamic change of bitmaps in online mode. The variable must be entered in the configuration dialog of an already inserted bitmap element and it must contain the name of a bitmap file managed in the project-global bitmap list.

For a description on the configuration of a bitmap element please see 2.4.1, category "Bitmap".

### '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 an instance.

### '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.

Like for a "Bitmap" element also for the filling of a button element an image file can be specified (static or dynamic use). See the description of the configuration dialog for category „Bitmap"(Kap.2.4.1).

**'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' 'ActiveX-Element'**

**Symbol:** 

Use this command to insert an ActiveX Control into your current visualization. It can be used later on Windows32 systems in CoDeSys HMI and the CoDeSys Target-Visualization. While pressing the left mouse button, bring up an area in the desired size. It will be inserted as a rectangle with the inscription "Control:".

To select a certain ActiveX-Control and to configure the method calls and the display open the dialog 'Configure ActiveX-Control' by a double-click on the element resp. via command 'Extras' 'Configure' (see Chapter 2.4.1, ActiveX-Element).

**'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 length 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.

**'Insert' 'Alarm table'**

Symbol: 

Use this command to insert an alarm table into your current visualization object.

While pressing the left mouse button, bring up an area in the desired size. Before the element gets displayed the configuration dialog 'Configure Alarm table' will open. Here you will find additionally to the standard categories Tooltip and Security the categories 'Alarmtable', 'Settings for sorting', 'Columns' and Settings for alarmtable' where you can define contents and appearance of the table.

An alarm table can be used to visualize the alarms, which have been defined in the Alarm configuration of the project.

**'Insert' 'Trend'**

Symbol: 

Use this command to insert a trend element into your current visualization object.. While pressing the left mouse button, bring up an area in the desired size. The configuration (axes, variables, history) is done in the configuration dialog 'Trend'.

The trend element, also named oscilloscope element, is used to display variable values within a certain time period. It stores the data in a file on the client and displays them as a graph. As soon as a value changes, a new entry will be made in the file, showing date/time and the new value.

The trend element is drawn transparently. So you can assign any desired background (bitmap, color).

## 2.3 Positioning of Visualization Elements...

---

### Selecting Visualization Elements

The selection mode is activated by default. 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 mouse-click. Then do further mouse-clicks while the <Ctrl>+<Shift>-buttons are 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.

In the element list (called by '**Extras' 'Element list'**) you can select an element in the visualization by selecting the respective 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 mouse-key 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.

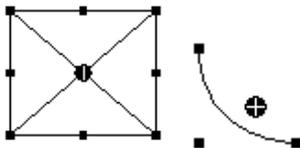
### Copying Visual Elements

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, depending on the current configuration all elements will be stretched, compressed or keep their original size; 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' 'Ungroup'. The configuration of the group (see Chapter 2.4.1, Group) will be lost in this case.

<b>Note:</b>	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.

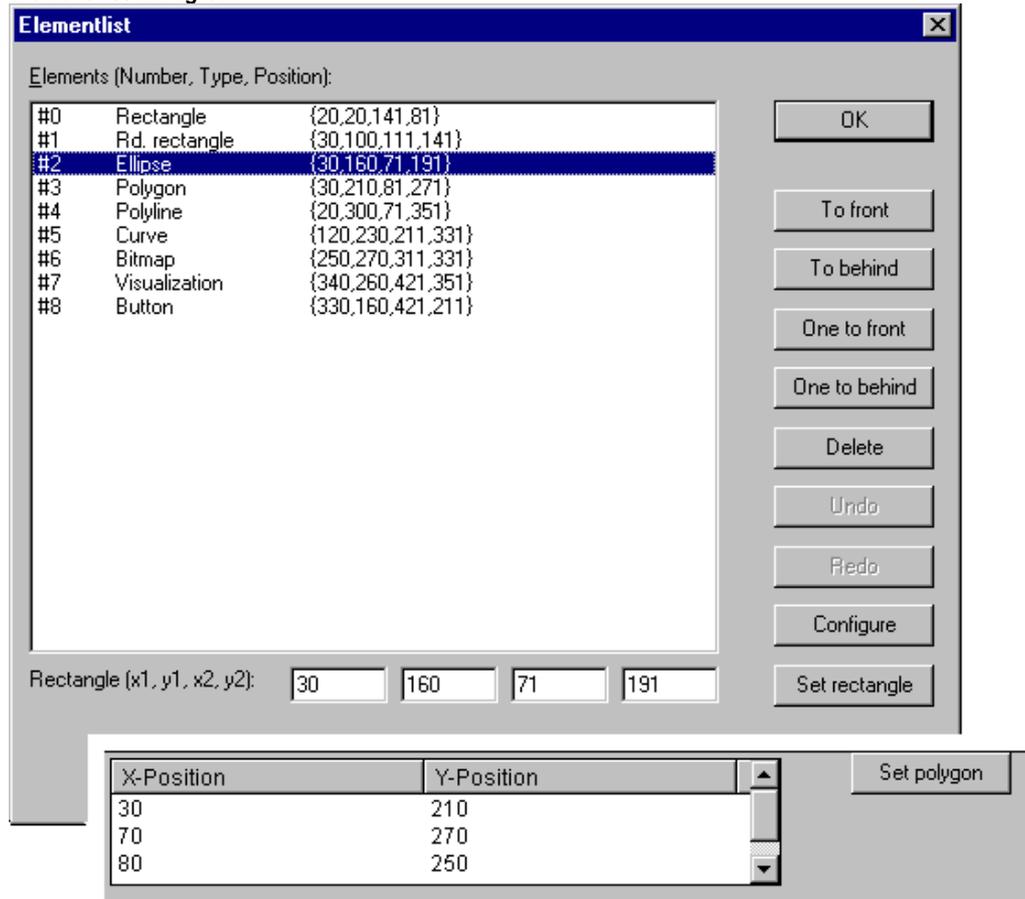
When one or more items have been selected, the corresponding elements in the visualization are marked for visual control and if necessary the display will scroll to that section of the visualization that contains the elements.

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.

Element list dialog box



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.

### 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, Configure - Overview

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.

## Placeholders

At each location in the configuration dialog where 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 1 for the placeholder concept.

### 2.4.1 Visualization Elements, Configure...

In the configuration dialogs opened via the 'Extras' 'Configure' command 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 interface for a PLC program (see also: INTERN in category 'Input' of the configuration dialogs, Keyboard usage)

**Attention:** Dialogs which include the configuration of **Colors and Fonts** for a visualization element, can look differently depending on the currently selected target system. Possibly instead of the standard dialog there is a dialog with restricted options. This might be reasonable for projects, which are designated for a use in Target-Visualizations. See also below: 'Colors' resp. 'Text'.

### '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 double-click 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.

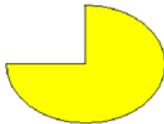
Regard also the possibility of using Placeholders (Chapter 2.4.1).

## Angle

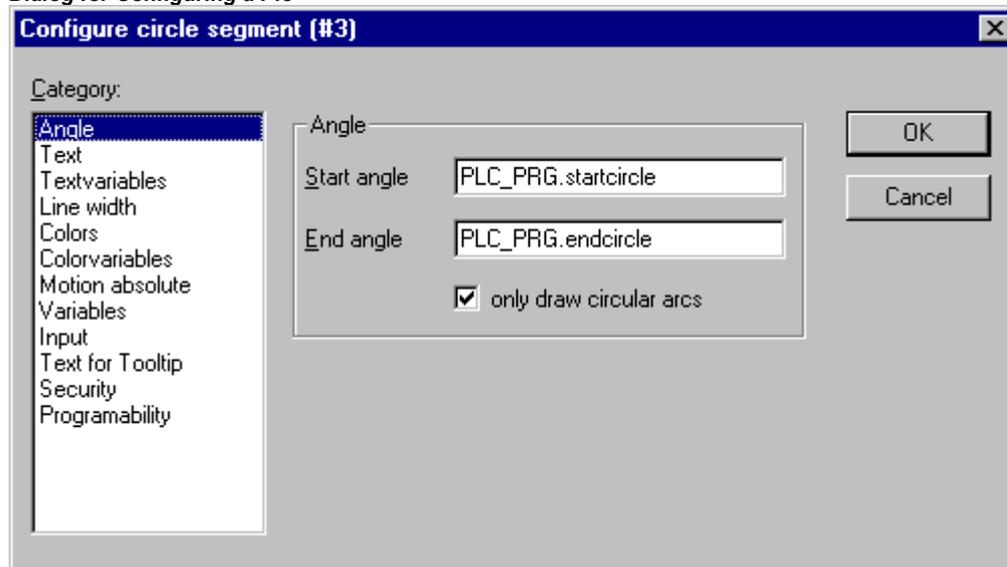
In the configuration dialog 'Configure Pie' in the **Angle** category you can each enter a value or a variable defining the **start angle** and the **end angle** of the sector element in degrees. The sector will be drawn clockwise from the start angle position to the end angle position.

**Example:**

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



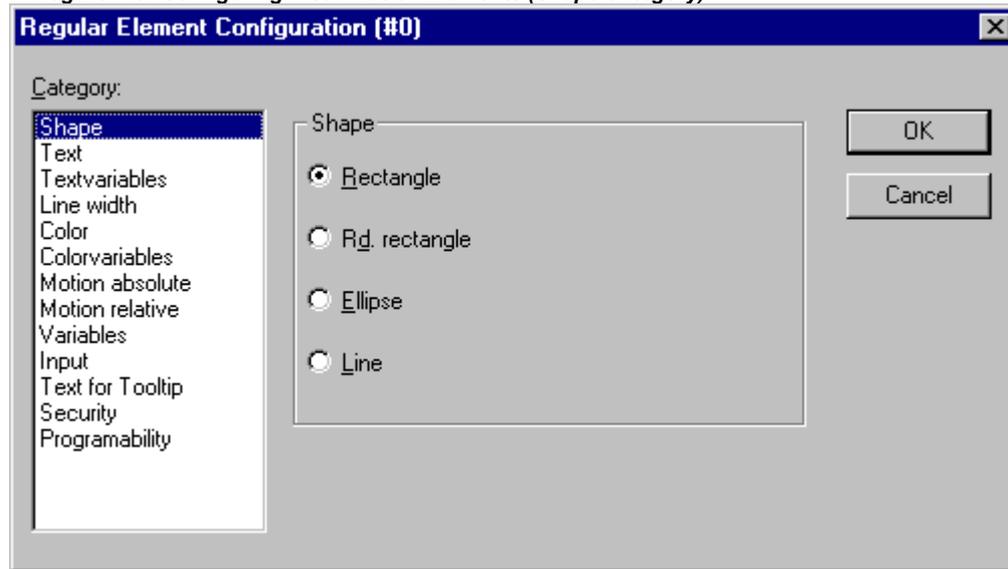
*Dialog for Configuring a Pie*



## 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)



## Text

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, category '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 (see Chapter 2.4.1).

Enter the text in the **Content** field. With the key combination <Ctrl>+<Enter> you can insert line breaks, with <Ctrl>+<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':

Character	Argument / Output as
d,i	Decimal number
o	Unsigned octal number (without leading zero)
x	Unsigned hexadecimal number (without leading 0x)
u	Unsigned decimal number
c	Single character
s	String
f	REAL-values [-]m.<accuracy>; plus- resp. minus-sign defines right (default) resp. left alignment; m defines minimal number of places to be displayed; accuracy defines the number of places behind the comma (default: 6).

**Attention:** (1) If you want to get displayed a percent sign % combined with one of the formatting strings mentioned above, you must enter "%%". For example: Enter "Rate in %: %s" to get displayed in online mode "Rate in %: 12" (if the text display variable currently is "12").  
(2) Be careful with letter case: capital letters, i.e. %S instead of %s not acceptable."

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 of a REAL variable in the 'Text Output' field, e.g.: `fvar1`

➔ 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

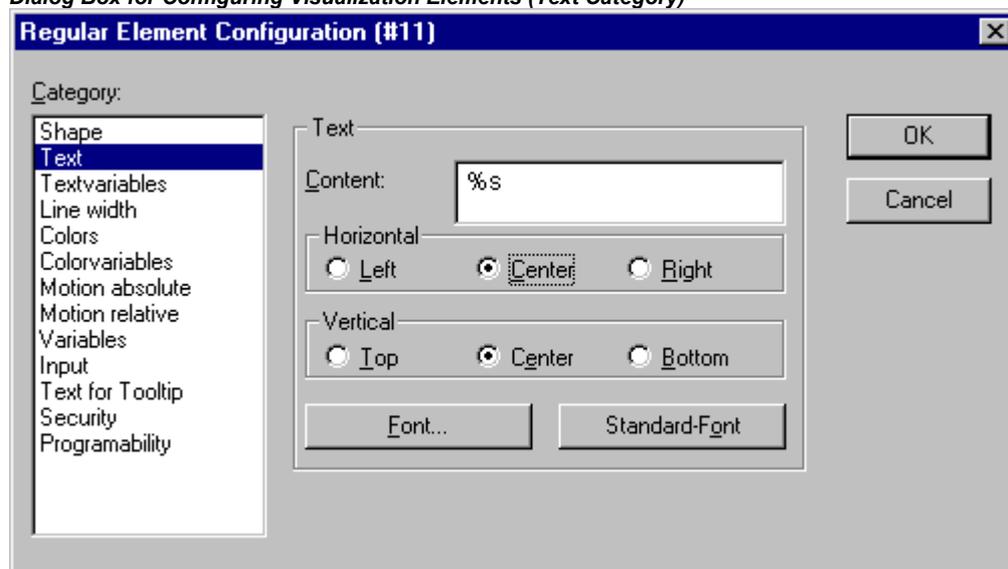
**Note:** If a text string is to be transferred into a **translation file**, which will then be used in Online mode to enable switching into another national language, it must be delimited at the beginning and end by #.  
Examples: "#Pump 1#" or else even "#Pump# 1"  
The second case might for example, 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 defined in a xml-file available for this purpose and listing all possible dynamic texts (see chapter 3.2.2). 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 (chapter 2.4.2) resp. of the Language switching in a visualization (chapter 3). See there also information on the usage of Unicode-Format.

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. The selection list might be restricted depending on the current target (see note in chapter 2.4.1). 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.

*Dialog Box for Configuring Visualization Elements (Text Category)*



## 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.

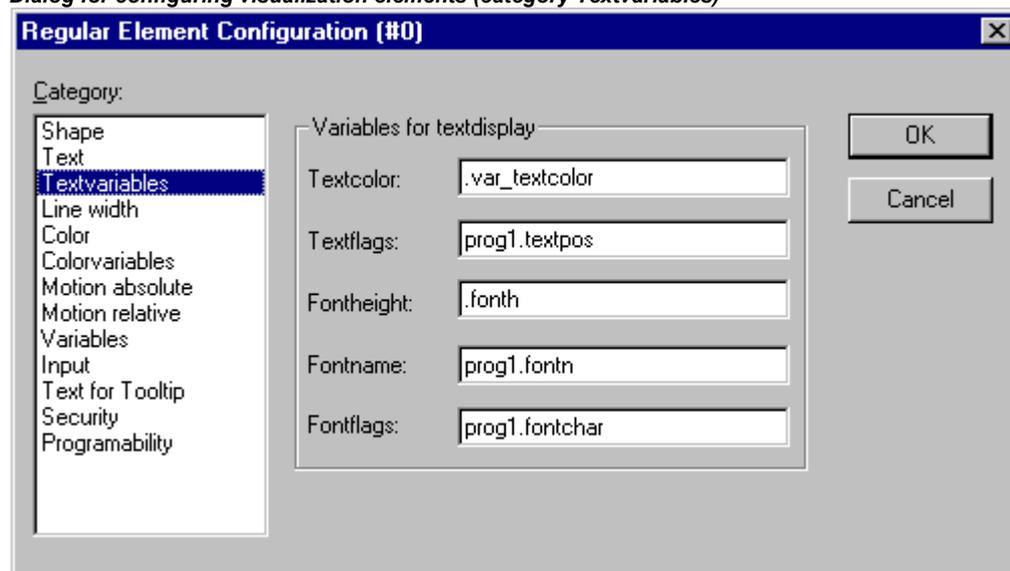
**Note:** If there are corresponding static definitions in category 'Text', these will be overwritten by the dynamic parameter values.

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)*

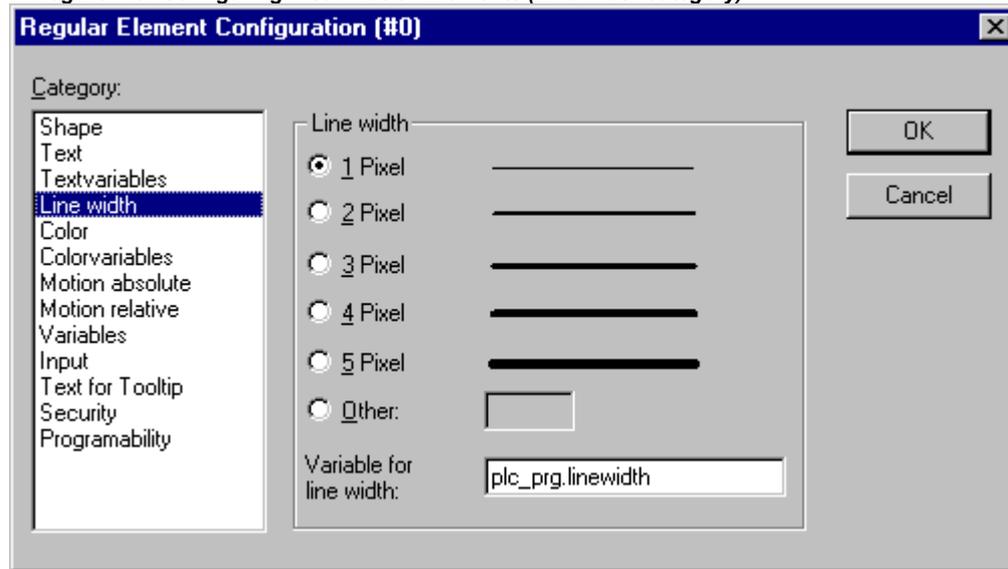


### Line width

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 'Programability'), the static setting will be overwritten in online mode.

Dialog Box for Configuring Visualization Elements (Line width category)



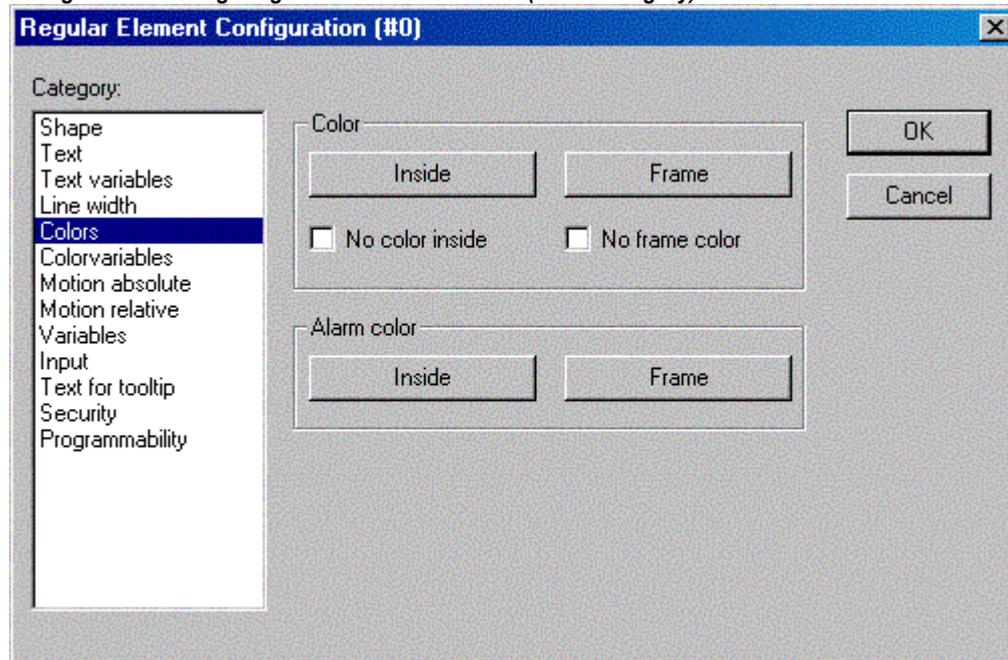
## Colors

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. Choosing 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)



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**.

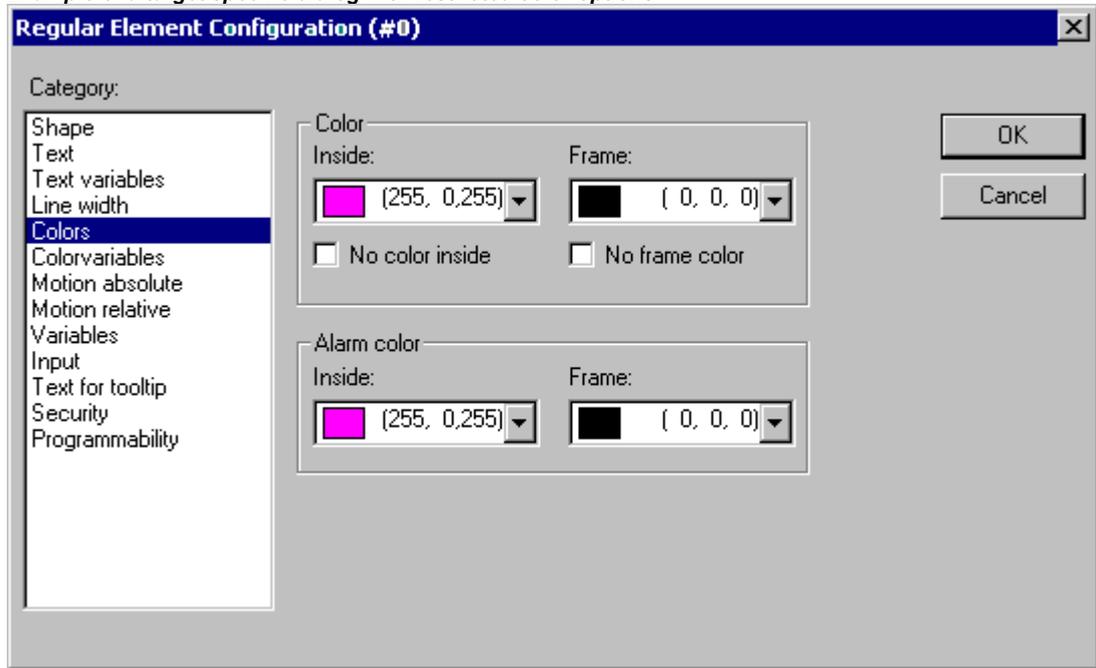
**Note:** The change color function only becomes active, if the PLC is in Online Mode!

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.

Depending on the target the dialog might offer only a **restricted selection of colors**. This can be reasonable for creating projects which are intended for Target-Visualizations. In this case the colors are defined via selection lists.

*Example of a target specific dialog with restricted color options*



**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 "Programmability" 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.

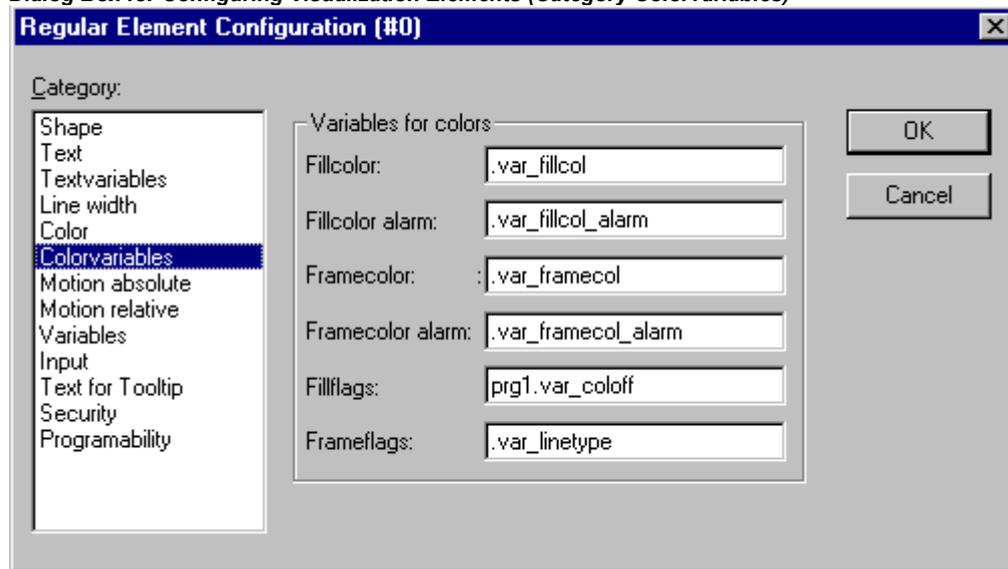
In case of multiple definition of an element property consider the specific order of precedence concerning according to which a value might be overwritten in online mode by another.

The parameters of the dialog:

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

<b>Framecolor alarm:</b>	frame color if the 'Change color' variable is TRUE	"plc_prg.var_framecol"	var_framecol:=16#FF00FF → alarm frame color farbe pink	dwFrameColorAlarm
<b>Fillflags:</b>	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
<b>Frameflags:</b>	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)



### Motion absolute

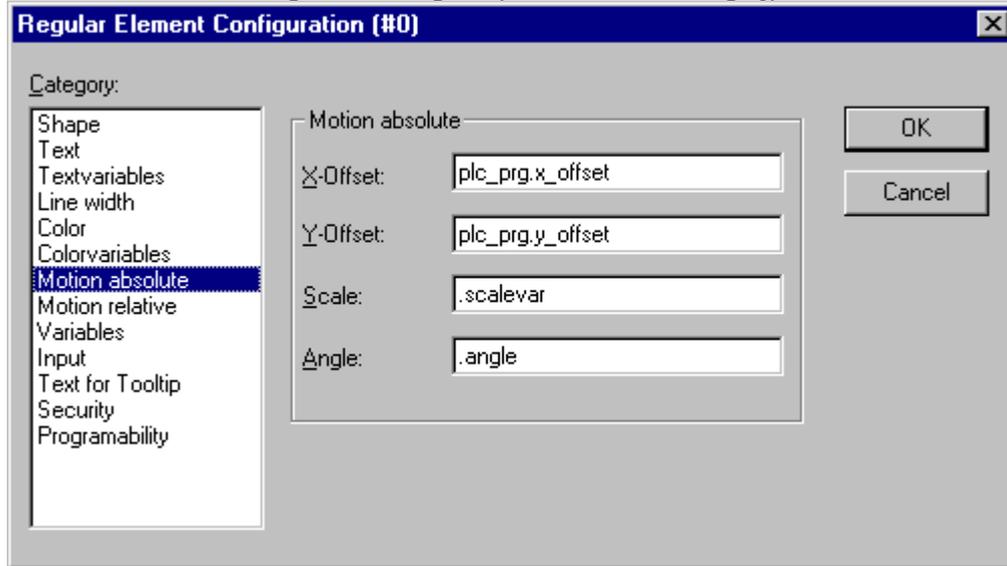
In the visualization element configuration dialog box, in the **Motion absolute** category, **X-** or **Y-Offset** fields variables can be entered. These variables can shift the element in the X or the Y direction, depending on the respective variable value. A variable in the **Scale** field will change the size of the element linear to its current value. This value, which is used as scaling factor, will be divided by 1000 implicitly, 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)*

### Motion relative

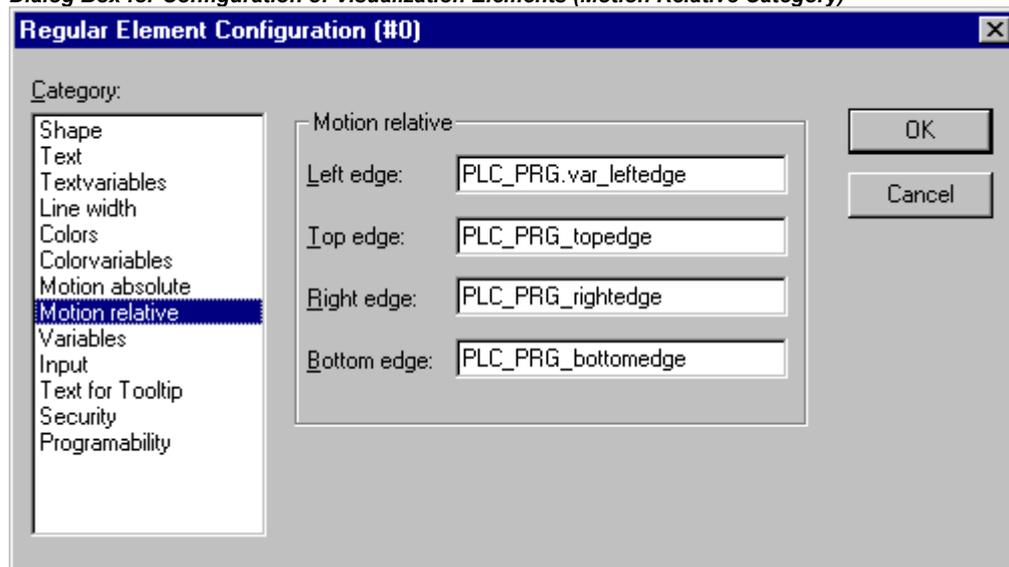
In the dialog for configuring visualization elements in the **Motion Relative** category, you can assign variables to the individual element edges. Depending on the values of the variables, the corresponding element edges are then moved. The easiest way to enter variables into the fields is to use the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

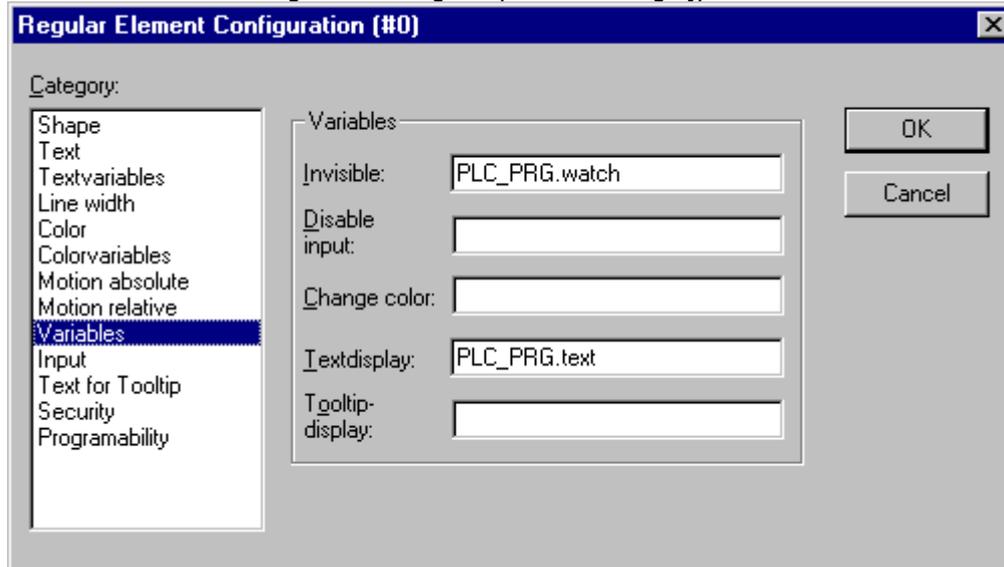
**Note:** In online mode the variables which are set in the 'Motion absolute' dialog will override the values of structure components which additionally might be used to define the same property ('Programmability').

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)*

## Variables

Visualization Element Configuration Dialog Box (Variables Category)



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.

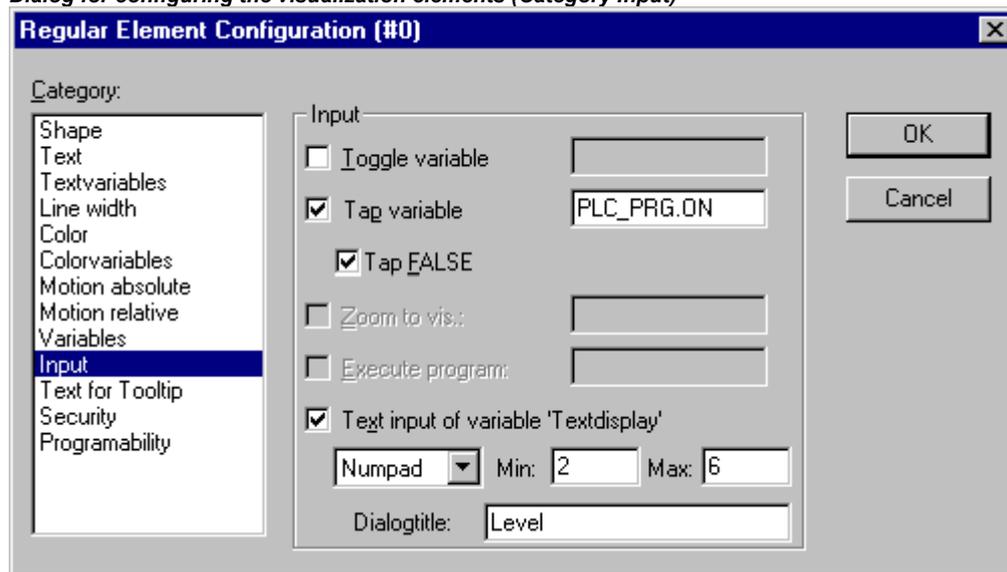
**Textdisplay:**

- 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), 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 (chapter 2.4.2) resp. of the Language switching in a visualization (chapter 3)
- 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)*



**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 mouse-key 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 instance 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> ).  
During compilation of the visualization it will be checked, whether the text matches with one of the defined replacement values defined in the placeholder list, if it does not, a warning will be output.

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, for implicit (system) variables see 10) 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. Regard that for compiler versions < V2.3.0.7, and if the library SysLibStr.lib is not included in the project, this will only work if the names of the visualization objects are specified in capital letters (see chapter 2.1).  
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. Press button "..." to get the dialog **Configure programs** where you can select the desired commands (Add) and arrange them in the desired order (Before, After). For example there is a command for language switching in a visualization. See further below chapter 'Special input possibilities for operating versions'.

---

**Note:** This feature especially is important if the visualization will be the only operating interface of a system (pure **operating version**).

---

**Text input of variable 'Textdisplay':** If this option is activated, in online mode you can enter text in an edit field in this visualization element. This value upon pressing <Enter> will 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:**.

---

**Note:** In case of using Target-Visualization regard the possibility to get information on **user inputs via mouse-clicks** by special interface functions and to use that directly in the project. (see chapter 9)

---

## ToolTip

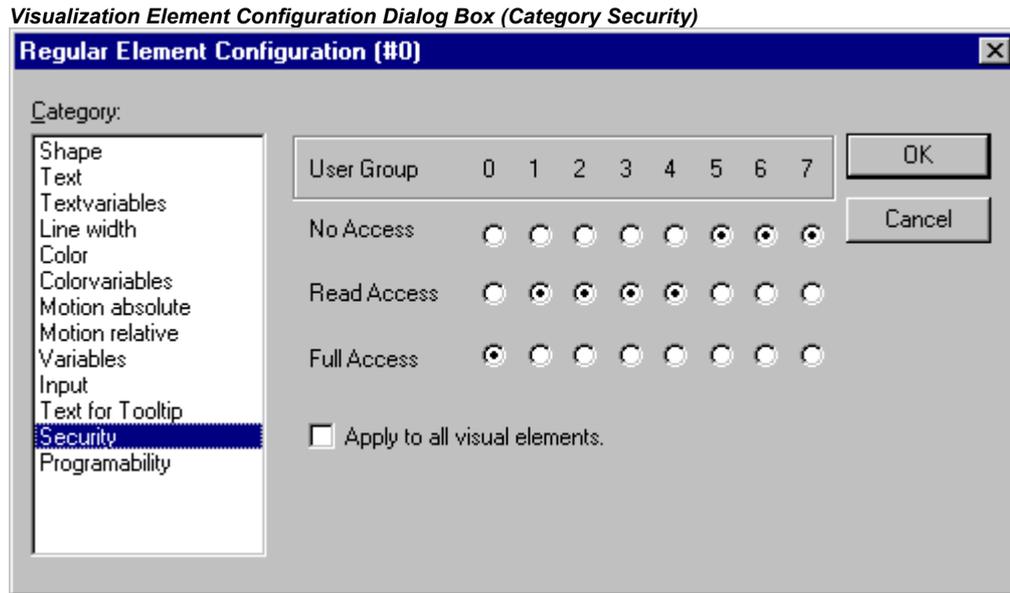
The dialog Text for Tooltip offers an input field for text which appears in a text field as soon as the mouse cursor is passed over the object in online mode. The text can be formatted with line breaks by using the key combination <Ctrl> + <Enter>.

## 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:

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

<b>No Access</b>	Element will not be visible
<b>Read Access</b>	Element will be visible but not operable (no inputs allowed)
<b>Full Access</b>	Element is not visible and not operable



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 a 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.

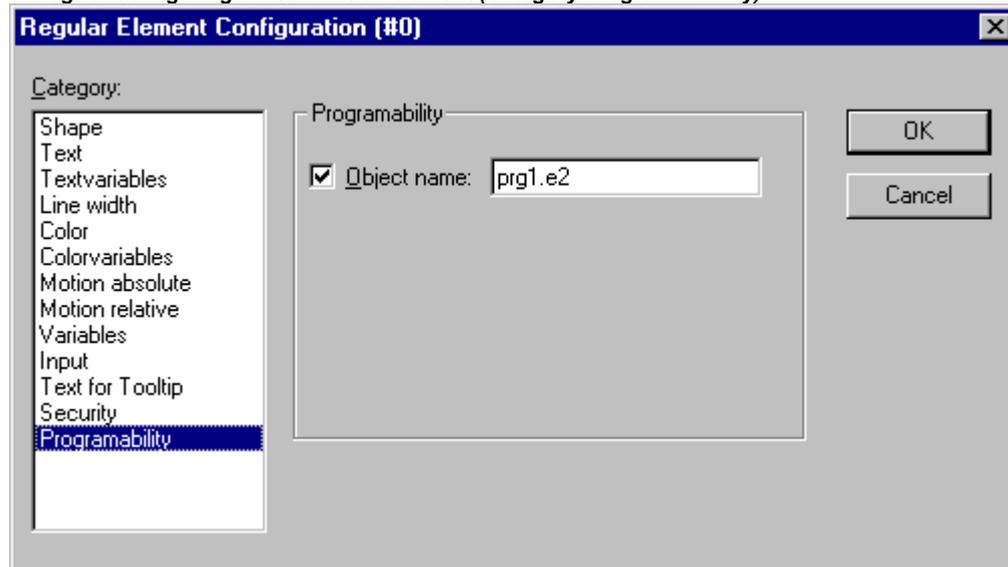
In order to configure the element properties by using a structure variable, do the following:

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 implicitly 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 Programmability)



### The structure 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:
<b>nXOffset : INT;</b>	Shift element in X-direction	<code>vis1.nXOffset:=val2;</code> (element is set to position X=val2)	- Cat. Motion absolute: X-Offset
<b>nYOffset : INT;</b>	Shift element in Y-direction	<code>vis1.nYOffset:=22;</code> (element is set to position Y=val2)	- Cat. Motion absolute: Y-Offset
<b>nScale : INT;</b>	Change of the size	<code>vis1.nScale:=plc_prg.scale_var;</code> (element size changes linear with change of value of plc_prg.scale_var)	- Cat. Motion absolute: Scaling
<b>nAngle : INT;</b>	Rotating element around its center	<code>vis1.anglevar:=15;</code> (element rotates clockwise by 15 )	- Cat. Motion absolute: angle
<b>blnvisible : BOOL;</b>	Element is visible / invisible	<code>vis1.visible:=TRUE;</code> (element is invisible)	- Cat. Color: No color inside + No frame color - Cat. Colorvariables: Fillcolor + Framecolor
<b>stTextDisplay : STRING;</b>	Text is displayed in element	<code>vis1.TextDisplay:='ON / OFF';</code> element is inscribed with this text	- Cat. Text: entry at Content'
<b>bToggleColor : BOOL;</b>	color change when toggling between TRUE and FALSE	<code>vis1.bToggleColor:=alarm_var;</code> (As soon as alarm_var gets TRUE, the element gets the color defined via the components dwFillColorAlarm, dwFrameColorAlarm resp. via the settings	- Cat. Input: Toggle variable + - Cat. Variables: change color

Component (+Data type)	Effect	Example (the Object Name "vis1" has been defined for the element)	corresponding entries in configuration dialog:
		in category 'Colorvariables' or 'Color'.	
<b>bInputDisabled: BOOL;</b>	if FALSE: Inputs in category 'Input' are ignored	<code>vis1.bInputDisabled:=FALSE;</code> (no input is possible for this element)	- Cat. Variables: Disable Input'
<b>stTooltipDisplay: STRING;</b>	Text of the tooltip	<code>vis1.stTooltipDisplay:='Switch for .....';</code>	- Cat. Text for Tooltip: 'Entry in Content:'
<b>dwTextFlags: DWORD;</b>	Text position: 1 left justified 2 right justified: 4 centered horizont. 8 top 10 bottom 20 centered vertically <b>Note:</b> Always set a horizontal <u>and</u> a vertical position (addition of values)!	<code>vis1.dwTextFlags:=24;</code> (Text will be placed in the center of the element (4 + 20)	- Cat. Text: Horizontal and Vertical options - Cat. Textvariables: Textflags
<b>dwTextColor : DWORD;</b>	Text color (definition of colors see subsequent to this table)	<code>vis1.dwTextColor := 16#00FF0000;</code> (Text is blue-colored)	- Cat. Text: Font   Color - Cat. Textvariables: Textcolor
<b>nFontHeight : INT;</b>	Font height in Pixel. should be in range 10-96	<code>vis1.nFontHeight:=16;</code> (Text height is 16 pt)	- Cat. Text: Font   Grad' - Cat. Textvariables: Font heighth
<b>dwFontFlags : DWORD;</b>	Font display. Available flags: 1 italic 2 bold 4 underlined 8 cancelled + combinations by adding values	<code>vis1.dwFontFlags:=10;</code> (Text is displayed blue and cancelled)	- Cat. Text: Schrift   Schriftschnitt - Cat. Textvariables: Fontflags
<b>stFontName : STRING;</b>	Change font	<code>vis1.stFontName:='Arial';</code> (Arial is used)	- Cat. Text: Schrift   Schriftart - Cat. Textvariables: Fontname
<b>nLineWidth : INT;</b>	Line width of the frame (pixels)	<code>vis1.nLWidth:=3;</code> (Frame width is 3 Pixels)	- Cat. Line width
<b>dwFillColor : DWORD;</b>	Fill color (definition of colors see subsequent to this table)	<code>vis1.dwFillColor":=16#00FF0000;</code> (Element ist im "Normalzustand" blau)	- Cat. Color: Color   Inside - Cat. Colorvariables: Inside
<b>dwFillColorAlarm : DWORD;</b>	Fill color as soon as <code>bToggleColor</code> gets TRUE, see above) (definition of colors see subsequent to this table)	<code>vis1.dwFillColorAlarm:=16#00808080;</code> (as soon as Variable <code>togglevar</code> gets TRUE, the element will be displayed grey-colored)	- Cat. Color: Alarm color   Inside - Cat. Colorvariables: Inside Alarm
<b>dwFrameColor: DWORD;</b>	Frame color (definition of colors see subsequent to this table)	<code>vis1.dwFrameColor:=16#00FF0000;</code> (Frame is blue-colored)	- Cat. Color: Color   Frame - Cat. Colorvariables: Frame

Component (+Data type)	Effect	Example (the Object Name "vis1" has been defined for the element)	corresponding entries in configuration dialog:
<b>dwFrameColorAlarm: DWORD;</b>	Fill color as soon as bFrameColor gets TRUE, see above (definition of colors see subsequent to this table)	<code>vis1.dwFrameColorAlarm:=16#00808080;</code> (as soon as Variable <code>vis1.bToggleColor</code> gets TRUE, the frame will be displayed grey-colored)	- Cat. Color: Alarm color   Frame - Cat. Colorvariables: Frame Alarm
<b>dwFillFlags: DWORD;</b>	Color, as defined by the color variables, can be displayed or ignored 0 = show color >0 = ignore setting	<code>vis1.dwFillFlags:=1;</code> (element gets invisible)	- Cat. Color: No color inside + No frame color - Cat. Colorvariables: Fillflags
<b>dwFrameFlags: DWORD;</b>	Display of frame: 0 full 1 dashed ( --- ) 2 dotted ( . ) 3 dash-point ( _._._ ) 4 dash-point-point ( _._. ) 8 blind out line	<code>vis1.FrameFlags:=1;</code> (Frame will be displayed as dashed line)	- Cat. Colorvariables: Frameflags

#### 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
```

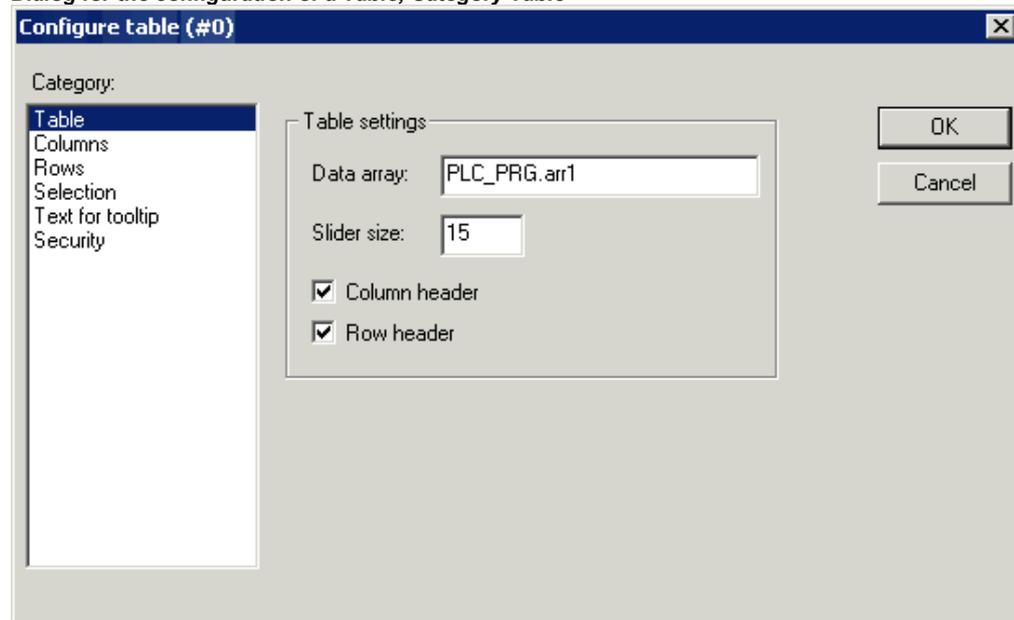
## Table

As soon as a table is inserted for the purpose of visualization of an array, the dialog **Configure Table** will be opened. 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:

Regard that the options in the color and font selection dialogs might deviate from the default depending on the current target (see note in chapter 2.4.1).

### Category Table:

#### Dialog for the configuration of a Table, Category Table



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.

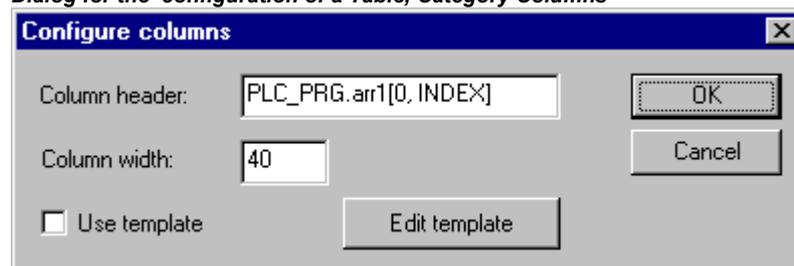
**Slider size:** Insert here the desired height of the slider which will appear at the bottom of the table element if the display of the array columns exceeds the element width.

**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:

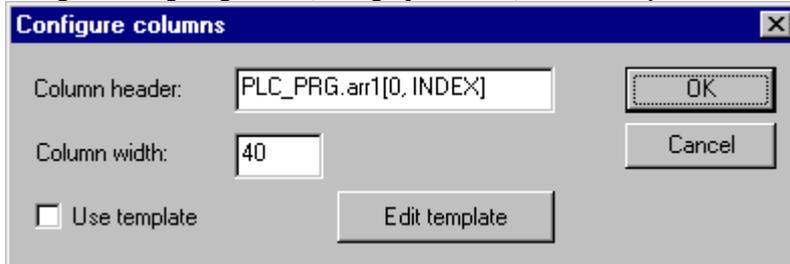
Here you define the **table elements**. In the left window you get a list of all elements, which are handled in the array per index. In case of an array of a structure these would be the structure components.

#### Dialog for the configuration of a Table, Category Columns



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 concerning 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**



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 Further on 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 button **Edit template** for the currently marked column, you can modify the parameters of the fields of this column, e.g. the line width, text input etc. The template affects all fields of the current column and can be edited via the known configuration dialog for a visualization element.

If you want to configure only one or several particular fields of the column, you can use the following placeholders determining the desired row and column: \$ROWCONST\$, \$COLCONST\$, INDEX. (INDEX has the same effect as \$ROWCONST\$).

**Examples for the use of placeholders in column-templates:**

Example 1:

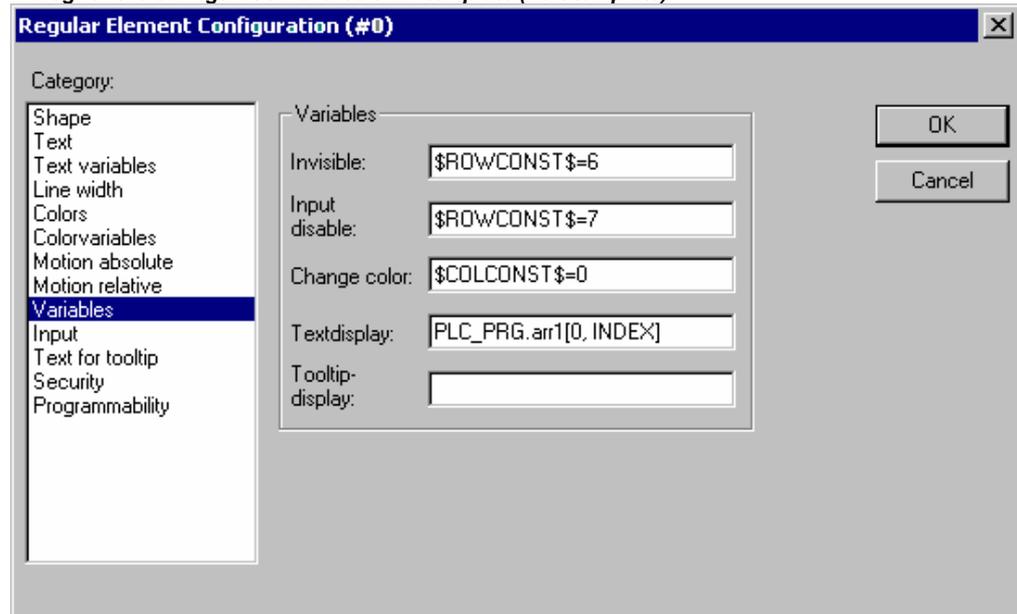
You want to visualize an array "arr1 [0..2] of BOOL" (table with 1 column) and you want, that in online mode by a mouse-click on a table cell the cell gets red-colored and the corresponding array element will be toggled and vice versa.

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].

Example 2:

The following template configuration has been created for a table column representing Index „0“ of an array:

*Dialog for the configuration of a column-template (Edit template)*

This means the following: For the concerned table column

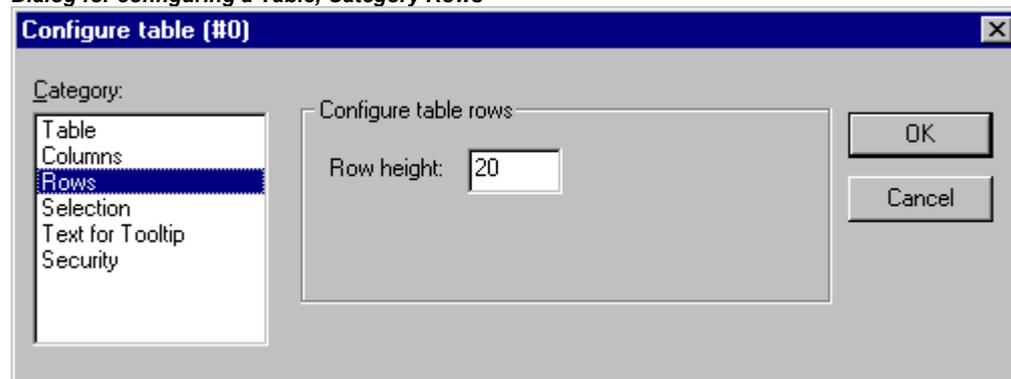
- the field in line 6 (line number, index) is invisible
- in the field in line 7 no edit field can be opened
- in all fields of column 0 the color will change to alarm color. The other fields of the column remain in base color.
- Always entered automatically, but of course changeable is the textdisplay configuration, which combined with the "%s" entry in category "Text" effects that the corresponding variable value will be displayed in the table field.

Placeholder entries in the configuration fields can be connected with „AND“ resp. “OR“; Example: "\$ROWCONST\$=1 OR \$ROWCONST\$=3" makes that both fields will get the respective setting.

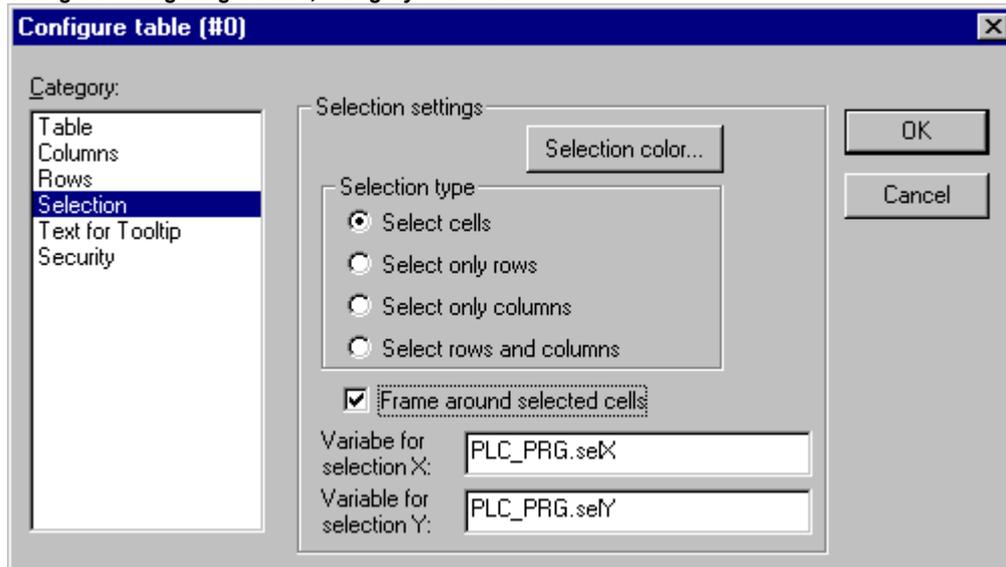
The use of the currently configured template can be activated or deactivated: Option **Use template**.

### Category Rows:

#### *Dialog for configuring a Table, Category Rows*



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

Category Selection:*Dialog for configuring a Table, Category Selection*

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

**Selection color:** Press this button to define a color for selected cells. You get the standard dialog for choosing a color or a target-specific color selection list (see note in chapter 2.4.1).

**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:

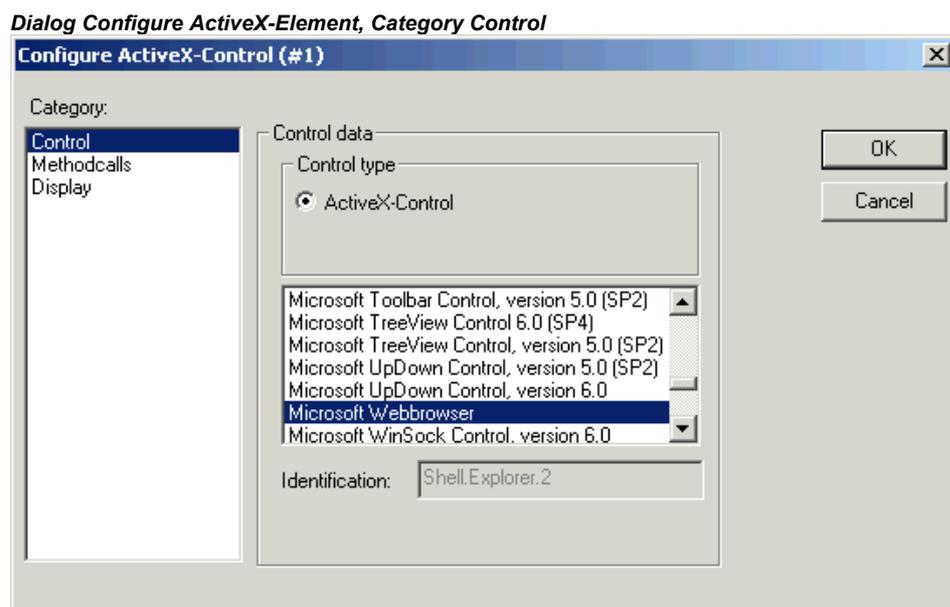
	Number	Value	Text
1	0	TRUE	text1
2	33	TRUE	text2
3	55	FALSE	abc
4	0	FALSE	
5	0	TRUE	

### ActiveX-Element

The ActiveX-Element serves for displaying a passive ActiveX Control within a visualization. The element is usable on Windows32 based systems in CoDeSys HMI and in the CoDeSys Target-Visualization.

The configuration dialog is opened by a double-click on the inserted element and offers three sub-dialogs, for selecting the control type, for defining method calls and for the configuration of the display:

Category Control:

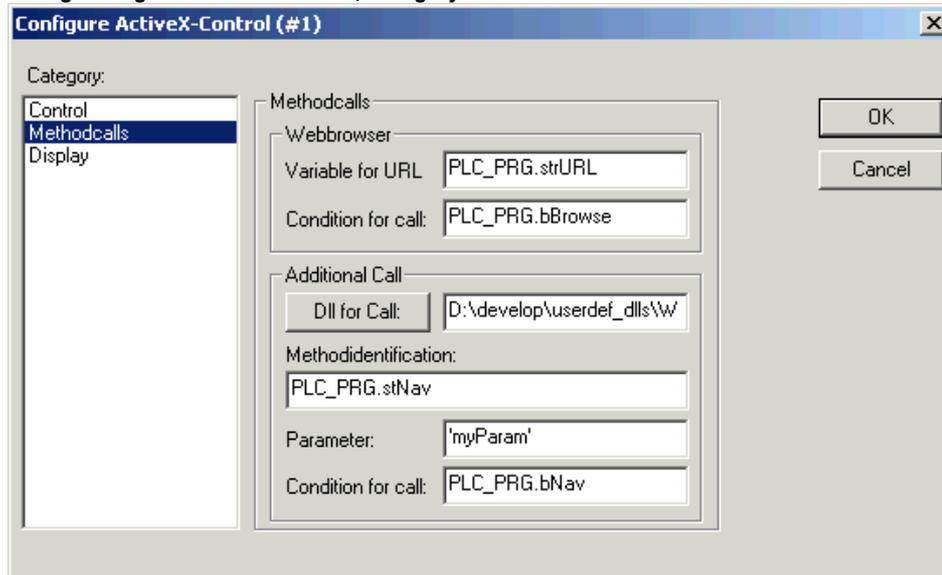


In this dialog you can mark the desired ActiveX Control in the selection list offering all ActiveX Controls which are registered on your computer.

### Category Methodcalls:

Here you configure the method calls for the chosen ActiveX Control:

#### *Dialog Configure ActiveX-Element, Category Methodcalls*



#### Webbrowser:

These input fields only can be edited if you are configuring a control element which supports the IWebBrowser interface (e.g. Internet Explorer or Mozilla Browser). In this case CoDeSys can directly call the method Navigate (other method calls must be controlled via a user defined Dll, see below 'Additional Call').

Enter an URL as a parameter value in field **Variable for URL** (Input as a string between inverted commas) resp. a project variable of type STRING defining an URL. The browser will be called as soon as the variable entered in field **Condition for call** gets TRUE (rising edge). If no call condition is defined here, in the Target-Visualization the browser will be called in each cycle of the visualization task!

#### Additional Call:

Via a user defined Windows-Dll you can define method calls for the ActiveX Control in order to control the behaviour of the control at a call. For this purpose you must enter the path of the Dll in the field at **Dll for Call**. If you press the button you get the dialog for opening a file where you can browse for a Dll. If the file is in the visualization files directory as defined in the project options, only the file name will be entered, otherwise the complete file path.

**Attention:** If the Dll is to be used on a runtime system with a Target-Visualization, it must explicitly be copied there. When the Control is called in the Target-Visualization, only the filename contained in the path will be regarded.

The Dll is called as soon as the variable defined below in Condition for call gets TRUE (rising edge). If no condition is specified, in the Target-Visualization it will be called in each cycle of the visualization task!

Regard the following when creating an appropriate Dll:

The Dll must export a method "ExecuteActiveXCall" with this function prototype:

```
void ExecuteActiveXCall(IUnknown* pUnk, char* pszId, char* pszParam,
char* pszReturnBuffer, int nReturnBufferSize, DWORD* pdwReturnFlag);
```

The function will be called with the following parameters which can be defined in the configuration dialog:

*pszId*: string resp. string variable specified in field Methodidentification

*pszParam*: value specified in field Parameter

The parameter *pUnk* allows a query of further Com(ActiveX-)interfaces. With these interfaces you can call any Method on your ActiveX-Control with any parameters packed in a string!

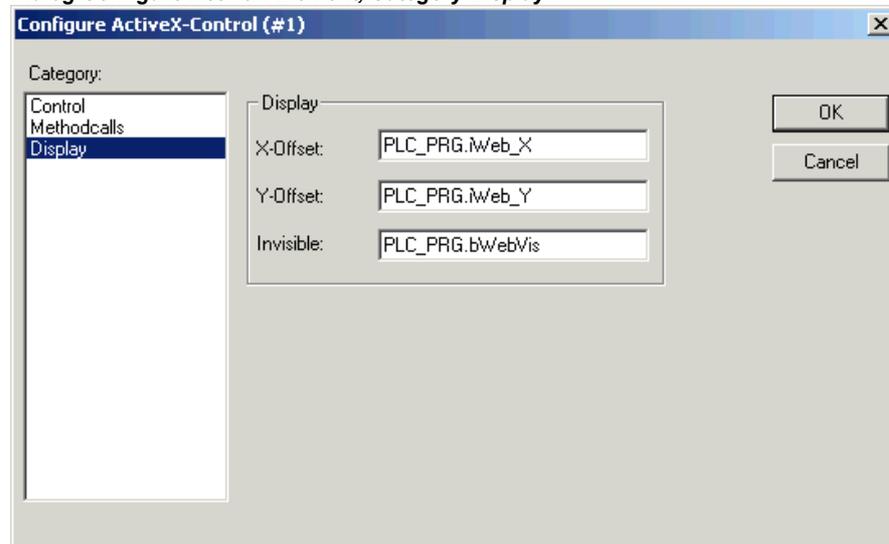
The parameters *pszReturnBuffer*, *nReturnBufferSize* and *pdwReturnFlag* currently are not used.

For an example of a Dll source file see Appendix B.

Category Display:

In this dialog you can specify the variables defining the position (X-Offset, Y-Offset, see description of configuration category 'Motion absolute')and visibility of the control element (Invisible, see description for configuration category 'Variables').

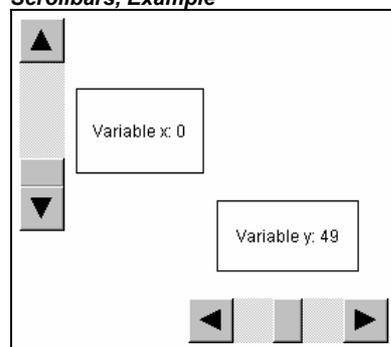
**Dialog Configure ActiveX-Element, Category Display**



**Scrollbar**

The scrollbar element can be used to modify the value of a variable (or vice versa). The slider position will correspond to the value of an assigned variable within a defined value range.

**Scrollbars, Example**



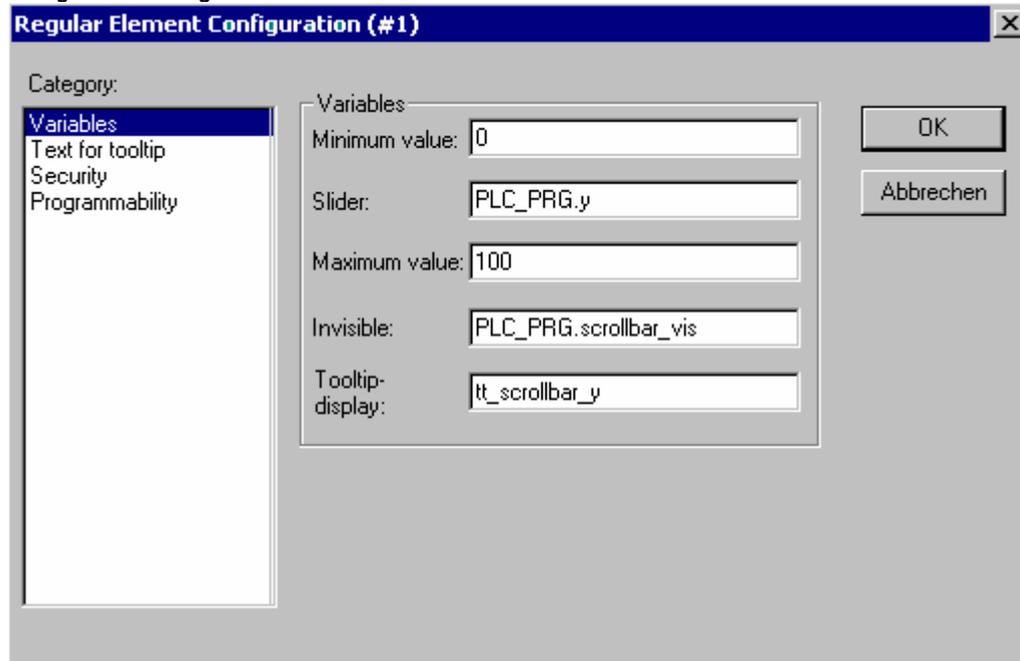
The user can move the slider in online mode by clicking on the slider and then moving it by moving the mouse - or alternatively by clicking on one of the arrow symbols on the scrollbar, whereby each mouse-click will move the slider (and thus the variable value) by 1 in the

respective direction: If the slider is moved to the right resp. up, the value will be increased, if the slider is moved to the left resp. down, the value will be decreased.

A vertical or horizontal display of the scrollbar can be reached by a respective shaping of the element (see 'Insert' 'Scrollbar').

The configuration dialog gets opened by a double-click on the element.

#### Configuration dialog for Scrollbar



**Slider:** Enter here the path of the project variable, the value of which should correspond to the slider position between the given Minimum value and Maximum value (e.g. "PLC\_PRG.ivar").

So, when the slider gets moved in online mode, the value of the variable will change. Vice versa, if the variable gets modified by any other input the slider would get moved correspondingly.

**Minimum value, Maximum value:** Boundaries for the value range which can be displayed for the variable assigned to the slider. In horizontal scrollbars the minimum value corresponds to the left-most position of the slider, in vertically scrollbars to the bottom-most position. The values can be entered directly or via a project variable (e.g. "0", "200", "PLC\_PRG.minvar").

**Invisible:** The visibility of the element in online mode can be controlled dynamically, if here an appropriate boolean variable is entered, e.g. "PLC\_PRG.bScrollbar\_vis".

A static definition is also possible ("TRUE", "FALSE" resp. "0", "1"). Default: visible.

**Tooltip-display:** If "dynamic texts" should be used for the tooltip text, here the ID of the prefix-ID-combination must be entered, which uniquely references the desired text in a specific language file (xml-format). You can enter the ID directly (string, e.g. "tt\_scrollbar1") or via a project variable of string format which gives the ID (e.g. "PLC\_PRG.tt\_ID"). Concerning this see the help pages on Tooltip and generally on Language Switching.

## Meter

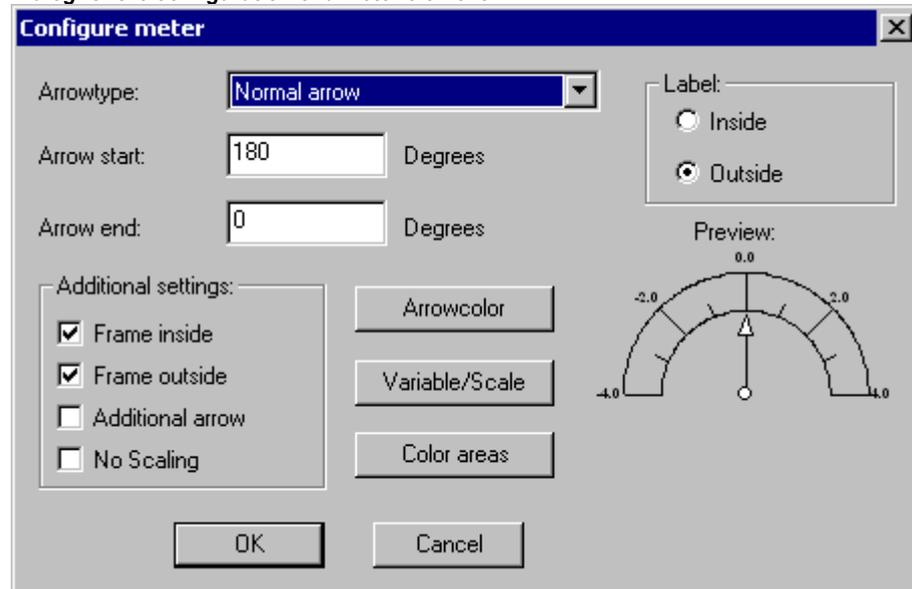
The dialog for configuring a Meter element will open automatically as soon as you insert the element into a visualization object. A **Preview** is part of the dialog, immediately showing how the element will look as a result of the currently set parameters:

**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).

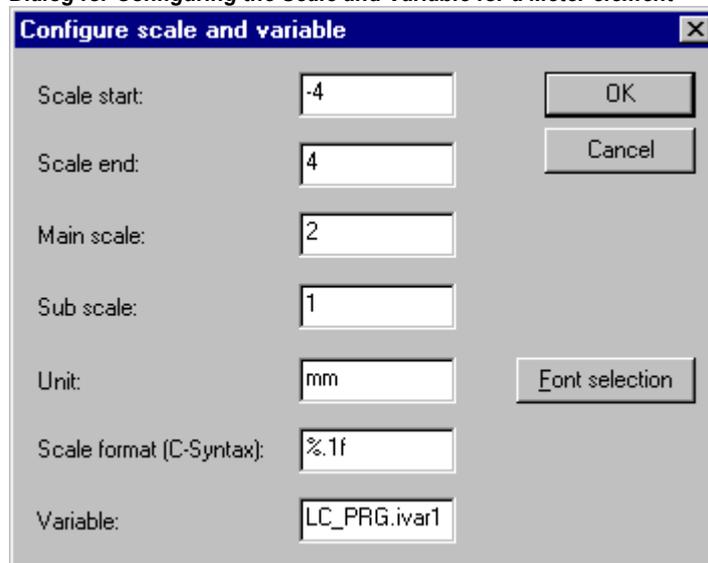
**Arrow color:** This button opens the standard dialog for choosing a color or a target-specific color selection list (see note in chapter 2.4.1). Define the color of the pointer.

*Dialog for the configuration of a Meter element*



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

*Dialog for Configuring the Scale and Variable for a Meter element*



**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 addition 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.

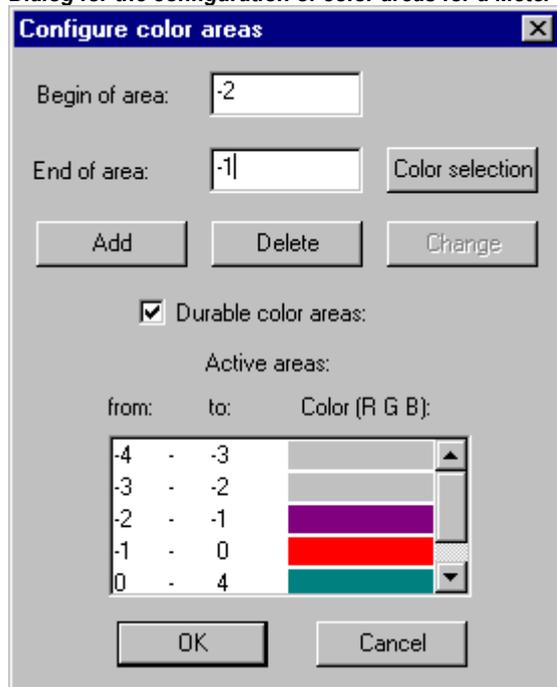
**Begin of area, End of area:** Insert here the start and end values of the scale partition which should get the color defined in the following:

**Color selection:** This button opens the standard dialog for choosing a color or a target-specific color selection list (see note in chapter 2.4.1). 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*



Additional settings:

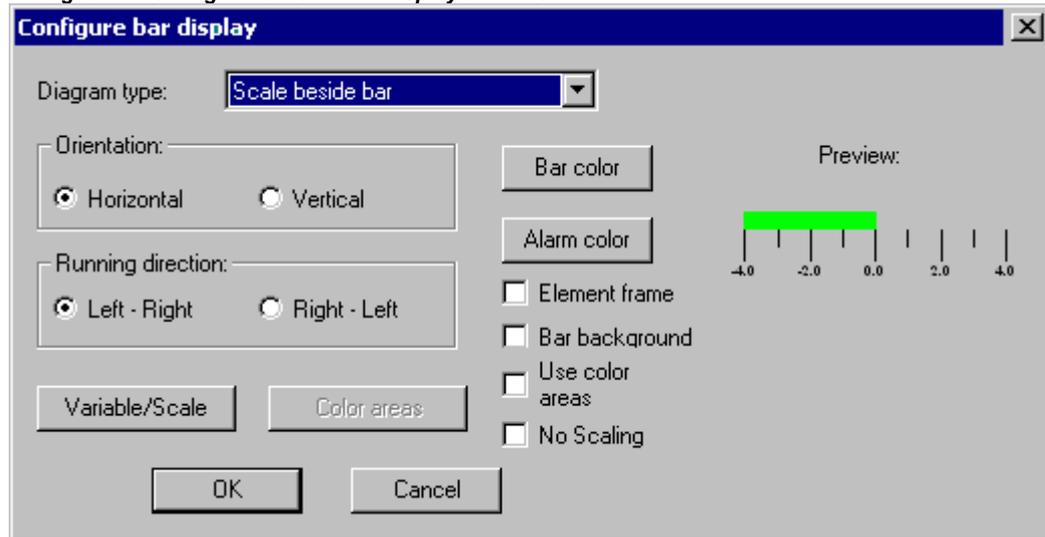
**Frame inside, Frame outside:** If one or both of these options is/are activated, an inner or outer frame will be added to the scale arc.

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

**No Scaling:** If this option is activated, the inserted element cannot get resized.

## Bar Display

*Dialog for the configuration of a Bar Display element*



This dialog will be opened as soon as you insert a Bar Display 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 or a target-specific color selection list (see note in chapter 2.4.1). 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 **Conditions 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 dialog or a target-specific color selection list (see note in chapter 2.4.1) for choosing the 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).

**No Scaling:** If this option is activated, the inserted element cannot get resized.

**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 height.

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:

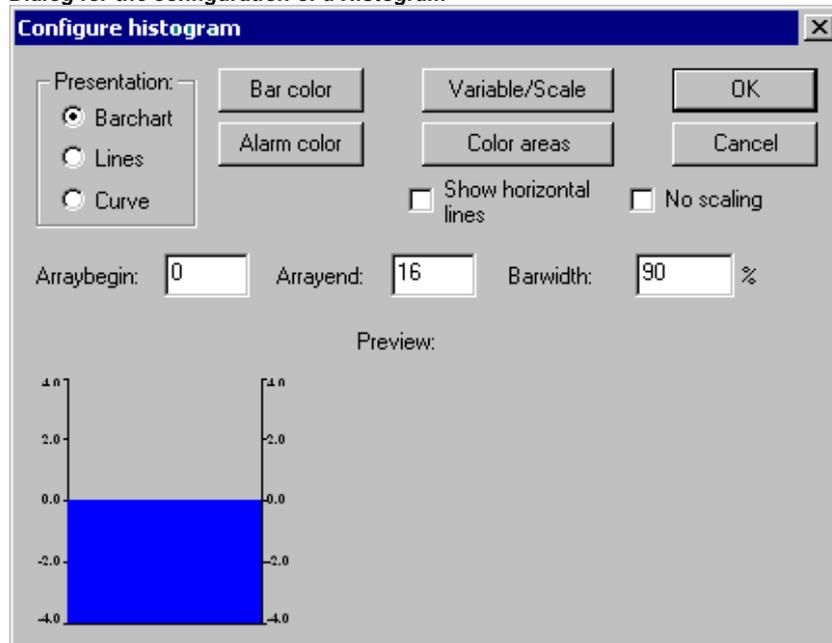
**Presentation:** Activate one of the options **Barchart** or **Lines**.

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

**No Scaling:** If this option is activated, the inserted element cannot get resized.

**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 **Conditions 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 or a target-specific (see note in chapter 2.4.1) for choosing the alarm color. Close both dialogs with OK in order to confirm the settings and to return to the main dialog for configuring the histogram.

*Dialog for the configuration of a 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 or a target-specific dialog (see note in chapter 2.4.1) for choosing a color. Define a color for the bar in normal state (no alarm).

Define which range of the array should be displayed:

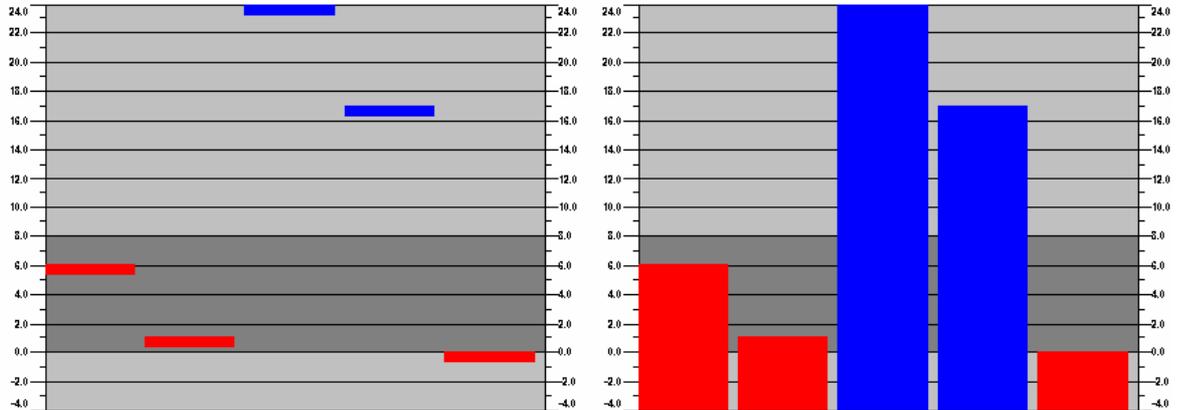
**Arraybegin:** First array element to be displayed (Index).

**Arrayend:** Last array element to be displayed (Index).

**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. Further on 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 being in alarm state:

**Alarm table**

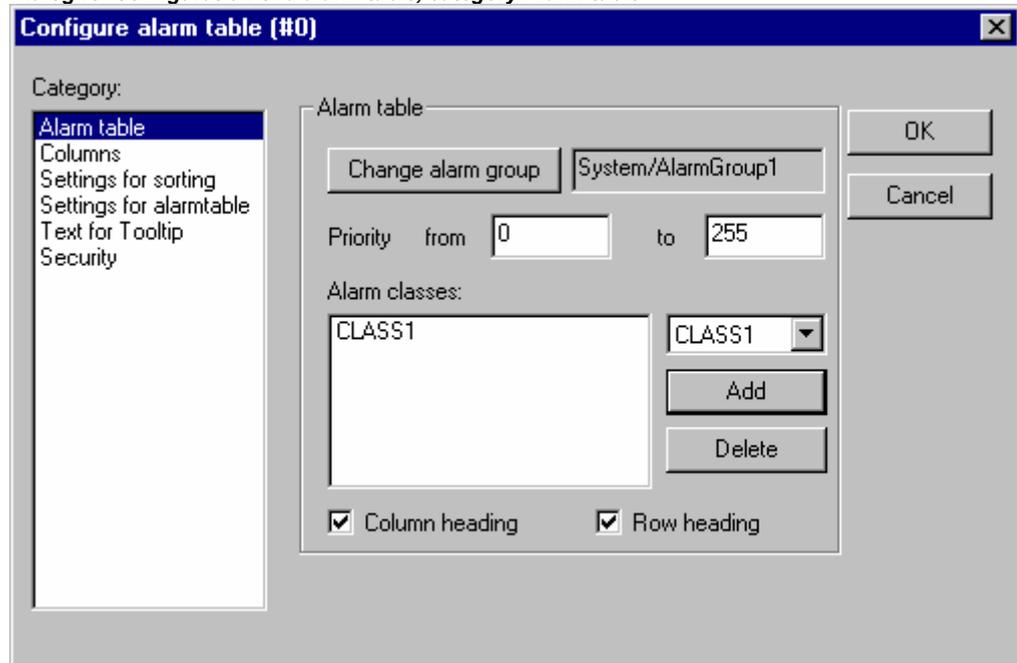
The element 'Alarm table' is used to visualize alarms, which must be defined before in the CoDeSys Alarm configuration.

As soon as the element gets inserted in the visualization object, the dialog 'Configure alarm table'. Besides the known categories for configuration of tooltip and security the settings described in the following concerning display and selection in the table can be made.

Regard that the options in the color and font selection dialogs might deviate from the default depending on the current target (see note in chapter 2.4.1).

Category Alarm table:

*Dialog for configuration of a alarm table, category Alarm table*



Define what you want to get displayed in the alarm table:

**Change alarm group:** Press this button to get the selection tree of the alarm configuration, which offers all alarm groups currently defined. Choose the desired group (which even may contain just one alarm).

**Note:** The name of the alarm group, as it is displayed here (e.g. "System/Alarmgroup1") can be used as PREFIX in a XML-file for dynamic texts in order to get **dynamic language switching** for the message texts in the alarm table; the associated ID in this must be the number of that line in the Alarm group configuration table, that defines the respective message text.

**Priority:** Define the priority for which you want to get displayed all alarms. Permissible range: 0 to 255.

**Alarm classes:** Mark a class which you want to get displayed and press button **Add** to add the class to the list in the window 'Alarm classes'. Do this for all required classes. In order to remove a marked entry from the alarm classes window press button **Delete**.

Activate options **Column heading** resp. **Row heading**, if the headings should be displayed in the alarm table.

Category Settings for sorting:

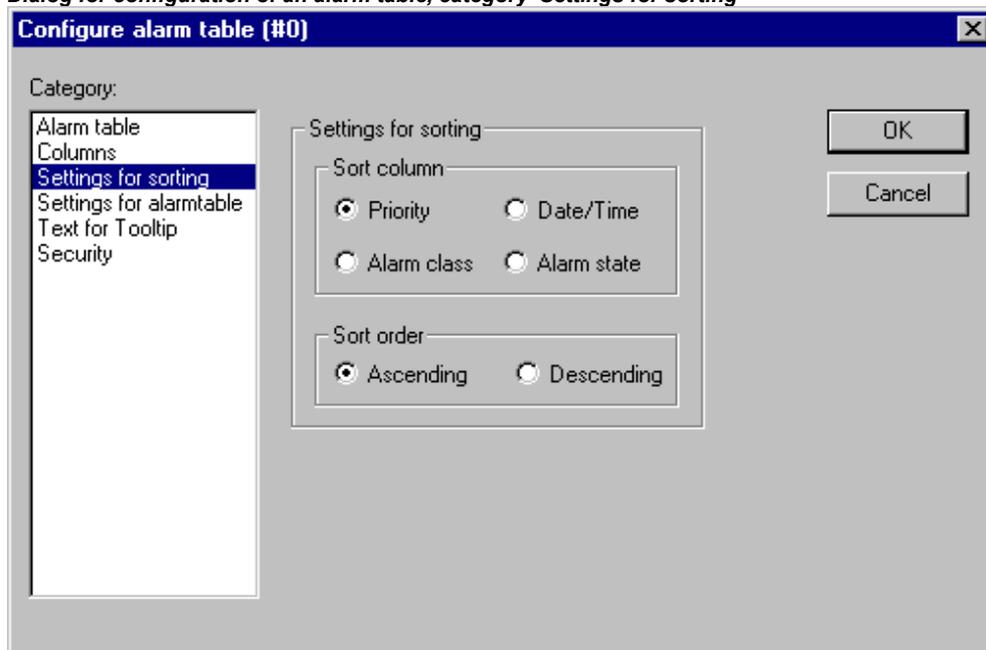
Define here according to which criteria the alarm table should be sorted:

**Sort column:** Sorting according to Priority, Alarm class, Date/Time or Alarm state

**Sort order:** Ascending or Descending; Example: Ascending according to priority means that the table will start with alarms of priority 0 (if available), followed by higher numbered priorities.

**Note:** When using in a Target-Visualization this settings are not regarded in the display of the alarm history. There the sorting always is descending according to the date.

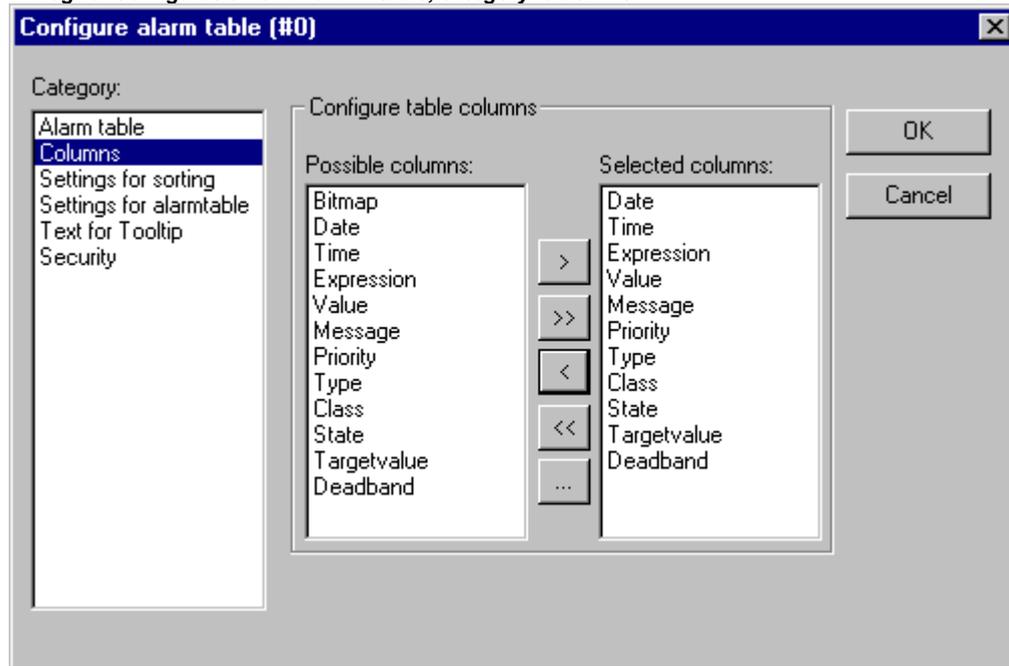
*Dialog for configuration of an alarm table, category 'Settings for sorting'*



**Category Columns:**

Define here, which of the columns (alarm parameters) should be displayed in the alarm table: The parameters are defined -- except for date and time (alarm is coming) and alarm state (see CoDeSys User Manual, Alarm Configuration) in the Configuration of the alarm groups: **Bitmap, Date, Time, Expression, Value, Message, Priority, Type, Class, State, Target value** (for alarm types DEV+ and DEV-), **Deadband**.

Dialog for configuration of an alarm table, category 'Columns'



Using the buttons ">", ">>", you can take single resp. all parameters from the left to the right window. The selection defined in the right window will be displayed in the alarm table. Using the buttons "<" resp. "<<" entries can be removed from the selection.

For each column you can open the dialog 'Configure columns' by a double-click on the entry in the right window. In this dialog **Column header** and **Column width** can be defined.

Category Selection settings for alarm table:

Define here some settings for the display for the chosen table fields:

**Selection color:** This button opens the standard dialog for choosing a color. Define the color in which selected fields should be displayed.

**Row height:** Height of the table rows in Pixel.

**Slider size:** Height of the slider (Pixel) at the bottom of the table.

**Frame around selected rows:** If this option is activated, selected table rows will get a frame.

**Show statusline:** If this option is activated, below the alarm table a status bar will be displayed providing the following buttons for the operation in online mode:

**Acknowledge:** All alarm entries marked in the alarm table get acknowledged.

**Acknowledge all:** All alarm entries listed in the alarm table get acknowledged.

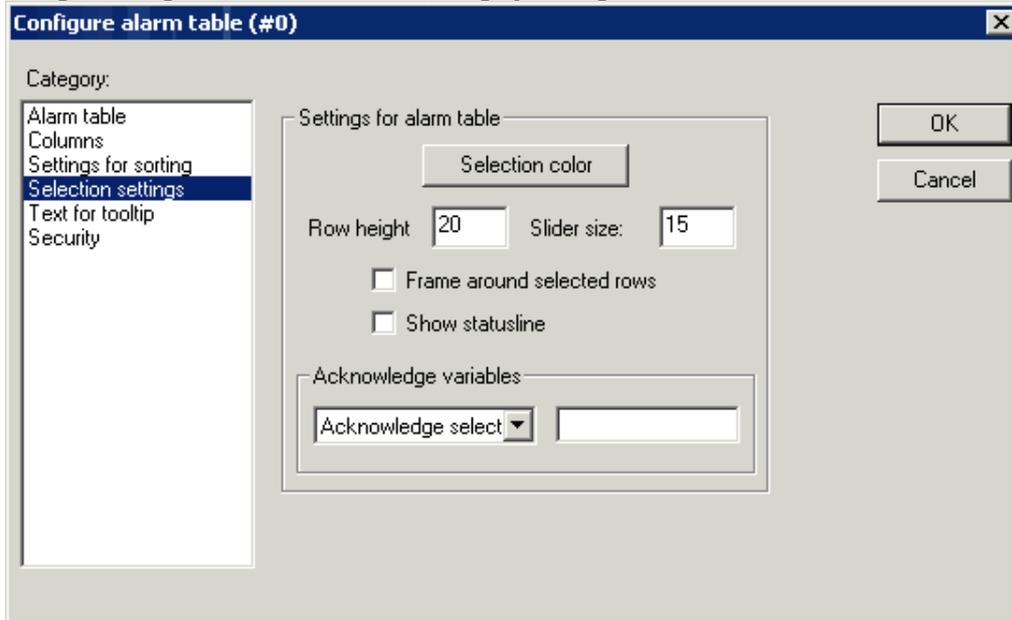
**History:** If this button is pressed, instead of the current status of the alarms the table will show a complete list of all events which have occurred up to now (all transitions between any alarm stati). In this list no acknowledgement is possible! Any new events will be added currently.

If you have defined a record file, also there you will find this history for all alarm classes, for which the action 'Save' has been activated.

**Start:** cancels **Stop** (see below)

**Stop:** The current update of the list with newly occurring events will be stopped until it is restarted by pressing button 'Start'.

Dialog for configuration of an alarm table, category 'Settings for alarmtable'



**Acknowledge variables:** This option is only available as long as you have not chosen option 'Show statusline' (see above). If it is activated, the functions described above for the status string buttons can get controlled by variables. In order to define these variables, choose a function from the selection list and enter a project variable in the assigned edit field. Thus for example the acknowledgement of all alarms in online mode can be done by a rising edge of the assigned variable.

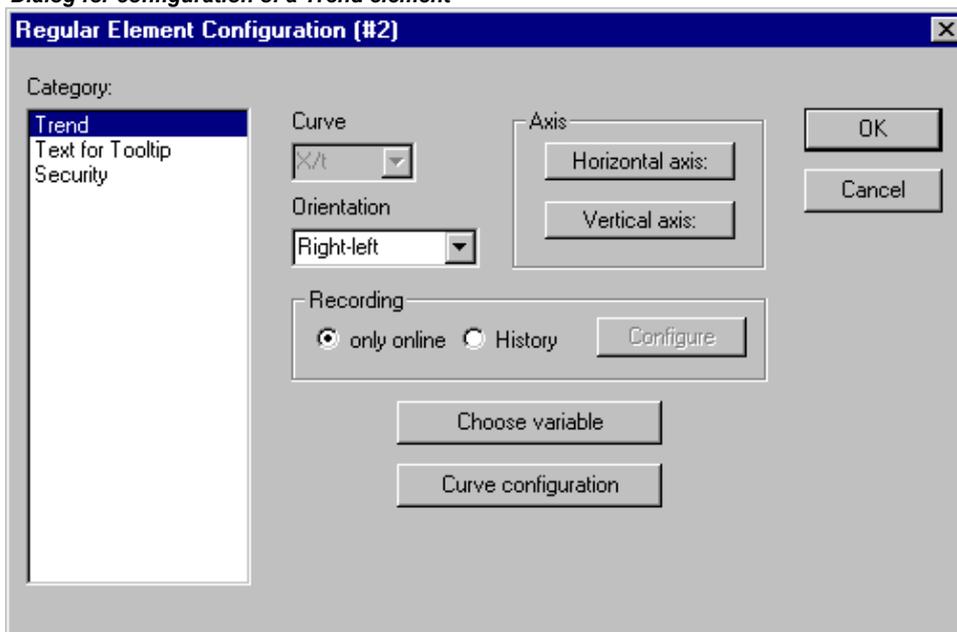
**Note:** The **Web-Visualization** might be configured in a way, that in **online mode** a tooltip will display the full string of a text entry, which is only partially visible in the alarm table.

## Trend

The Trend element can be used to log the time dependent behaviour of variable values in the online mode. It can be compared with the trace functionality. The online presentation is done in a diagram, in case of logging to a text file each of the values is written to a separate line.

In the dialog for configuring visualization elements in category 'Trend' you can do the following settings:

Dialog for configuration of a Trend element



**Curve:** X/t, horizontal axis = time axis, vertical axis = scale of values

**Orientation:** Left-right or Right.-left: The latest value will be displayed on the left/right side;

**Axis:**

Horizontal Axis:

*Dialog for configuration of the horizontal axis in the trend element*

**Division lines:** Activate option 'visible', if vertical division lines should be displayed which are elongating the scale marks. In this case define the 'Scale': The given number defines the interval between the division lines on the horizontal axis. Type (normal \_\_\_\_, dashed \_ \_ \_, dotted ....., dashdotted \_ . \_ .) and color of the lines can be defined in dialogs which will open when you perform a mouse-click on the corresponding rectangle showing the currently set line type resp. color.

**Scale:** The shown range of the scale is determined by the entry for **Duration**. If here e.g. "T#20s0ms" is defined, the scale will display a period of 20 seconds. The **Main** division and the **Sub scale** division, which will be displayed by the means of long and short marks are to be defined according to the same syntax.

**Degree of accuracy:** Define here (in the standard format for dates, e.g. T#5ms) the interval for displaying the current values of the variables.

**Legend:** Here you define the display of the legend. Via button **Font** the standard dialog for setting the font will be opened. At **Scaling** define the distances between the particular letterings on the scale (e.g. T#4ms, if the scale markings should get a lettering each 4 milliseconds. The lettering will contain the time and/or date, depending on which options are activated. The desired format each can be defined in the field 'format'.

**Variables:** Here you can define project variables, which contain the zoom values resp. offset values for the horizontal scale. For example the offset of the display range of the horizontal axis will be set to "10" as soon as the variable assigned here gets value 10.

**Symbol bar:** If option use is activated, at the bottom of the element a horizontal symbol bar will be added, providing buttons for scrolling and zooming in online mode. The simple arrow buttons will move the displayed range along the time axis step by step, the double arrow buttons will shift it to the end resp. start of the record. The zoom buttons allow a zooming of the horizontal

scale step by step. To get a possibility to restore the original settings concerning zoom and offset, define the vertical symbol bar to get the 'home' symbol.

Vertical axis:

*Dialog for configuration of the vertical axis in the trend element*

**Division lines:** corresponding to the horizontal axis (see above)

**Scale:** Define whether the scale should be displayed at the left or right border of the trend diagram. Choose the **Start** value (lower end) and **End** value (upper end) of the scale as well as the **Main** and **Sub scale** divisions (longer and shorter markings will be displayed in the here defined distances).

**Legend:** Font and divisions; see above, horizontal axis

**Variables:** see above, horizontal axis

**Symbol bar:** see above, horizontal axis, additionally there is a  "home" button for restoring the standard settings concerning zooming and offset of the axes.

**Recording:** Define here whether the trend should be recorded 'only online', i.e. the time dependent behaviour of the variable values will be displayed using the chosen range of the scale, or whether the record should be saved to history file, which can be configured after pressing the button 'History'. The dialog corresponds to that which is used for the configuration of the alarm log file. Regard: If the target-specific option 'Store trend data in the PLC' is activated (see Target Settings, category Visualization), the path defined here will be ignored and the log file will be stored in the download directory of the PLC.

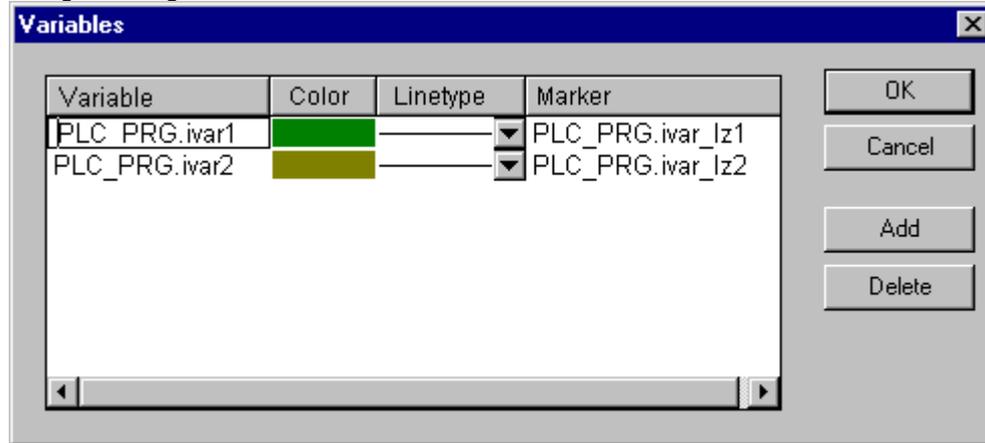
In the log file for each time of measurement a separate line is written which contains the name and the values of all regarded variables. Each line starts with a unique identifier in DWORD format, which is built from the date of measuring.

**Choose variable:** Press this button to get the dialog Variables, where you can configure the variables for which the trend record should be done and how they should be displayed.

Enter a project variable in column **Variable** (mouse-click on the field will open an edit frame). It is recommended to use the input assistant <F2> or Intellisense function.

**Color** and **Line type** for the display of the variable in the record you can define by a mouse-click on the corresponding field in column Color (standard dialog for choosing a color resp. by selecting a line type in the corresponding field of column Line type (normal \_\_\_, dashed \_ \_\_, dotted ....., dashdotted \_ . \_)).

*Dialog for configuration of the variables in the trend element*

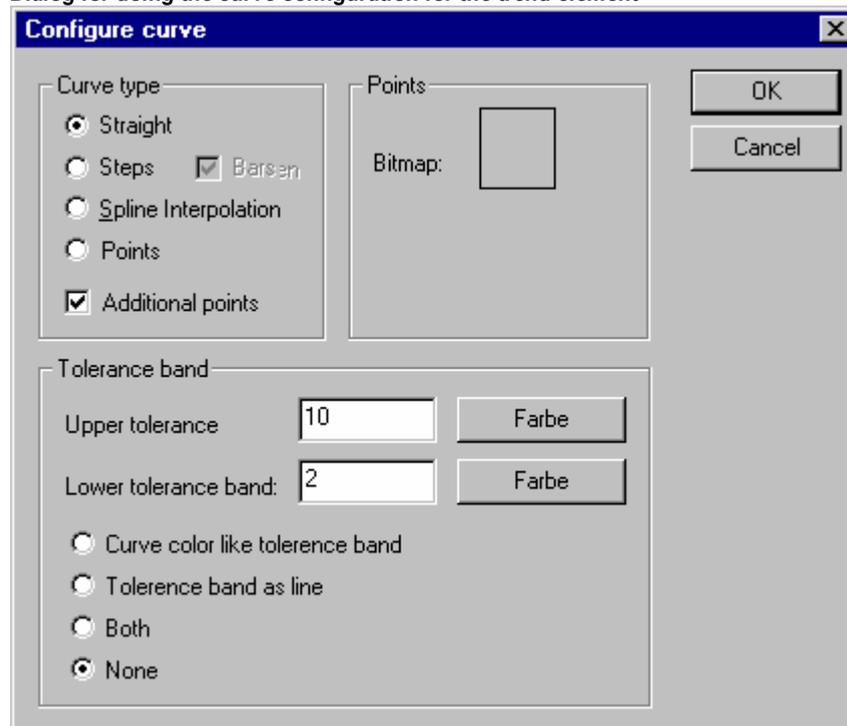


In column **Marker** you can define a variable, which will provide the currently recorded value when you use the marker function in online mode. The marker will be displayed as a little grey triangle in the upper left corner of the diagram. If you click on the triangle and keep the mouse-button pressed, you can shift a vertical marker line along the horizontal time axis. The variable defined as 'marker' then will read the corresponding value from the record curve of the associated project variable.

Do the settings for all variables you want to record. Via button **Add** a further line will be added at the end of the list. A line can be deleted by button **Delete**.

**Curve configuration:** This button opens the dialog Curve configuration. Here some settings concerning the trend curves can be done:

*Dialog for doing the curve configuration for the trend element*



**Curve type:** Select one of the options Straight line, Steps or Points. For the first two types the display of **Additional points** can be defined. For displaying a point a bitmap can be defined, otherwise a filled rectangle (same color as curve) will be used as point symbol. Press the rectangle next to **Bitmap** to get the standard dialog for selecting a bitmap file. Via **Delete** the currently set bitmap can be removed from the configuration.

**Tolerance band:** You can define an upper and lower limit value on the vertical axis to be displayed as a tolerance band. For each band a color (Press the color rectangle to get the selection dialog) can be defined. If the bands should be displayed in online mode, activate option **Tolerance band as line**. If you want the curve to get displayed in the color defined for the respective band as soon as exceeding the tolerance value, activate **Curve color outside tolerance like tolerance band**. Activate **Both** or **None** if you want to activate both or none of the above described display options at a time.

Example: Display of a trend element in online mode:

1. Declaration in program PLC\_PRG:

VAR

```
in: INT;
rSinus:REAL;
rValue:REAL;
rSlider1:REAL; (*for marker function*)
rSlider2:REAL; (*für marker function*)
```

END\_VAR

2. Program part of PLC\_PRG:

```
n:=n+1;
rValue := rValue + 0.01;
rSinus:=SIN(rValue)*50 + 50;
IF n>100 THEN
n:=0;
END_IF
```

3. Configuration of a trend element in a visualization:

Orientation Left-Right, History activated

Horizontal axis: Division lines: T#2s, Duration: T#10s, Main: T#1s, Sub scale: T#500ms, Degree if accuracy: T#200ms, Legend: Time Format ('hh':'mm':'ss'), Scaling T#2s. Symbol bar activated.

Vertical axis: Division lines visible, Scale: 10, dotted, grey; Scale left, Start: 0, End: 100, Main: 10, Sub scale: 5; Legend: 10; Symbol bar activated.

Variables:

1. Variable PLC\_PRG.rsinus, blue line, Marker: PLC\_PRG\_TRD.rSlider1;
2. Variable PLC\_PRG.n, red line, Marker: PLC\_PRG\_TRD.rSlider2

Curve configuration: Straight line, no tolerance band

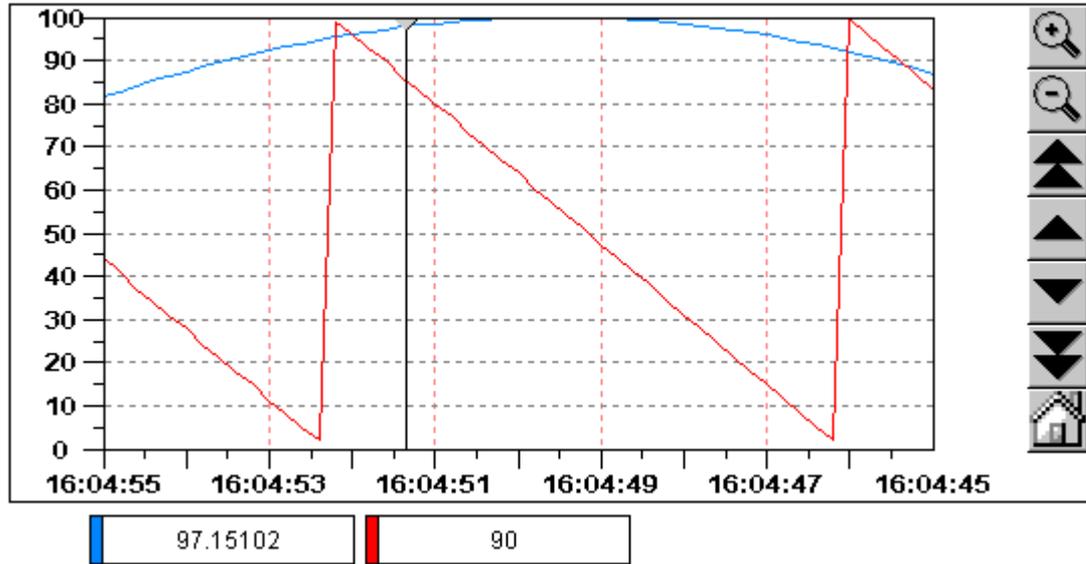
Configuration of two display fields for the current record values provided by the marker variables:

Rectangle element 1: Category Text: insert "%s" in the Content field; Category Variables: insert in field Textdisplay: PLC\_PRG.rSlider1

Rectangle element 2: Category Text: insert "%s" in the Content field; Category Variables: insert in field Textdisplay: PLC\_PRG.rSlider2

(Additionally insert a rectangle element at the left border of the rectangle elements 1 and 2, showing the curve color of the corresponding record variable)

Result in online mode after login and start of the program:



The record is running from the left to the right; the latest value is shown on the leftmost position; every 200 milliseconds the current value will be added to the display. The arrow buttons in the symbol bars allow shifting the displayed time range. Using the simple arrow buttons you can shift step by step, using the double arrows you get to the end resp. start of the record. For example: if you go to the start of the record by pressing the double arrows pointing to the left, you get a still display of the former values. If you then move the marker (grey tr upper left corner) along the time axis, you can read the exact values of each of the both recorded variables for each time in the rectangle elements below the diagram.

## Bitmap

You can do the following settings in category 'Bitmap' in the visualization element configuration dialog box:

Specify, which image file should be used. This specification can be static or dynamic. If there are entries both in the 'Bitmap' field and in the 'Bitmap Variable' field, then that in the 'Bitmap' field will be ignored:

**Bitmap:** Static definition: You can enter the path of an image file available in the local file system. Via button ... the standard dialog for browsing for a file will be opened, where you can select the desired file.

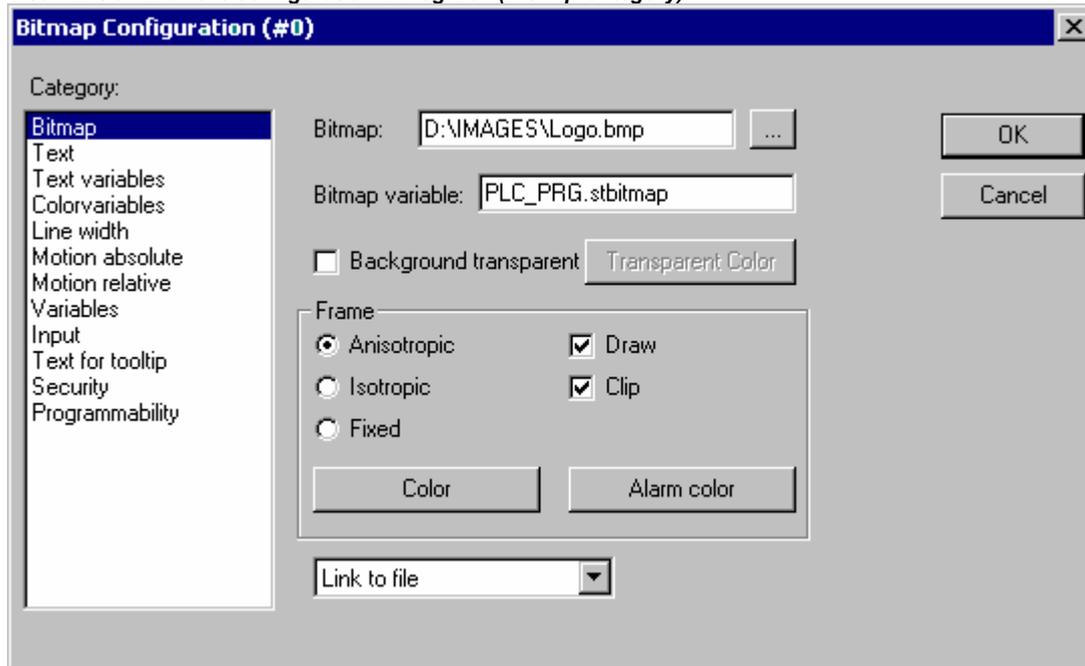
**Bitmap variable:** Dynamic definition: A project variable of type STRING can be entered here, containing the name of the image file currently to be used. This allows dynamic change of images in online mode, but only works with image files which are entered in the project-global **"bitmap list"** (see chapter 'Extras' 'Bitmap list'). The string-variable always only must get assigned the file name, even if there is a full path specified in the bitmap list.

See in the following example 'stBitmap' as string-variable, which is got assigned various image files that are part of the global bitmap list.

```
CASE nId OF
  0: stBitmap := 'background.bmp';
  1: stBitmap := 'deutest.bmp';
  2: stBitmap := 'alarm.bmp';
END_CASE
```

If stBitmap e.g. is declared in object PLC\_PRG, it can be entered here in the configuration dialog in field 'Bitmap variable' like follows: „PLC\_PRG.stBitmap“.

Visualization Element Configuration Dialog Box (Bitmap Category)



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.

The following 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 (default- or target-specific) 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 (**Embed**) or whether just a link to an external bitmap-file (path as entered above in the 'Bitmap' field) should be created (**Remember the Link**). 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.

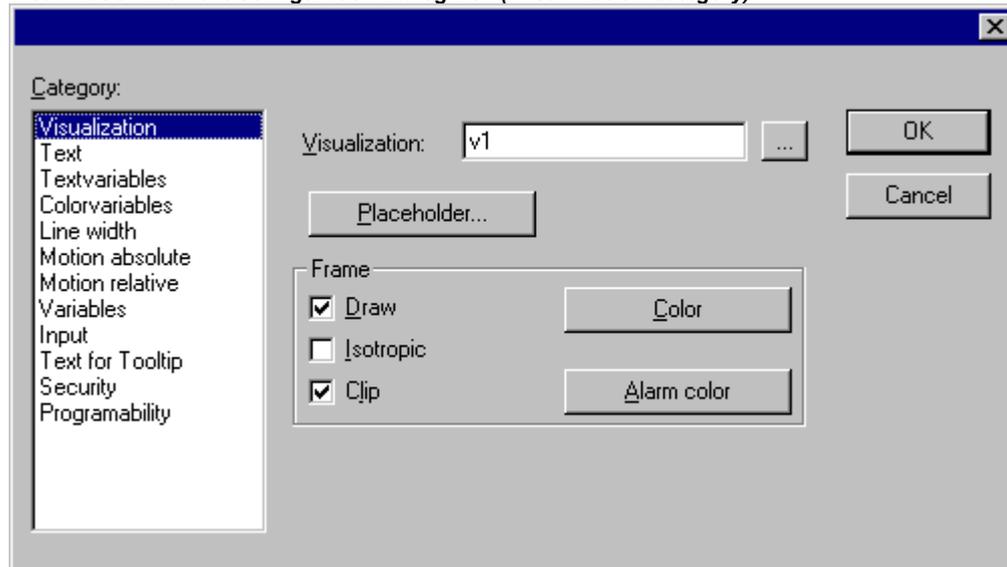
## Visualization

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

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

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 possible settings concerning the visualization **frame** are the same as described for a bitmap (see above).

**Visualization Element Configuration Dialog Box ( Visualization Category)**

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 instance 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 instances occurs directly when you define the call of 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..).

### Example of an application of the placeholder concept:

Instances of a function block can easily be displayed with the help of instances 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 an instance of visu is then used (by inserting visu in another visualization or by calling via 'Zoom to vis.'), then in the configuration of this instance 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 instances of 'visu').

Mark the first instance 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 instance 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 instance of 'visu'.

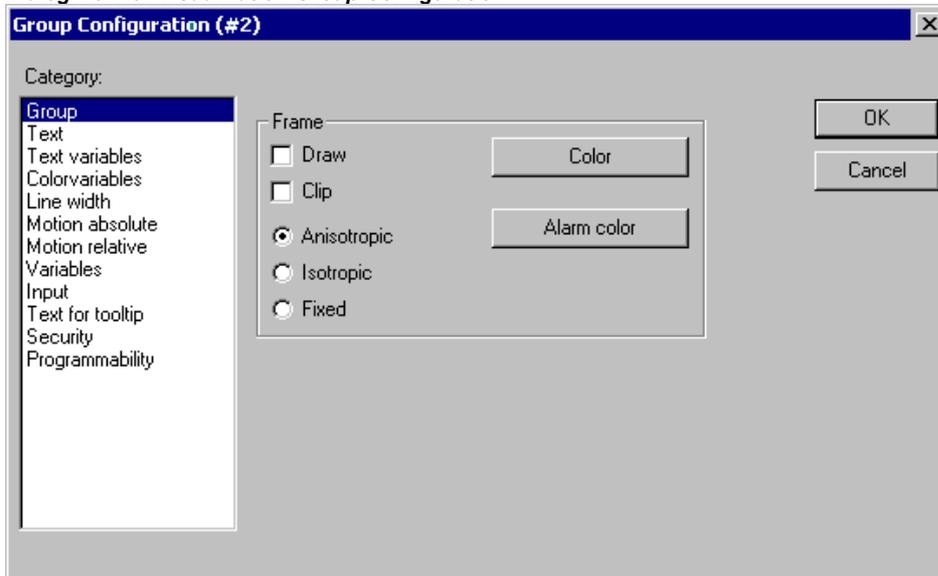
Of course the placeholder \$FUB\$ can be used at all places in the configuration of 'visu' where variables or text strings are entered.

**Attention: Online behaviour of a visualization instance:** If you insert a visualization and then select and configure this instance, 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 configure the instance, then in online mode its particular visualization elements will react exactly like those of the original visualization.

## Group

The configuration dialog for a group of visualization elements (see Chapter 2.3, Grouping elements) offers the same options **Anisotropic**, **Isotropic**, **Fixed**, **Draw**, **Clip**, **Color** and **Alarm color** in field 'Frame' as that for a Bitmap (see above). For example regard the possibility to keep the size of the single elements of the group even when the frame gets stretched or compressed.

*Dialog Box for Visualization Group Configuration*



## 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 being 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 being used in CoDeSys HMI. They are available in the configuration dialog for a visualization element:

Enter internal commands in the field **Execute program** in the category **Input** 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	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;
<Path executable program> [Path of the file to be opened] <sup>2</sup>	Program call	The program will be executed. Example: C:\programms\notepad.exe text.txt
LANGUAGEDIALOG <sup>2)</sup>	visualization settings	The dialog for visualization settings which includes the category language gets opened.
LANGUAGE <language identifier as used in the currently set language file *.xml, *.vis, *.tlt or *.txt>	visualization settings, Language	The desired language is set without using the dialog for visualization settings. See chapter 3 , Language switching in the visualization.
LANGUAGE DEFAULT	visualization settings, Language	For dynamic texts the default language will be used, which is defined in the currently included xml-file . See chapter 3 , Language switching in the visualization.
DELAY <delay time in milliseconds> <sup>1) 2)</sup>		The next command will not be executed before this time has elapsed. E.g. a delay of 500 ms is necessary between DEFINERECEIPT, READRECEIPT and SAVEWATCH.
DEFINERECEIPT <name of watch list>	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 of watch list>	'Read receipt'	In the defined watch list the pre-definition of the variables will be replaced by the current values. Regard: The watch list must be defined before via DEFINERECEIPT and a delay of 500 ms must be inserted ( see above: command DELAY) !
WRITERECEIPT <name of watch list>	'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 ! Important: call a previous DEFINERECEIPT to define the current receipt and insert a delay of 500 ms (see above: command DELAY) !
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 <sup>1) 2)</sup>	'File' 'Save'	The project will be saved.
EXITPROGRAM <sup>1) 2)</sup>	'File' 'Close'	The program will be exited.
PRINT <sup>1) 2)</sup>	'File' 'Print'	The current visualization will be printed out online.

Command	Equivalent in the programming version of CoDeSys	Explanation
HELP <name of help file> <sup>1)2)</sup>	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 <b>CoDeSys.ini</b> file. (see 'Extras' 'settings')
TRACE <sup>1)2)</sup>	Resources, Sampling Trace	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 <b>CoDeSys</b> are available in this window.
CAM <sup>1)2)</sup>	CAMs	If there is a CAMs definition available in the project (Resources), the CAM-Editor will be opened. As soon as the editor will be closed again, it will be returned to the visualization.
CNC <sup>1)2)</sup>	Resources, CNC program list	If there is a CNC program list available in the project (Resources), the CNC editor will be opened. As soon as the CNC editor will be closed again, it will be returned to the visualization.
1) not supported for Target-Visualization 2) not supported for Web- Visualization		

Only for usage in a Web- or Target-Visualization:

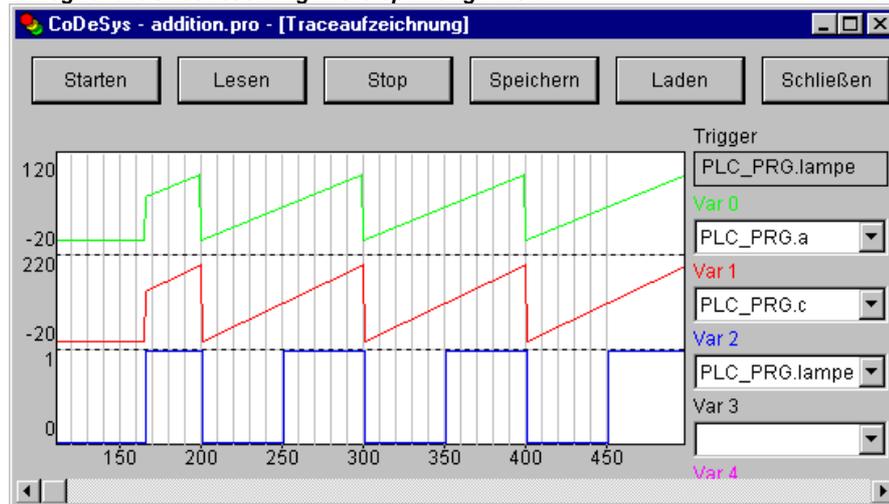
The following commands can be used to give the write access in online mode on a visualization to a certain client. This is of interest if multiple clients at the same time might modify data on the PLC. Please see [HERE](#) for further information.

GETWRITEACCESS	Request for write access
RELEASEWRITEACCESS	Deallocation of the write access which has been requested before
GLOBALRELEASEWRITEACCESS	Global deallocation of the write access

only for usage in a Web-Visualization:

INTERN LINK <URL>	The Web-Visualization will switch over within the browser to the defined URL (Unified resource location); e.g.."INTERN LINK http://www.3s-software.com"
INTERN LINK <HTTP file path>	The defined file will be opened; e.g. "INTERN LINK http://localhost:8080/test.pdf"
INTERN LINK mailto:<EMail-Adresse>	The entry mask for sending an EMail to the defined address will be opened; e.g. " INTERN LINK mailto:s.sdfjksk@companyxy.com"
INTERN CONNECT_TO <PLC-Name> <Start-Visu>	The target PLC will be changed; Precondition: The Web-Server is configured with the communication parameters for multiple PLCs and the appropriate ini-file is available for the PLC-Handler. PLC-Name: Name of the PLC as defined in the ini-file of the PLC-Handler. Start-Visu: Name of the desired start visualization page. The Web-Server will establish the connection to the respective PLC automatically. Example: "INTERN CONNECT_TO PLC1 PLC_VISU"

Dialog for the trace recording in the operating version



## 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.

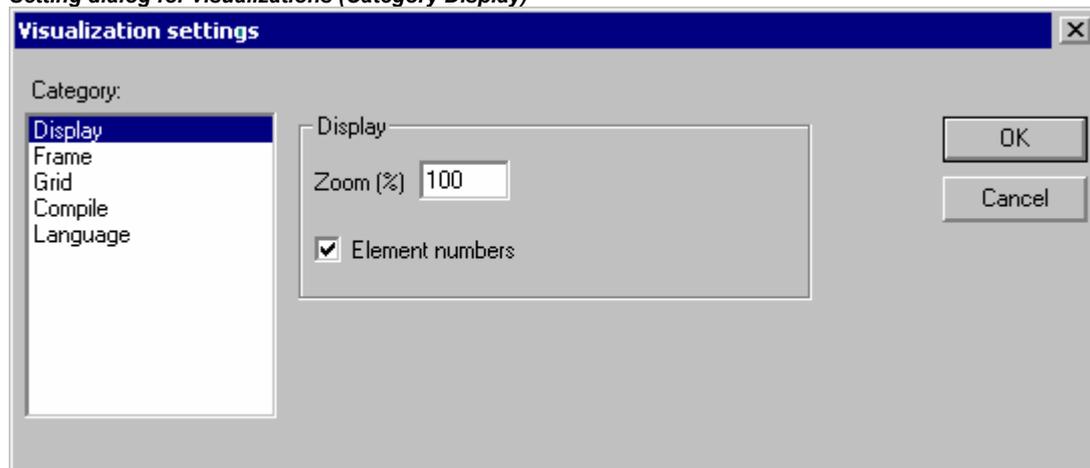
### 'Extras' 'Settings' / Display, Frame, Grid, Language

When this command is used, a dialog box will open in which you can make certain settings that affect the display and language of a visualization as well as the check of the visualization variables.

**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)



**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. If **Active** is set, the elements will 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 Compile:** Per default the used variables get not checked for validity before going online with a project. If you want this check to be done already during a build run of the project (command 'Project' 'Build' resp. 'Rebuild all'), then activate option **Check visualization variables on compile**. Invalid variables will be announced by a warning "... invalid watch expression..." in the message window.

**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. See chapter 3, Language switching in the visualization.

---

**Note:** The text display changes only in Online mode!

---

### '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'

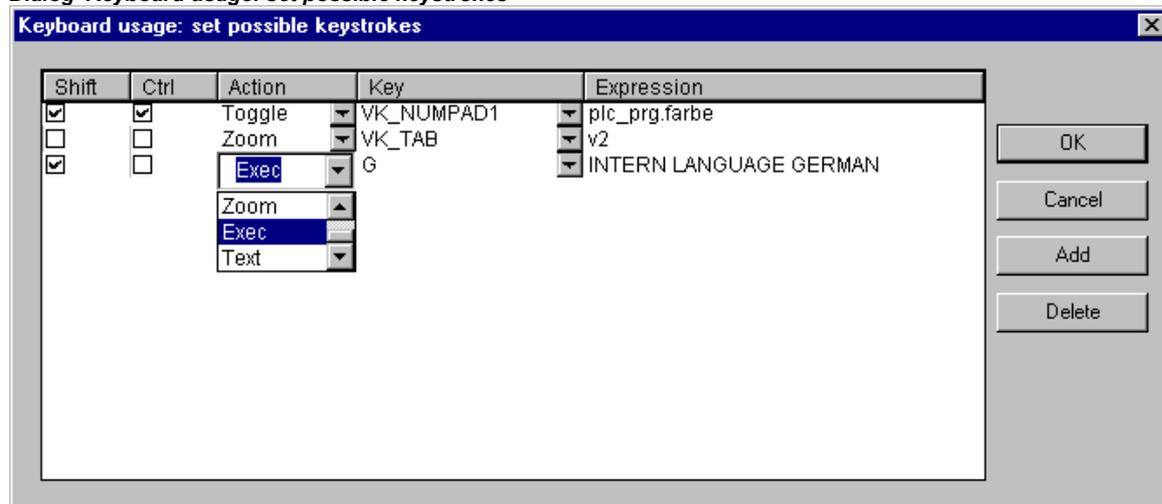
The use of hotkeys can optimize the pure keyboard operation of a visualization.

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 mouse-click 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'*



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	End-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. See Appendix C for which combinations are supported by the particular visualization variants.

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 or 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 visualization will switch to German. <Ctrl><Z> here will cause a jump to visualization PLC\_VISU.

## Master layout

The usage of a "Master layout" in visualizations for example could be used to provide a dialog in various visualizations without the need to explicitly insert it in each of them. Whether and when the dialog will be displayed in online mode could be controlled via a variable which is defined in the master layout configuration for the visibility of the dialog elements.

### Use as master layout

If a visualization is defined as "Master layout", it will be inserted automatically in all other visualizations (if those are not explicitly excluded, see below) and will be available there with all its functions during online mode. It always will be inserted on the front level, if you however it always want to get it in the background, then activate option 'in background' in the properties dialog of the visualization element (see below). A master layout cannot be edited any longer in the visualization where it has been inserted. Modifications of the configuration only can be done in the master layout visualization itself.

Appointing a visualization to be the Master layout is done in the **Properties dialog**, which can be opened for an object currently selected in the Object Organizer via command 'Project' 'Object' 'Properties' (see CoDeSys User Manual, Chapter 4.3). Option **Master layout** must be activated for this purpose. If prior to this another visualization has been defined to be a master layout, that one will automatically be re-defined to be a "normal" visualization (option **Visualization** in the Properties dialog).

### Visualization without master layout

Also in the Properties dialog of a visualization object you can define that it should be used as a **Visualization without master layout**.

## Use as Web- or Target-Visualization

If a project is created for a Web- or Target-Visualization (see Chapter 8 resp. 9), for each visualization object you can define whether it should be used for this purpose or not.

There to select the visualization object in the Object Organizer and open the **Properties dialog** ('Project' Object' 'Properties', see CoDeSys User Manual, Chapter 4.3). If in the Target Settings the options Web- resp. Target-Visualization are activated, in the Properties dialog the corresponding options **Use as Web-Visualization** resp. **as Target-Visualization** automatically also will be activated. In order to explicitly exclude the object from the use in a Web- or Target-Visualization deactivate the appropriate option.

### 2.4.3 Image files in the visualization

Image files can be used in a visualization object for the background as well as in visualization elements of type "Bitmap". The following formats are supported:

- CoDeSys (HMI): \*.bitmap, \*.jpg, \*.tif
- Web-Visualisierung : \*.bmp, \*.jpg
- Target-Visualisierung: \*.bmp; ob \*.jpg und \*.tif unterstützt werden, ist zielsystemabhängig.

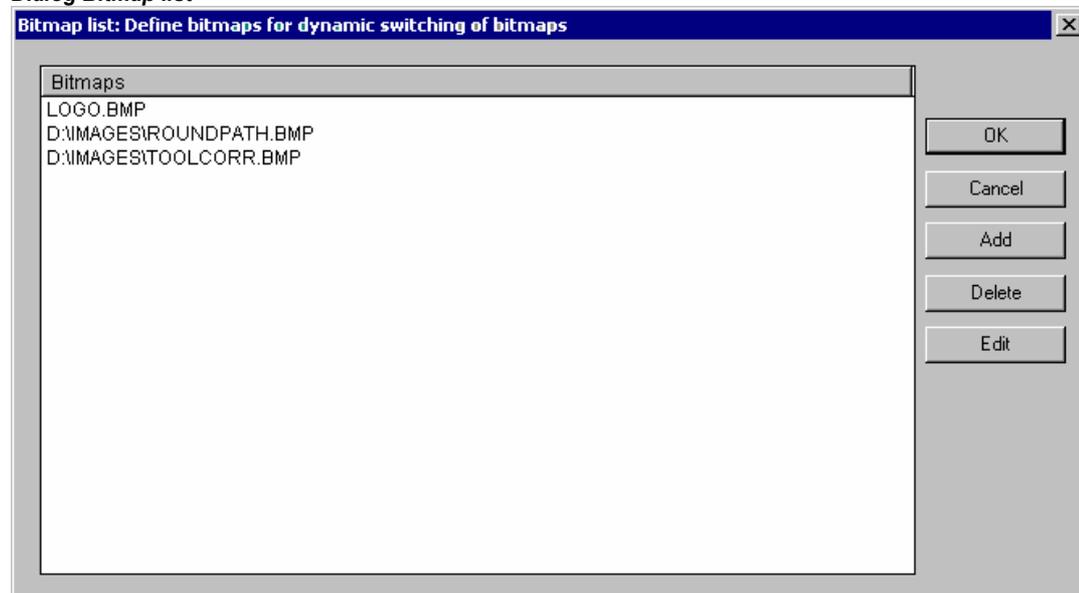
In „Bitmap“ elements a dynamic change of images can be reached by specifying the image file name by a project variable instead of using a definite file reference. See for further information chapter 2.4.1, configuration category 'bitmap' and the following chapter 'Extras' 'Bitmap list'.

The background image of a visualization can only be defined by a static file reference; see chapter 2.4.2, 'Extras' 'Select Background Bitmap'.

#### 'Extras' 'Bitmap list'

In the 'Extras' menu which is available when the Visualization editor is active, command 'Bitmap list' opens dialog 'Bitmap list:...'. Here you can enter image files, which then can be used in all visualization objects via a project variable defining the content of a "Bitmap" element. This allows a dynamic change of images in online mode - alternatively to the static use of image files, which for this purpose are not to be part of the bitmap list.

##### *Dialog Bitmap list*



In order to enter an image file in the bitmap list, use button Add. The standard dialog for browsing for a file will be opened, where you can select an image file (\*.bmp, \*.tif, \*.jpg, regard the supported file formats, see introductory paragraph of chapter 2.4.3!). If the chosen file is in the project directory, only the file name will be added to the list, otherwise the full path.

Via button Delete the currently selected entry (click by the right mouse-button) can be deleted.

In order to modify an entry, either edit it directly (click by the left mouse-button on the entry opens the edit frame), or use button Edit to open the standard dialog for browsing for a file. For the latter the entry first must be selected by a click with the right mouse-button.

OK saves the current list.

Each file name specified in the list now can be assigned to a variable of type STRING, which is entered in the bitmap configuration of a visualization element in order to define the image to be used. Concerning this please see: Chapter 2.4.1, Configuration of visualization elements, category „Bitmap“.

## 3 Language switching in the Visualization

---

The language switching for texts, tooltips and alarm messages in a visualization can be done via static or via dynamic texts, which must be provided by a file.

How can a language switching be done:

In the configuration dialog 'Visualization settings' (see chapter 2.4.2, 'Extras' 'Settings', category 'Language') in the selection list below Language you can choose one of the languages defined in the currently used language file, which should be used as start language in online mode; for the example shown below: german and english.

A language switch in online mode is done via an input on a visualization element. For this purpose the internal commands "INTERN LANGUAGE <language> and "INTERN LANGUAGE DIALOG" are available (see chapter 2.4.1, "Special input possibilities for operating versions"), which can be used in the configuration dialog in category 'Input'.

For example you can insert a button element which can be used to switch the visualization texts to German. For this purpose label the element with 'German', in configuration category 'Input' activate option 'Execute program' and define a command "INTERN LANGUAGE <language>". "language" is to be replaced by the language shortcut used in the language file, thus for the vis-file example shown in chapter 3.1: "INTERN LANGUAGE german". If the button will be operated in online mode the visualization texts will be displayed according to the entries which are available for "german" in the language file.

### Unicode Format:

Unicode format is only possible with dynamic texts. An appropriate entry must be available in the XML language file. Additionally the following preconditions must be complied:

1. In the configuration of the visualization object a unicode-enabled font must be specified. Attention: Currently for tooltip texts this font must be specified in the codesys.ini file ("FaceTooltip=<font>") and is not read from the XML language file!
2. In order to get work the unicode font in the Target-Visualization, it must be supported by the target system.
3. In order to get work the unicode font in the Web-Visualization, the configuration entry TOOLTIPFONT in webvisu.htm must be set with a unicode-enabled font.

### 3.1 Static language switching

---

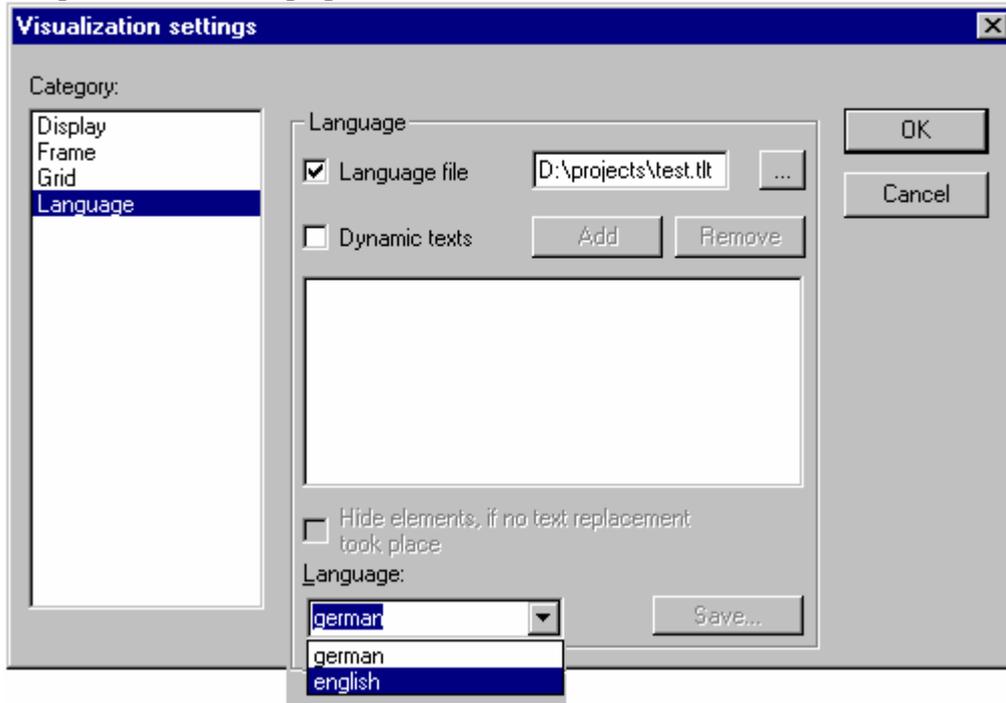
For a static switch of the language a **language file** (\*.vis, \*.tlt, \*.txt) file can be used (for how to create see below). The difference to the dynamic language switching is that the language cannot be defined by a project variable during run time.

**Attention:** For visualizations it is recommended to use the \*.vis language file, because \*.tlt- resp. \*.txt-translation files only work for visualizations in CoDeSys resp. CoDeSys HMI and also in those not for the Meter, Bar Display and Histogram elements.

In dialog 'Visualization settings' (see chapter 2.4.2) you configure, which language file should be used with the project. In order to choose a translation (\*.tlt, \*.txt) or a pure visualization language file (\*.vis), which contains the texts in the various languages, activate the **Language file** option and in the input

field next to it enter the appropriate file path. Via the  button you get the standard dialog for opening a file.

Dialog for selection of a language file for a visualization



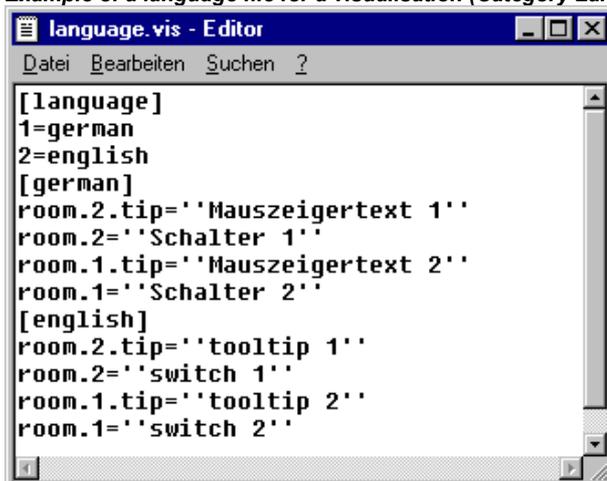
Regarding creating a **translation file \*.tlt or \*.txt**, please see 'Project' 'Translate into other languages' (User Manual for CoDeSys).

For creating a special **language file \*.vis** see the following steps:

Open likewise the Visualization settings 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)



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.

## 3.2 Dynamic Language switching

---

Dynamic texts allow switching between different, always language-assigned text versions for a visualization element (texts, texts for tooltips, message texts in alarm tables). The difference to static texts is that the definite text selection also can be done via a variable used in the application.

In the configuration of the element a **Prefix-ID-combination** is entered, which is assigned to a text in a **XML-file** (also named "textlist" in the following). The **ID can be defined by a project variable** (see chapter 3.2.2).

Example of application: The ID represents an error number, as Prefix e.g. "Error" is used. The language file provides via the Prefix-ID-combination an appropriate error message, which – depending on the currently set language – will be displayed in this language (see chapter 3.2.2).

### Please regard:

- The language files for dynamic texts can be created in Unicode (UTF-16) or ANSI (ISO-8859-1), e.g. " <?xml version='1.0' encoding='UTF-16'?>". For the preconditions for the usage of unicode-able fonts please see [HERE](#).
- For the Target-Visualization the start language, the directory for the xml-files to be used and a list of xml-files can be defined by the target system. This allows to modify these parameters later without the need of creating a new boot project. Thus in an easy way existing textlists can be modified (start language, texts) resp. new languages can be added. If the target system is providing such a configuration, the textlists which are defined for the visualization in CoDeSys, will not be regarded in online mode! If no target-specific configuration is available for the language switching, then after a modification of the textlists defined in CoDeSys a project download must be done.

### 3.2.1 Configuration

---

You can control dynamically which text (also alarm message text in an alarm table) will be displayed in a visualization element resp. as tooltip of an element in online mode by using prefix-ID-combinations, each pointing to another text defined in a XML-file (see chapter 3.2.2).

For this purpose a XML-file describing the text assignments must be linked to the project in the **configuration of the visualization** (see chapter 2.4.2, 'Extras' 'Settings'). The XML-file must have a certain format. A language code is added to the particular texts, thus later you not only can switch between different text contents but also language switching is supported.

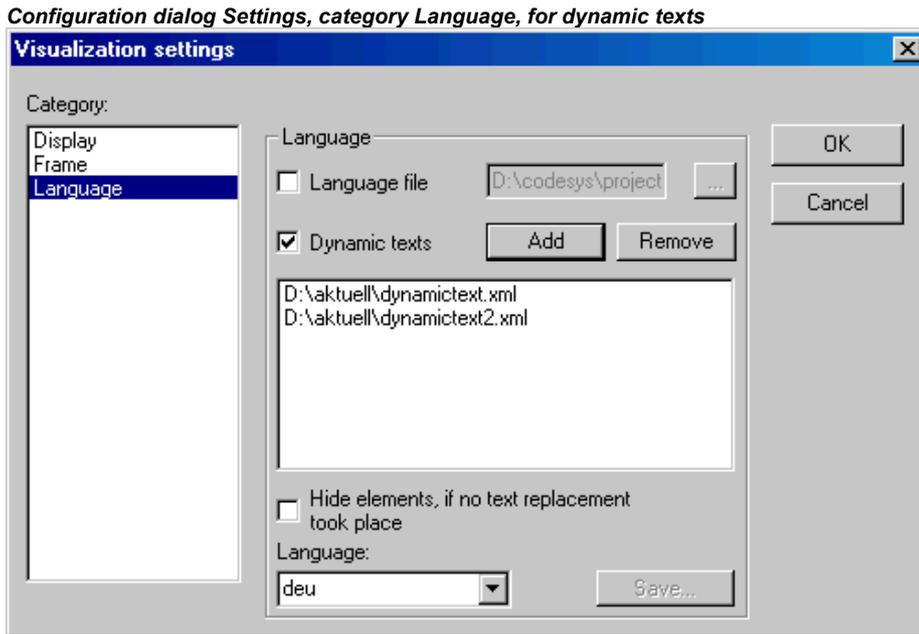
In the **configuration of a visualization element**, for which the text display should be switched dynamically, Prefix and ID are entered (see below) whereby the ID can be provided by a project variable. A default language can be defined via an INTERN command (see chapter 2.4.2, Special input possibilities for operating versions)

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. Link the XML-file(s) and choose the (start) language: Dialog 'Settings' category Language:

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 those text versions (for the corresponding prefix-ID-combination) which are marked with that language identifier in the XML.-file.



2. Specify the ID in configuration dialog 'Variables' in field 'Textdisplay' resp. 'Tooltip-display':

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

In case of message texts of an alarm table, the ID must match the respective line number in the table.

3. Define the text format in configuration dialog 'Text' resp. 'Text for Tooltip':

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 string matching with a PREFIX-definition used in the XML-textlist. See the description for the 'Text' configuration dialog.

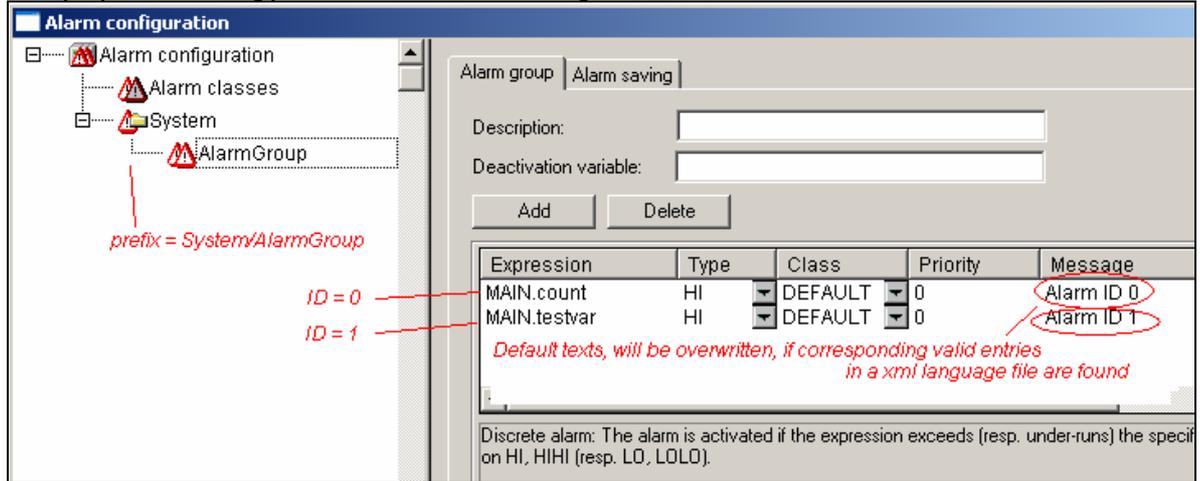
For each prefix-ID-combination, which is found in a linked XML-file, the assigned text will be displayed in the visualization element in online mode. If no appropriate entry is found, no replacement will be done.

### 3.2.2 XML-File for Dynamic Texts

For a description how to use dynamic texts in the visualization see chapters 3.2 (Language switching) and 3.2.1 (Configuration).

The underlying 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 an ID). In case of normal texts and tooltip texts these prefix-ID combinations can be entered in the configuration of a visualization element (see above Chapter 3.2.1). In case of an alarm table the prefix and ID are given by the "path" of the alarm group in the alarm configuration (prefix) and the line number of the text definition in the alarm group configuration.

Example picture showing prefix and ID in an alarm configuration



In the header section of the file a default language and a default font assigned to a language can be defined.

The descriptions in the xml-file are enclosed by tag <dynamic-text> and <\dynamic\_text> which have to be entered at the beginning resp. end of the file.

The language files for dynamic texts can be created in **Unicode** (UTF-16) or **ANSI** (ISO-8859-1). This is to be defined via the encoding syntax at the beginning of the xml-file, e.g. "<?xml version='1.0' encoding='UTF-16'?>". For the preconditions for the usage of unicode-able fonts please see the introductory of chapter 3.

See a file example at the end of the chapter.

**Please note:**

- **Primary formats** of the xml-file, which do not use the <dynamic\_text>\<\dynamic\_text> tags or the header section, will be supported further on!
- The **Target-Visualization** offers an interface for scanning the entries of dynamic textlists. Thus those can be used directly in the program.

File structure:

The **header section** starts with <header> and is closed with </header>. If you want to define a default language, use entry <default-language>. A default font which is assigned to a certain language, can be defined via entry <default-font>. These entries are optional. If they are missing, the dynamic text in the visualization will be displayed according to the local configuration settings of the visualization.

<pre>&lt;header&gt;</pre>	
<pre>&lt;default-language&gt; &lt;language&gt; &lt;/default-language&gt;</pre>	<p>Default language; that means that if there is no text entry available for the currently set language, that text will be used which is found within the same text entry for the default language. If also for the default language no text is found, "&lt;PREFIX&gt; &lt; ID&gt;" will be displayed.</p> <p>If multiple XML-files are used, thus providing multiple headers, only that header section will be regarded, which is read at last. So it is reasonable to use only one header section! The language token must correspond to one of those used in the text entries (see below).</p> <p>Note: In online mode the default language can be set explicitly via a visualization element configured with command INTERN LANGUAGE DEFAULT in category Input, Execute program (see chapter 2.4.1, Special input possibilities for operating versions).</p>

<pre> &lt;default-font&gt;    &lt;language&gt;&lt;language&gt;&lt;/language&gt;    &lt;font-name&gt;&lt;font&gt;&lt;/font-name&gt; &lt;/default-font&gt; &lt;default-font&gt;   &lt;language&gt;.....   .... &lt;/default-font&gt; &lt;/header&gt; </pre>	<p>Default font for &lt;language&gt;: The given font (e.g. "Arial" will automatically be used for all elements, which display dynamic texts in &lt;language&gt;. The language token must correspond to one of those used in the text entries (see below).</p> <p>For the preconditions for the usage of unicode-able fonts please see the introductory of chapter 3.</p> <p>further default fonts for other languages</p>
---	---

The **list of assignments** Prefix-ID-text starts with **<text list>** and ends with **</text list>**.

The particular text entries each start with **<text prefix>** and end with **<text>**. A text entry which is assigned to a Prefix-ID-combination must contain the following lines:

<pre> &lt;text prefix&gt;= "&lt;PREFIX&gt; id="&lt;ID&gt;" </pre>	<p>"PREFIX" corresponds to the &lt;PREFIX&gt; used in the visualization element configuration (category 'Text' or 'Tooltip'); In case of a message text to be displayed in an alarm table the path of the respective alarm group in the alarm configuration as from the "System" node must be specified, e.g. "System/Alarmgroup_1".</p> <p>"ID" for normal texts or tooltip texts corresponds to the entry in category 'Variables', Textdisplay or Tooltipdisplay; in case of an of an message text used in an alarm table the number of the line which defines that text in the configuration dialog of the alarm group must be specified.</p> <p>(See the CoDeSYS V2.3 User Manual for CoDeSys for a description on the Alarm Configuration)</p>
<pre> &lt;language&gt; &lt;CDATA[&lt;TEXT&gt;]] &lt;/language&gt; </pre>	<p>Use any string as a 'language' identifier (e.g. "english"). This identifier then 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.</p>
<pre> &lt;/text&gt; </pre>	

For each prefix-ID-combination at least for **one** language a text entry must be available. E.g. see in the file example shown below: <deutsch> indicates the start of the german version of a text, </deutsch> terminates the text.

Dynamic 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 for the use of an language xml-file:** You want to have two visualization elements, one for visualizing the current machine identification, the other for visualizing an error message according to a currently given error number:

- (1) Define in PLC\_PRG the following variables: *ivar* of type INT, defining the current machine identification; *errnum* of type INT defining the current error number.

- (2) Configure a visualization element for displaying the current machine identification:
  - a. Enter in category ‚Text‘ in the text field: "%<Maschine>"
  - b. Enter in category ‚Variables‘ at Textdisplay: „PLC\_PRG.ivar“
- (3) Configure another visualization element for displaying the error message for the currently occurred error:
  - a. Enter in category ‚Text‘ in the text field: "%<Error>"
  - b. Enter in category ‚Variables‘ at Textdisplay: "PLC\_PRG.errnum“
- (4) Create a xml-file, e.g. with name *dynamictexts.sample.xml*, according to the syntax described above, which should look as follows for the current example:

```

<dynamic-text>
<header>
  <default-language>deutsch</default-language>
  <default-font>
    <language>deutsch</language>
    <font-name> Arial </font-name>
    <font-color>0,0,0</font-color>
    <font-height>-13</font-height>
    <font-weight>700</font-weight>
    <font-italic>>false</font-italic>
    <font-underline>>false</font-underline>
    <font-strike-out>>false</font-strike-out>
    <font-char-set>0</font-char-set>
  </default-font>
  <default-font>
    <language>english</language>
    <font-name> Arial </font-name>
    <font-color>0,0,0</font-color>
    <font-height>-13</font-height>
    <font-weight>700</font-weight>
    <font-italic>>false</font-italic>
    <font-underline>>false</font-underline>
    <font-strike-out>>false</font-strike-out>
    <font-char-set>0</font-char-set>
  </default-font>
</header>
<text-list>
  <text prefix="ERROR" id="4711">
    <deutsch> Fehler an Position 4711 </deutsch>
    <english> Error at position 4711 </english>
  </text>
  <text prefix="ERROR" id="815">
    <deutsch> Fehler an Position 815 </deutsch>
    <english> Error at position 815 </english>
  </text>
  <text prefix="ERROR" id="2000">
    <deutsch> <![CDATA[Das ist ein Fehlertext über
      mehrere Zeilen]]> </deutsch>
    <english> <![CDATA[This is a error text over more than
      one line]]> </english>
  </text>
  <text prefix="MASCHINE" id="1">
    <deutsch> <![CDATA[Vorschub]]> </deutsch>
    <english> <![CDATA[Feed rate]]> </english>
  </text>
  <text prefix="MASCHINE" id="2">
    <deutsch> <![CDATA[Beschleunigung]]> </deutsch>
    <english> <![CDATA[Acceleration]]> </english>
  </text>
</text-list>
</dynamic-text>

```

- (5) In the visualization open dialog ‚Settings‘, category Language: Activate option ‚Dynamic Texts‘; Add file *dynamictexts.sample.xml*, now available on your computer, to the file list.

- (6) Go online with the project.
- (7) In the visualization settings set language to "deutsch". Set PLC\_PRG.ivar to "1" and PLC\_PRG.errnum to "4711". Now in the visualization elements the following texts should be displayed: "Vorschub" resp. "Fehler an Position 4711". The texts will be displayed in Arial 13.
- (8) Set PLC\_PRG.ivar to „2“ and PLC\_PRG.errnum to "2000". The texts will change to "Beschleunigung" and "Das ist ein Fehlertext über mehrere Zeilen“.
- (9) In the visualization settings change the language to "english". Now the following texts will be displayed: "Acceleration" and "This is an error text over more than one line".

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

### **3.3 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\<hilfedatei_german>.chm
English=C:\PROGRAMME\HELP\<hilfedatei_english>.chm
```

## 4 Placeholder Concept

At each location in the configuration dialog where 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 2.4.1, „Configuring an inserted visualization“, there you also find an example for using 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 those replacements.

### 'Extras' 'List of Placeholders...'

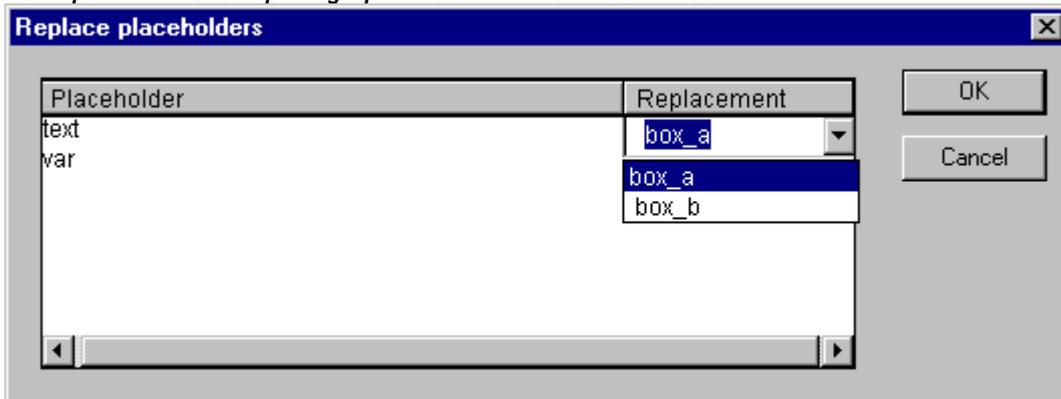
This list is used at two places in CoDeSys: to manage placeholders and to configure them:

1. 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 additionally 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:
2. 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 instance.

*Placeholder list for input of possible replacements for the placeholders*



3. 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 an 'instance') 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.

## 5 Visualization in Online Mode...

### 5.1 Overview

---

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 (Programmability), 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.
- Before a download of the project regard the current setting of option 'Prevent download of visualization files' in the target settings. This concerns all files which are used in the current visualization will not be downloaded to the target system. Visualization files are only downloaded for Target- or Web-Visualization and can be bitmaps, language files and for Web-Visualization also XML description files.
- The **configuration** settings for Display, Frame and Language can also be edited in online mode.
- As long as a visualization "**instance**" is not configured explicitly, the particular elements of the instance in online mode will react on inputs like those of the original visualization ("mother" of the instances).
- When you switch the **language** ('Extras' 'Settings') this will only effect the display in online mode.
- A visualization can be **printed** in online mode.
- If a visualization is used as Target-Visualization, information on user entries via mouse-clicks can be scanned with the help of special interface functions and thus be used in the project.

### 5.2 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. Within tables you will get to the next field. Pressing the key while keeping the **<Shift>** key depressed selects the previous element. Depending on the target a **simplified input handling** may be possible (see CoDeSys User Manual, Target Settings for Visualization).
- 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 or if it is a table field, a text input field will be opened which displays the text contents of the variable resp. the field. 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 other functions than the above described standards.

The individual elements of instances behave in Online mode identically to the corresponding ones in the visualization that is instanced. They will therefore react the same way as individual elements to inputs and operation by mouse and keyboard; the display of tooltips in instances 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 instance proceeds from the location of the instance in the element list before jumping to the next element of the list.

<b>Note:</b>	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! In the Web-Visualization a specific setting in webvisu.htm (see chapter 8.3) allows to keep effective the operation over the keyboard even if currently an entry field is opened.
--------------	--

### 5.3 '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.

### 5.4 Access Protection in multi-client operation

Due to the fact that it is possible to connect multiple visualization clients to one PLC (Target- and Web-Visualization), multiple clients might access data in the PLC at the same time. If this is not desired, an access protection can be defined by certain INTERN commands (see also chapter 2.4.1), so that always only one client can write the data on the PLC.

The following INTERN commands are available for restricting the write access to one client:

GETWRITEACCESS	<p>The client requests write access on the visualization.</p> <p>Using the system variable CurrentWriteAccessClientId the client then can check whether it has got write access. This implicit variable stores the identification (ID) of the client which currently has write access. Each client has a unique ID "CurrentClientId".</p> <p>Thus the following expression can be used to allow a certain input on a visualization:</p> <p>CurrentWriteAccessClientId = CurrentClientId</p> <p>(Example: The expression could be used in the configuration of an element to define its visibility : If the expression is TRUE, i.e. the client has write access, then the element is visible, i.e. allows input)</p> <p>The Web-Server will check at each write request, whether the requesting client currently has write access.</p>
RELEASEWRITEACCESS	The client deallocates its write access.
GLOBALRELEASEWRITEACCESS	The write access is deallocated client-independently.

Automatische Rückgabe des Schreibrechts:

Wenn während des Monitorings festgestellt wird, dass der aktuell schreibberechtigte Client nicht mehr mit der Steuerung verbunden ist, wird die Variable CurrentWriteAccessClientId auf "-1" gesetzt, was bedeutet, dass kein Client mehr Schreibrecht besitzt. Dies kann auch über die IEC-Applikation erfolgen.

## 6 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 instances 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 cause errors 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 instances or visualization calls in the program first the visualizations in the project, and only thereafter the ones in the loaded libraries will be implemented.



## 7 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 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.

### 7.1 Installation, Start and Operating

CoDeSys HMI can be installed with the standard CoDeSys setup. If no valid license is available, a time-limited demo version can be used.

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 existent 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:

<b>"/simulation" resp. "/target"</b>	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 explicitly whether the project should run in simulation mode or on the target.
<b>/visu &lt;visualization POU&gt;</b>	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>".
<b>/visudownload</b>	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".
<b>/visucompactload</b>	This parameter can be used for optimization at starting a project for which no download is required. If a download gets necessary anyway, e.g. effected by parameter "/visudownload", "/visucompactload" will be ignored.

**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.

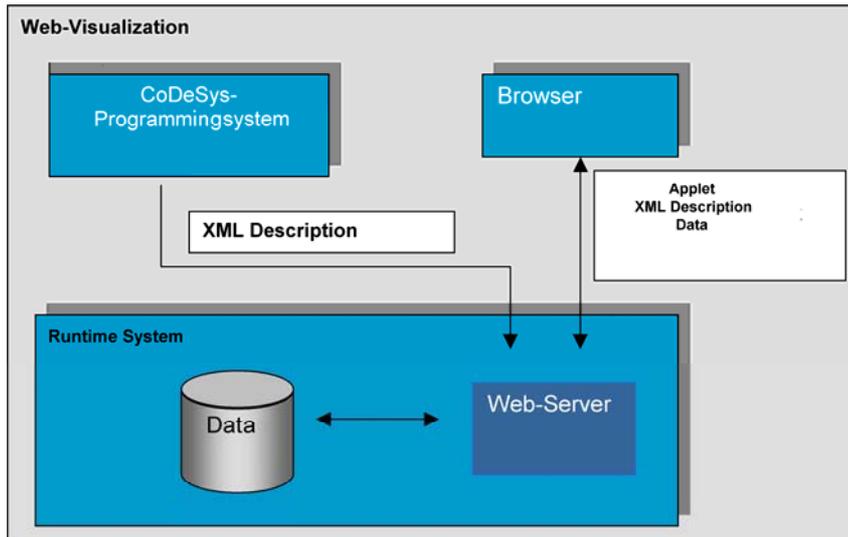
If there is no visualization element configured with a corresponding function, CoDeSys HMI at any time can be terminated by pressing **<Alt><F4>**.

## 8 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 continuously 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).

The Web-Server can connect dynamically to several controllers if configured appropriately. For this purpose the PLCHandler will be used as a basic component of the visualization. Visualization elements can be configured in a way that they will effect a connection change to another target system (see chapter 8.7).



### 8.1 Preconditions

In order to make visualization created in CoDeSys available as Web-Visualization, the following preconditions must be fulfilled:

- 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 correctly configured **WebServer** must be started. (see below)
- In order to get the Web-Visualization displayed on a computer, a **Web-Browser** (IE-Explorer or Netscape) is required.
- **Operating system:** Windows NT/2000, Windows CE, Linux, RTE

**Note:** Before a download regard the current setting of option 'Prevent download of visualization files' in the target settings. This concerns all files which are used in the current visualization. Those can be bitmaps, language files and for Web-Visualization also XML description files.

### 8.2 Status check, Auto-Reload, File error\_ini.xml

If a configuration file error\_ini.xml is provided by the manufacturer for the target system, status changes of the PLC resp. the visualization project are notified as messages. Usually in this case communication errors, start/stop of the PLC, breakpoint states, new download etc. will be reported in message boxes. The ini-file however also might define error messages like "page cannot be

displayed", "too many variables to be monitored" or the like. Also as a reaction on a download or online change of the project an automatic reload of the Web-Visualization might be defined.

The time interval for a cyclic status check can be defined in the file WebVisu.htm (PLCSTATEINTERVAL).

<b>Note:</b>	A communication failure during the monitoring of the Web-Visualization will not be dumped as an error message, but will be displayed in the status line of the webbrowser.
--------------	--

### 8.3 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. This is the basic html-page for the Web-Visualization. Before it gets downloaded to the target system together with the Web-Visualization project, it can be edited in a text editor in order to change the behaviour of the Web-Visualization:

Webvisu.htm per default contains the following:

```
<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:

- Width, height - Definition of the size of the screen. Regard the possibility to make visible this size already during creating a visualization in CoDeSys (Target Settings: Display width/height in pixel).
- 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' (see chapter 2.1) is changed by the PLC program. (For information on implicit variables see 10.)

Optionally the file can be extended by the following entries in the APPLET definition part:

```
<param name="USEFIXSOCKETCONNECTION" value="FALSE|TRUE">
```

If this parameter is TRUE, a fix socket connection will be used for monitoring; if it is FALSE or if the entry is missing at all, for each monitoring request a new socket will be used. Default: FALSE. Attention: If parameter USEURLCONNECTION (see below) is used, USEFIXSOCKETCONNECTION may not be set TRUE.

```
<param name="FORCEDLOAD" value="Comma-delimited visualizations list">
```

The visualizations specified here will be loaded already when the Web-Visualization is loaded, not just when they are opened for the first time. Thus time is saved at later changes of visualizations, because then the data not have to be transferred first by the WebServer.

Examples:

```
<param name="FORCEDLOAD" value="VISU_1, VISU_2, VISU_3 ">
<param name="FORCEDLOAD" value="TREND">
<param name="COMPRESSEDFILES" value="FALSE|TRUE">
```

The files to be transferred for the Web-Visualization to the Web-Server can be provided by CoDeSys in a packed format ("`<filename>_<extension original format>.zip`"). See for details: CoDeSys User Manual CoDeSys\_V23\_E.pdf, Target Settings / Visualization.

If parameter "COMPRESSEDFILES" is set TRUE, the Web-Visualization first will try to load from the Web-Server all needed files which have got appended the extension „.zip“and to unzip those. If the request for a zip-file fails, it will be tried to load the original file.

If parameter "COMPRESSEDFILES" is not available or is configured with value FALSE, any available zip-files will be ignored.

```
<param name="USEURLCONNECTION" value="URL">
```

If this parameter is configured, the communication will be done via the specified URL-connection. Per default a simple socket connection is used.

Attention: If parameter USEFIXSOCKETCONNECTION (see above) is set TRUE, USEFIXSOCKETCONNECTION may not be used additionally.

Example:

```
<param name="USEURLCONNECTION" value=http://192.168.100.19:8080/webvisu.htm
```

```
<param name="SELECTION" value="Line width|RED|GREEN|BLUE">>
```

Here the line width and color for the display of the current selection can be defined. Syntax: LINEWIDTH|RED|GREEN|BLUE; Example: „4|0|0|255“

```
<param name="ERROR_SENSITIVITY" value="Number of trials to get a file transferred">>
```

This parameter defines how many trials will be done to get a visualization file transferred from the Web-Server, before an applet error will appear.

```
<param name="KEYPADINDIALOGS" value="FALSE or TRUE">>
```

If a touch panel is used for working with the Web-Visualization, this parameter should be set TRUE in order to get an input possibility in any case for each dialog; if applicable via numpad/keypad.

```
<param name="KEYBOARDUSAGEFROMDIALOGS" value="FALSE or TRUE">>
```

If this parameter is set TRUE, the keyboard usage is always active, even if a modal dialog - like e.g. the numpad - is currently opened.

```
<param name="WRITEACCESSLOCK" value="FALSE or TRUE">>
```

This parameter only should be set TRUE, if the Web-Server supports multi-client processing and if an access lock for various clients is desired. Concerning access control in multi-client operation please see [HERE](#).

```
<param name="DEFAULTENCODING" value="FALSE or TRUE">>
```

If this parameter is set TRUE and the language switching is done via ASCII language files, the default encoding - currently set in the system - will be used for the interpretation of the language file.

```
<param name="ENCODINGSTRING" value="encoding string">>
```

If the default encoding of the system is not set as desired, you can define here the desired encoding by entering the appropriate string. Examples for encoding strings:

German: "ISO-8859-1"

Russian: "ISO-8859-5"

Japanese: "MS932"

```
<param name="PLCSTATEINTERVAL" value="Zykluszeit">>
```

Cycle time in milliseconds according to which the Web-Client will check the PLC status. It will be checked whether the PLC is in Start or Stop status and whether a download has been done.

```
<param name="ALARMUPDATEBLOCKSIZE" value="Number of alarm states to be updated per cycle">>
```

This parameter can be set in order to change the update of the alarm states. Due to the fact that not all alarm states can be updated within one cycle, it might be useful to exactly defined the

number of alarms which should be updated per cycle. This number can be specified as a numeric value.

```
<param name="SUPPORTTOOLTIPSINALARMTABLE" value="TRUE oder FALSE">>
```

If this parameter is set TRUE, the tooltip functionality in the alarm table will be activated. This means: If any text entry in the alarm table cannot be displayed completely, a tooltip will be available showing the full text string as soon as the mouse pointer is moved on the respective table cell.

```
<param name="TOOLTIPFONT" value="Schrifttyp|Schriftgröße">>
```

This parameter serves to define the font for all tooltips. Syntax: Font|Size; Example: „Arial|11“.

```
<param name="FILEOPENSAVEDIALOGFONT" value="Schrifttyp|Schriftgröße ">>
```

This parameter serves to define the font for the File-Open-dialog. Syntax: Font|Size; Example: „Arial|11“.

```
<param name="ALARMTABLEFONT" value="NAME|HEIGHT|WEIGHT|CHARSET|ITALIC|HORZ_ALIGN|VERT_ALIGN">>
```

This parameter serves to define the font for the alarm table. Syntax: NAME|HEIGHT|WEIGHT|CHARSET|ITALIC|HORZ\_ALIGN|VERT\_ALIGN

Example: „Arial|11|0|false|left|center“.

```
<param name="USECURRENTLANGUAGE" value="TRUE oder FALSE">>
```

If this parameter is set TRUE, the current language setting always will be synchronized between Web- and Target-Visualization (via implicit variable CurrentLanguage); i.e. at a language switch caused by an input in one of the both visualization types each in the other type the language will be switched too. (The CoDeSys Visualization currently is not included in this match).

## 8.4 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. Regard the effects of the currently active target settings, category Visualization (see CoDeSys User Manual CoDeSys\_V23\_E.pdf).
- 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'.
- Regard the possibility to use visualization elements for the purpose of switching between several controllers (see chapter 8.7), to which the WebServer then will connect automatically.
- If needed, modify the basic html-page WebVisu.htm (see chapter 8.2). For example instead of PLC\_VISU another visualization can be defined as start page of the Web-Visualization.
- Perform command 'Project' 'Clean all', then 'Project' 'Build'. Before downloading the project regard the current setting of option 'Prevent download of visualization files' in the target settings. This concerns all files which are used in the current visualization. Those can be bitmaps, language files and for Web-Visualization also XML description files.
- Log in to the target system ('Project' 'Online' 'Login') and start the project on the PLC.

## 8.5 Configuration and Start of the Web Server

---

- The Web Server must be available as an appropriate executable file (\*.exe) for the used target system.  
It also can be installed and started as a service. The following command line parameters can be used for this purpose when webserver.exe is started:
  - i = Web-Server gets installed as a service

- u = Web-Server service gets uninstalled
- s = Web-Server service gets started
- e = Web-Server service gets terminated

- The configuration of the server can be done by a configuration file or - restricted usable - 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-Visualization 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: ""
use-intel-byte-order	Type of byte-order (true or false): if "false, the data will get swapped (Motorola Byte-Order).	Default: true
The following entries refer to the use of a description file for the PLC-Handler in order to be able to use the MultiPLC-functionality. If these entries are missing, automatically the TCP/IP-connection to the above specified target system will be used.		
plc-description-file	Path of the ini-file for the PLCHandler, describing the communication parameters for all PLCs (controllers) to which the WebServer then can connect via Gateway/PLCHandler.(Per default the ini-file is in the same directory as webserver.exe). If the visualization (Web-Client) via INTERN-command CONNECT_TO tells to which PLC a connection should be established, this connection will be built automatically by the server and will be maintained until the next change.	Default: ""
plc-entries	Section containing the entries <plc-entry> for the various PLCs.	Default: ""
plc-entry	Entry for a PLC, contains PLC name and directory.	Default: ""
plc-name	Name of the PLC, as defined in the PLCHandler.ini-file.	Default: ""
plc-directory	Directory of the PLC files; can be defined absolutely or relatively to the file-upload directory (see above). Example: For the example configuration shown below a definition ".\FD" for the plc-directory results in the following location: "C:\Programme\CoDeSysV23\FD"	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>
<use-intel-byte-order> true </use-intel-byte-order>
<plc-description-file> PlcHandler.ini </plc-description-file>
<plc-entries>
  <plc-entry>
    <plc-name> MASTER </plc-name>
    <plc-directory> .\MASTER </plc-directory>
  </plc-entry>
  <plc-entry>
    <plc-name> FD </plc-name>
    <plc-directory> .\FD </plc-directory>
  </plc-entry>
  <plc-entry>
    <plc-name> DL </plc-name>
    <plc-directory> .\DL </plc-directory>
  </plc-entry>
</plc-entries>
</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]
```

Only these parameters can be used in a command line. The others described above, concerning byte-order and multiPLC-functionality must come via the configuration file.

Thus a call corresponding to the above shown configuration example would look like follows, whereby the settings for `<plc-description-file>` and `<plc-entries>` must be available in a configuration file:

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

---

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

---

## 8.6 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
```

Example:

```
http://localhost:8080/webvisu.htm
```

**WebVisu** is the default HTML-file. It contains a `<applet>` tag which will start the WebVisu-Applet so that the desired start page of the visualization will be displayed. Now you can start to operate the visualization.

---

**Note:** If the browser accepts cookies (if necessary, activate this option) the current language setting for dynamic texts will be stored and at the next call of the Web-Visualization will be re-used automatically. If the stored language is not available in the newly opened visualization, the default

language of this visualization will be used. If no default language is defined, the visualization texts will not be translated until the language gets switched explicitly.

## 8.7 Restrictions and Special Features

The following features are not yet realized in the Web-Visualization:

### INTERN commands

PRINT	Printout of the current visualization.
Execute external program	Execution of an external program.
LANGUAGEDIALOG	The configuration dialog containing the category 'Language' will be called.
EXITPROGRAM	The program will be exited. Can be realized via command INTERN LINK.
TRACE	Open Sampling Trace window. This function will be taken by the Trend element.
SAVEPROJECT	Saving the project. This function cannot be realized for the Web-Visualization.

### Accessing variables

Dynamic indexing within an Array	„Array1[Index].a“ is not possible, however „Array1[10].a“ is possible. The dynamic indexing will be realized with CoDeSys 3.0.
Replacing placeholders containing an expression	Placeholder: \$abc\$ + 5 Replacement: PLC_PRG.n + 500 -> should result in PLC_PRG.n + 500 + 5, this however is not possible in the Web-Visualization. This function will be realized with CoDeSys 3.0.
Pointer variables	Pointer variables like PLC_PRG.pdw2^ cannot be monitored. This function will be realized with CoDeSys 3.0.

### Others

Transparent bitmaps      Transparent bitmaps are not yet supported.

### Alarm handling

Actions	Action „Print“ is not yet supported.
Settings for sorting	The sorting within the alarm table, displayed via button 'History', always is according to date. The settings as done in the configuration of the alarm table element are not regarded.

The following features are **only** supported by the Web-Visualization:

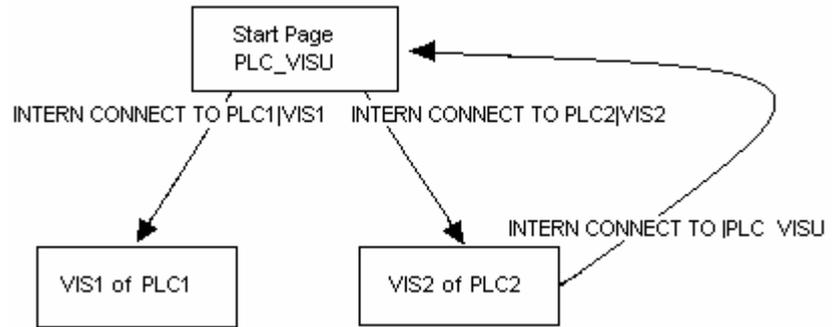
### INTERN commands

INTERN LINK	Via “INTERN LINK <URL>” a change to another web page can be defined. (Example: "INTERN LINK http://www.3s-software.com") “INTERN LINK <webaddress file>” opens a PDF file, which must be available on the server. (Example: "INTERN LINK http://localhost:8080/test.pdf ") “INTERN LINK <Email address>” opens a window for sending an E-Mail. (Example:"INTERN LINK mailto:support@3s-software.com")
INTERN CONNECT_TO <PLC-Name> <Start-Visu>	The target PLC can be changed if the WebServer is configured appropriately with the connection parameters for multiple PLCs (see Configuration and Start of the Web Server). PLC-Name:      Name of the PLC, as defined in the ini-file of the PLCHandler. Start-Visu:      Name of the visualization which should be displayed at

start of the Web-Visualization.

As soon as the WebServer gets the request for a PLC-change, it will automatically establish the connection to the respective PLC.

The following diagram shows a possible navigation scenario with multiple PLCs:



### Language for dynamic texts

If the web browser accepts cookies (if necessary, activate this option) the current language setting for dynamic texts will be stored and at the next call of the Web-Visualization will be re-used automatically. See the note in chapter 8.6.

## 9 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.

**Note:** Before download regard the current setting of option 'Prevent download of visualization files' in the target settings. This concerns all files which are used in the current visualization. Those can be bitmaps, language files and for Web-Visualization also XML description files.

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.

### 9.1 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. Among other things it provides functions for scanning mouse-actions performed by the user and language depending texts from a dynamic textlist, which thus can be directly used in the program. 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. Regard that also the following, implicitly used libraries will be loaded automatically as soon as the option 'Target-Visualization' is activated: SysLibAlarmTrend.lib, SysLibSockets.lib, SysLibFile.lib, SysLibTime.lib, SysLibMem.lib.)
- 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.).

### 9.2 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.)

A further improvement concerning the performance might be reached by switching off the **Keyboard usage for tables** (see CoDeSys V2.3 User Manual, App. H, Target Settings in Category Visualization).

Please regard also: In order to save memory space when later used in a Target-Visualization, the configuration dialogs for visualization elements might offer **restricted color- and font options**, defined by manufacturer-specific target settings.

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.

Consider whether **implicit visualization variables** in the current project should be handled as **remanent** variables and add the appropriate declarations in the Global Variables list (see 9.4).

**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. In the **Target Settings** in tab 'Visualization' (it depends on the target system, whether the particular options are available in this dialog!) configure that the project should be prepared for use in the Target Visualization: Activate option 'Target visualization'. Regard also the current setting of option 'Prevent download of the visualization files' in the target settings. This concerns all files which are used in the current visualization. Those can be bitmaps, language files and for Web-Visualization also XML description files.

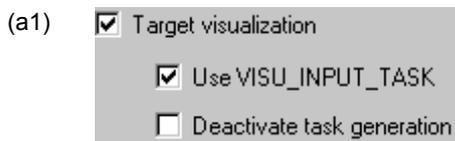
Additionally here you can define whether the user inputs and the re-painting of the visualization elements....

... should be controlled via VISU-tasks which are generated automatically or via individual programming: Activate option 'Deactivate task generation'.

... should be processed by one or by two POU's resp. tasks: Activate option 'Use VISU\_INPUT\_TASK'. (Don't get irritated by the term "...\_TASK" in this case; this option is also of effect if no VISU\_INPUT\_TASK is generated. Please see the descriptions in the following table.)

Thus, besides the possibility of deactivating the **keyboard usage for table**, the following configurations are possible:

- (a) The Target visualization is controlled by VISU-tasks which are generated automatically and which call the POU's MAINTARGETVISU\_PAINT\_CODE and MAINTTARGETVISU\_INPUT\_CODE: Deactivate task generation is switched off:



Use VISU\_INPUT\_TASK is activated:

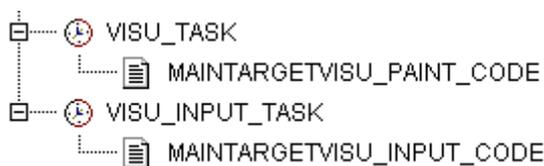
Automatically two tasks will be generated, each calling a program:

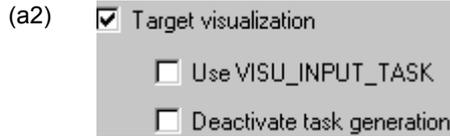
- VISU\_TASK calls the implicitly available POU MAINTARGETVISU\_PAINT\_CODE, which does the repainting of the visualization elements.
- VISU\_INPUT\_TASK calls the implicitly available POU MAINTARGETVISU\_INPUT\_CODE, which does the processing of the user inputs.

The default settings of the tasks:

- VISU\_INPUT\_TASK: cyclic, priority 14, interval t#200ms.
- VISU\_TASK: cyclic, priority 15, interval t#50ms.

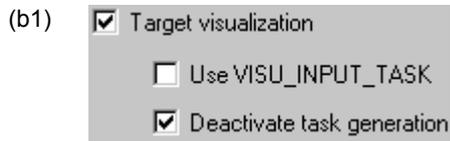
Of course the parameters can be modified. But: VISU\_INPUT\_TASK always should be processed before VISU\_TASK in order to guarantee a useful interaction of user inputs and update of the visualization. The task calling the main program (e.g. PLC\_PRG), should at least be processed as often as VISU\_INPUT\_TASK, ideally even with a higher priority, but it also could be added directly to VISU\_INPUT\_TASK.





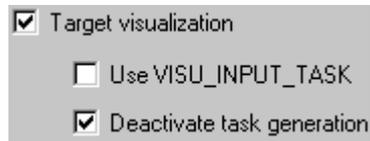
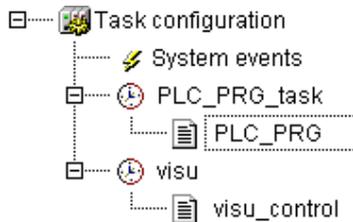
Use VISU\_INPUT\_TASK is deactivated:  
 Only task VISU\_TASK will be generated automatically, but in this case will include the functionality of VISU\_INPUT\_TASK. The implicit program POU MAINTARGETVISU\_PAINT\_CODE in this case additionally will include the functionality of program MAINTTARGETVISU\_INPUT\_CODE.  
 This configuration is intended for systems which do not allow multi-tasking. Disadvantageously no differentiated cycle times can be configured for the processing of user inputs and the repainting of the visualization elements, see above.

(b) The Target visualization is not controlled by automatically created tasks; the implicitly available POU MAINTARGETVISU\_PAINT\_CODE and MAINTTARGETVISU\_INPUT\_CODE can be called by the application program resp. can be appended to any task: Deactivate task generation is activated:



```

n:=n+1;
IF n=(n MOD 4) THEN
MAINTARGETVISU_INPUT_CODE();
END_IF;
MAINTARGETVISU_PAINT_CODE();
  
```



If additionally option Use VISU\_INPUT\_TASK is activated, both implicit POU are available and can be called individually resp. can be appended to any task. (Referring to this regard the hints in (a1).  
 Example for calls of the Target-Visualization POU in the application program: Here in program visu\_control the POU which processes the user inputs is only called after each fourth call of the repainting POU - thus reducing the danger of getting disturbed the repainting by another user input.

**When creating the application program, absolutely pay attention to reduce this danger of incorrect display !**

Example for use of any desired task for controlling the Target visualization POU;  
 here the above shown program visu\_control is called by task visu.

If option Use VISU\_INPUT\_TASK is not activated, only the implicitly available program POU MAINTARGET VISU\_PAINT\_CODE can be used, but in this case it will additionally include the functionality of MAINTARGET VISU\_INPUT\_CODE.

This POU can be called in the application program resp. can be appended to any task.

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

### 9.3 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 first one in the list of visualization objects in the Object Organizer.

### 9.4 Scan of mouse-clicks and dynamic texts

The library SysLibTargetVisu.lib, which gets included automatically as soon as target setting 'Target-Visualization' is activated, offers the following functions for scanning mouse-clicks performed by the user resp. entries of the dynamic textlist currently used for language dependent text display in the visualization.

#### Function GetText : BOOL

This function provides a language depending text from the currently used dynamic textlist.

Parameters:

stResult: STRING(256);	Serves as an IN_OUT parameter and gets assigned the text found by prefix string „stPrefix“ and ID „dwID“.
nResultLength:INT;	Define here the maximum length of the string „stResult“, if this is <256 characters.
stPrefix: STRING;	Prefix of the text entry in the currently used dynamic textlist.
dwID: DWORD;	ID of the text entry in the currently used dynamic textlist.

Return value:

FALSE – No text matching „stPrefix“ and „dwID“ was found.

TRUE - A text matching „stPrefix“ and „dwID“ was found.

#### Function GetTextByStringId : BOOL

This function - like function GetText - can provide a language depending text from the currently used dynamic textlist. The difference to GetText is: The ID of the text entry is given as a string in parameter „stID“ instead of a numeric value. Thus it is also possible to use IDs which are defined as strings in the xml-file for the dynamic texts (e.g. „Text123“).

stResult: STRING(256);	Serves as an IN_OUT parameter and gets assigned the text found by prefix string „stPrefix“ and ID „stID“.
nResultLength:INT;	Define here the maximum length of the string „stResult“, if this is <256 characters.
stPrefix: STRING;	Prefix of the text entry in the currently used dynamic textlist.
stID: STRING	ID of the text entry in the currently used dynamic textlist.

Return value:

FALSE – No text matching „stPrefix“ and „stID“ was found.

TRUE - A text matching „stPrefix“ and „stID“ was found.

**Function GetLastMouseDownEvent : BOOL**

This function provides information on the last performed MouseDown Event. It contains a pointer (pMouseEvent : POINTER TO MOUSEEVENT;) on structure MouseEvent consisting of the following parameters:

dwCounter : DWORD;	Number of MouseDown-Events since system start. With this parameters it can be evaluated whether no, one or multiple events have been prompted since the last request.
nXPos : INT;	Last mouse position in X/Y-coordinates.
nYPos : INT;	

Return value: No return value.

**Function GetLastMouseMoveEvent : BOOL**

This function provides information on the last performed MouseMove-Event. It contains a pointer (pMouseEvent : POINTER TO MOUSEEVENT;) on structure MouseEvent, see above: function GetLastMouseDownEvent.

Return value: No return value.

**Function GetLastMouseUpEvent : BOOL**

This function provides information on the last performed MouseUp-Event. It contains a pointer (pMouseEvent : POINTER TO MOUSEEVENT;) on structure MouseEvent, see above: function GetLastMouseDownEvent.

Return value: No return value.

## 9.5 Restrictions

---

Regard the following restrictions when using a visualization created in CoDeSys as Target-Visualization:

**INTERN commands**

PRINT	Printout of the current visualization is not possible.
EXITPROGRAM	This command is not supported.
TRACE	This command for opening the Sampling Trace window is not supported. This function will be taken by the Trend element.
SAVEPROJECT	This command for saving the project is not usable for the Target-Visualization.

**Grafic formats**

In the Target-Visualization currently only simple bitmaps are supported. Not yet supported formats: .jpg, .tif, .ico

**Texts**

Clipping	Texts exceeding the borders of an element currently do not get clipped.
----------	---

**Alarm handling**

Actions	Actions „Print“ and „Message“ currently are not supported. An appropriate warning will be dumped when compiling the project.
Settings for sorting (history)	The sorting within the alarm table, displayed via button 'History', always is according to date. The settings as done in the configuration of the alarm table element are not regarded.

### Trend

Degree of accuracy      The degree of accuracy can be defined for the horizontal axis. This effects that the number of readout points in the "duration" area can vary. The number of readout points results from:  
Number of readout points = Duration/Degree of accuracy  
The number of readout points on some devices effects that the cycle time for VISU\_TASK exceeds 200 ms, which will lead to a CPU load of 100% !

### Others

Slider in table      The slider for scrolling is not displayed.

Zoom to vis.      'Zoom to vis.' will not work in Target-Visualization, if placeholders are used.

Input on elements behind transparent elements      If a transparent rectangle, polygon or ellipse element "lies on" another element, the subjacent element will be reached by a mouse-click on the transparent area of the overlying element, like it would be positioned behind an inserted visualization reference.  
This is different to the default behaviour in all other visualization types: There you can click „through“ visualization references, but not through transparent rectangles, polygons etc.

## 10 APPENDIX

### Appendix A Implicit Variables in the Visualization

The following system variables might be used when programming a visualization:

Implicitly generated variable	Data type	Function	Currently usable in			
			HMI	Simulation	Target Visu	Web Visu
CurrentVisu	String[40]	Name of the currently opened visualization. If the name gets changed, a change to another visualization will be done. <b>Note:</b> With compiler versions < V2.3.7.0, and if the library SysLibStr.lib is not included in the project, the name string MUST be defined in <b>capital letters</b> .	x	x	x	x
CurrentCaller	String[40]	Name of the previously opened visualization. Is used for the ZOOMTOCALLER functionality. Only set and automatically modified in a Target-Visualization.	-	-	x	-
CurrentLanguage	String[40]	Currently set language, available in the language file. The language. Only set and automatically modified in a Target-Visualization.	-	-	x	-
CurrentUserLeve *	INT	Currently set user level 0..7.	x	x	x	x
CurrentPasswords[0 .. 7] *	ARRAY [0..7] of String[20]	All passwords which are defined in CoDeSys in "Usergroup passwords".	x	x	x	x
CurrentWriteAccessClientId	DWORD	ID of the visualization client which currently has the write access on the PLC data (see Kap.5.4).	-	-	x	x
CurrentClientId	DWORD	ID of the current visualization client (see Kap.5.4).	-	-	x	x

#### Implicit variables as remanent variables:

The implicit variables CurrentLanguage, CurrentUserLevel and CurrentPasswords[0..7] can be declared as **remanent** variables for the use in a Target-Visualization.

For this purpose the variables must be declared explicitly as **global variables**. This declaration MUST be done in the topmost global variables list in folder 'Global Variables' in the Resources tab. If the declaration is placed in another global variable list, a compile error will occur.

Example:

```
VAR_GLOBAL RETAIN
  VisuDoExecuteUserlevelInit : BOOL := TRUE;
  CurrentUserLevel : INT := 0;
  CurrentPasswords : ARRAY[0..7] OF STRING[20] := 'a','b','c','d','e','f','g','h';
END_VAR
```

\* **Regard** for the variables `CurrentUserLevel`, `CurrentPasswords[...]`: Those must be of the same type (normal, RETAIN, PERSISTENT...)! If they are defined as remanent variables, additionally a variable `"VisuDoExecuteUserlevelInit"` of type `BOOL` must be declared as a RETAIN variable in the global variables list, initialized with `TRUE` (`" VisuDoExecuteUserlevelInit : BOOL := TRUE;"`).

## Appendix B Example Dll for ActiveX Control

---

See in the following an example of a source file of a Dll used for an ActiveX-Element in the CoDeSys visualization.

This sample dll will only call the methods GoBack or GoForward of the control if this supports the IWebBrowser interface (see chapter 2.4.1, ActiveX-Element).

The method is chosen by the parameter pszId.

```
#include "stdafx.h"
#include <unknwn.h>
#include <exdisp.h>
BOOL APIENTRY DllMain( HANDLE hModule,
                      DWORD  ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    return TRUE;
}
extern "C" __declspec (dllexport) void ExecuteActiveXCall(IUnknown* pUnk,
char* pszId, char* pszParam, char* pszReturnBuffer, int nReturnBufferSize,
DWORD* pdwReturnFlag)
{
    if (strcmp(pszId, "IWebBrowser|GoBack") == 0)
    {
        IUnknown* pNewUnk;
        IWebBrowser* pwb;
        pUnk->QueryInterface(IID_IWebBrowser, (void**) &pNewUnk);
        pwb = (IWebBrowser*) pNewUnk;
        if (pwb)
        {
            pwb->GoBack();
            pwb->Release();
        }
    }
    else if (strcmp(pszId, "IWebBrowser|GoForward") == 0)
    {
        IUnknown* pNewUnk;
        IWebBrowser* pwb;
        pUnk->QueryInterface(IID_IWebBrowser, (void**) &pNewUnk);
        pwb = (IWebBrowser*) pNewUnk;
        if (pwb)
        {
            pwb->GoForward();
            pwb->Release();
        }
    }
}
```



## Appendix C Possible Key Combinations

The table below shows all key combinations usable for keyboard usage (see chapter 2.4.2) in the particular visualization variants. The following abbreviations are used:

- **C** for CoDeSys/CoDeSysHMI
- **TV** for Target-Visualization
- **WV** for Web-Visualization

If an abbreviation is displayed in a column, the corresponding key (combination) is supported by the concerning visualization variant. The comments for the particular lines you find below in a subsequent table.

	no modifier	Shift	Ctrl	Shift+Ctrl	comment
VK_TAB	C	C	C	C	K4
VK_RETURN	C / TV	C / TV	C / TV	C / TV	
VK_SPACE	C / WV	C / WV	C / WV	C / WV	K4;K5
VK_ESCAPE	C / TV / WV	C / TV / WV			K3
VK_INSERT	C / TV	C / TV	C / TV	C / TV	
VK_DELETE	C / TV	C / TV	C / TV	C / TV	
VK_HOME	C / TV / WV				
VK_END	C / TV / WV				
VK_PRIOR	C / TV / WV				
VK_NEXT	C / TV / WV				
VK_LEFT	C / TV / WV				
VK_RIGHT	C / TV / WV				
VK_UP	C / TV / WV				
VK_DOWN	C / TV / WV				
F1	C / TV / WV	K1			
F2	C / TV / WV				
F3	C / TV / WV				
F4	C / TV / WV				
F5	C / TV / WV				
F6	C / TV / WV				
F7	C / TV / WV				
F8	C / TV / WV				
F9	C / TV / WV				
F10	C / TV	C / TV	C / TV	C / TV	K2
F11	C / TV / WV				
F12	C / WV	C / WV	C / WV	C / WV	
0	C / TV / WV				
1	C / TV / WV				
2	C / TV / WV				
3	C / TV / WV				
4	C / TV / WV				

Appendix C

	<b>no modifier</b>	<b>Shift</b>	<b>Ctrl</b>	<b>Shift+Ctrl</b>	<b>comment</b>
5	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
6	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
7	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
8	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
9	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
A	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
B	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
C	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
D	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
E	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
F	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
G	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
H	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
I	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
J	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
K	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
L	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
M	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
N	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
O	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
P	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
Q	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
R	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
S	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
T	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
U	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
V	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
W	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
X	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
Y	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
Z	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
VK_NUMPAD0	C / TV / WV	C / TV / WV			
VK_NUMPAD1	C / TV / WV	C / TV / WV			
VK_NUMPAD2	C / TV / WV	C / TV / WV			
VK_NUMPAD3	C / TV / WV	C / TV / WV			
VK_NUMPAD4	C / TV / WV	C / TV / WV			
VK_NUMPAD5	C / TV / WV	C / TV / WV			
VK_NUMPAD6	C / TV / WV	C / TV / WV			
VK_NUMPAD7	C / TV / WV	C / TV / WV			
VK_NUMPAD8	C / TV / WV	C / TV / WV			

	<b>no modifier</b>	<b>Shift</b>	<b>Ctrl</b>	<b>Shift+Ctrl</b>	<b>comment</b>
VK_NUMPAD9	C / TV / WV	C / TV / WV			
VK_ADD	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
VK_SUBTRACT	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
VK_MULTIPLY	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	
VK_DIVIDE	C / TV / WV	C / TV / WV	C / TV / WV	C / TV / WV	

K1	In CoDeSys additionally the online help will be opened
K2	In the Web-Visualization and the IExplorer the File Menu will be opened
K3	Ctrl/Esc opens start menu , Shift/Ctrl/Esc opens TaskManager
K4	Tab and Space have a different function in the Target-Visualization
K5	Shift/Space opens the applet configuration



# Index

---

## A

Access protection for vizualization 5-2  
 Access rights in Visualization 2-22  
 Acknowledge variables 2-39  
 ActiveX-Element  
   Condition for call 2-32  
   Configuration 2-31  
   Dll for Call 2-32, 10-3  
   Insert 2-5  
   Method call 2-32  
   Variable for URL 2-32  
 Alarm class 2-39  
 Alarm handling  
   Target.-Visualization 9-5  
 Alarm handling  
   Web-Visualization 8-7  
 Alarm table 2-6  
 Alarm table configuration 2-39  
 ALARMTABLEFONT 8-4  
 ALARMUPDATEBLOCKSIZE 8-3  
 Align 2-8  
 Angle 2-11  
 array  
   indexing 8-7  
 Array 2-27, 2-38  
 ASSIGN 2-51

## B

Background Bitmap 2-54  
 Bar Display 2-5  
 Bar Display in Visualization 2-37  
 bitmap 2-57  
 Bitmap 2-4, 2-47, 2-54  
 Bitmap configuration 2-47  
 bitmap list 2-47  
 Bitmap list 2-57  
 Bitmap variable 2-47  
 bitmaps  
   Web-Visualization 8-7  
 bmp-file 2-57  
 Button 2-4

## C

CAM 2-52  
 CHANGEPASSWORD 2-51  
 CHANGEUSERLEVEL 2-51  
 Changing the Selection and Insert Mode 2-7  
 Check visualization variables on compile 2-54  
 Clear Background Bitmap 2-54  
 CNC 2-52  
 CoDeSys HMI 7-3  
 CoDeSys HMI  
   Special configuration possibilities in Visualisation 2-50  
 CoDeSys HMI 2-50  
 COLCONST 2-28  
 Color in visualization elements 2-10  
 Color Variables in Visualization 2-17  
 Colors in Visualization 2-16  
 Columns 2-39

COMPRESSEDFILES 8-2  
 Configuration 2-10  
 Configuration of a Web-Server 8-4  
 Configure programs 2-22  
 Configure Visualization 2-9  
 Configure visualization elements 2-10  
 Copying Visualization Elements 2-7  
 Create visualization 2-2  
 CurrentCaller 10-1  
 CurrentClientId 10-1  
 CurrentClientId 5-2  
 CurrentLanguage 10-1  
 CurrentPasswords 10-1  
 CurrentUserLevel 10-1  
 CurrentVisu 10-1  
 CurrentWriteAccessClientId 10-1  
 CurrentWriteAccessClientId 5-2  
 Curve 2-3

## D

Data array 2-27  
 Deactivate task generation 9-2  
 Default font 3-5  
 Default language 3-5  
 DEFAULTENCODING 8-3  
 DEFINERECEIPT 2-51  
 DELAY 2-51  
 Display in visualization 2-53  
 download Web-Visualization 8-1  
 Dragging Visualization Elements 2-7  
 dynamic images 2-47  
 dynamic language switching 3-3  
 dynamic texts 3-3  
 dynamic texts.lib 9-4

## E

Elementlist 2-8  
 Ellipse 2-3  
 ENCODINGSTRING 8-3  
 error messages for Web-Visualization 8-1  
 error\_ini.xml file 8-1  
 ERROR\_SENSITIVITY 8-3  
 Execute program 2-22, 8-7  
 EXIT 9-5  
 EXITPROGRAM 2-51, 8-7  
 Extras  
   Align 2-8  
   Bitmap list 2-57  
   Clear Background Bitmap 2-54  
   Configure 2-10  
   Elementlist 2-8  
   Group 2-7  
   Keyboard Usage 2-54  
   List of placeholders 4-1  
   Select 2-7  
   Select All 2-7  
   Select Background Bitmap 2-54  
   Send to Back 2-8  
   Send to Front 2-8  
   Settings 2-53  
   Ungroup 2-8

**F**

File  
 Print in online mode 5-2  
 FILEOPENSAVEDIALOGFONT 8-4  
 Font  
 Unicode 3-1  
 Font in visualization elements 2-10  
 FORCEDLOAD 8-2  
 Formatting strings 2-12  
 Frame 2-53

**G**

GetLastMouseDownEvent 9-5  
 GetLastMouseMoveEvent 9-5  
 GetLastMouseUpEvent 9-5  
 GetText 9-4  
 GetTextByStringId 9-4  
 GETWRITEACCESS 2-52  
 GETWRITEACCESS 5-2  
 GLOBALRELEASEWRITEACCESS 2-52, 5-2  
 Grid 2-54  
 Group configuration 2-50  
 Grouping visualization elements 2-7, 2-50

**H**

height of visualization screen 8-2  
 HELP 2-52  
 Help file in Visualization 3-8  
 Hide elements, if no replacement took place 3-3  
 Histogram 2-6  
 Histogram configuration 2-38

**I**

ID 3-3, 3-4  
 image file formats 2-57  
 images for dynamic switching 2-57  
 images in visualization 2-57  
 Implicit Variables 10-1  
 INDEX 2-28  
 Input 2-21  
 input on transparent elements 9-6  
 Insert  
 ActiveX-Element 2-5  
 Alarm table 2-6  
 Bar Display 2-5  
 Bitmap 2-4  
 Button 2-4  
 Curve 2-3  
 Ellipse 2-3  
 Histogram 2-6  
 Meter 2-5  
 Pie 2-3  
 Polygon 2-3  
 Polyline 2-3  
 Rectangle 2-3  
 Rounded Rectangle 2-3  
 Table 2-5  
 Trend 2-6  
 Visualization 2-4  
 WMF file 2-5  
 Insert Metafile 2-5  
 Instance of a visualization 2-10

INTERN CONNECT\_TO 8-7  
 INTERN GETWRITEACCESS 2-52, 5-2  
 INTERN GLOBALRELEASEWRITEACCESS 2-52, 5-2  
 INTERN LINK 8-7  
 INTERN LINK 2-52  
 INTERN RELEASEWRITEACCESS 2-52, 5-2  
 Invisible scrollbar 2-34

**J**

jpg-file 2-57

**K**

key combination 10-5  
 Keyboard operation 5-1  
 keyboard usage 10-5  
 Keyboard usage in Visualization 2-54  
 KEYBOARDUSAGEFROMDIALOGS 8-3  
 KEYPADINDIALOGS 8-3

**L**

language 3-3  
 Language 3-5  
 LANGUAGE 2-51  
 LANGUAGE DEFAULT 2-51  
 language file 3-2  
 Language in visualization 2-54  
 language setting in Web-Visualization 8-6  
 language switching 9-4  
 language switching 3-1, 3-3  
 language switching configuration 3-3  
 LANGUAGEDIALOG 2-51, 8-7  
 Line width in Visualization 2-15  
 LINK 2-52  
 List of placeholders 4-1  
 LOADWATCH 2-51

**M**

MAINTARGETVISU\_INPUT\_CODE 9-2  
 MAINTARGETVISU\_PAINT\_CODE 9-2  
 Master layout 2-56  
 Metafile 2-5  
 Meter 2-5  
 Meter element in Visualization 2-34  
 Method call 2-32  
 Modifying Visualization Elements 2-7  
 Motion absolute in Visualization 2-18  
 Motion relative in Visualization 2-19  
 MouseEvent 9-5  
 Mult-Client Visualization 5-2

**O**

Object properties 2-56  
 online change Web-Visualization 8-1  
 Online Help in visualization 3-8  
 Online Mode  
 Visualization 5-1  
 Online Mode 5-1  
 Operating version 2-50  
 Operation over the keyboard in online mode 5-1

**P**

Password \\* MERGEFORMAT 10-1  
 Pie 2-3  
 Placeholder 2-10, 2-49, 8-7  
 Placeholder Concept 4-1  
 Placeholder List 4-1  
 Placeholders in Target-Visualization 9-6  
 PLC\_VISU 8-4, 9-1  
 PLCSTATEINTERVAL 8-3  
 pointer variables in Web-Visualization 8-7  
 Polygon 2-3  
 Polyline 2-3  
 Position 2-8, 2-9  
 PREFIX 3-3, 3-4, 3-6  
 Prefix-ID-combination 3-4  
 Prefix-ID-combination 3-3  
 PRINT 9-5  
 PRINT 2-51, 8-7  
 Print in online mode 5-2  
 Programmability 2-23  
 Properties of visualization object 2-56

**R**

READRECEIPT 2-51  
 Rectangle 2-3  
 Reference 2-10  
 RELEASEWRITEACCESS 2-52, 5-2  
 reload of Web-Visualization 8-1  
 Remanent variables for Target-Visualization 10-1  
 Replace placeholder 2-49  
 Rounded Rectangle 2-3  
 ROWCONST 2-28

**S**

SAVEPROJECT 9-5  
 SAVEPROJECT 2-51, 8-7  
 SAVEWATCH 2-51  
 Scrollbar 2-33  
 Security in Visualisierung 2-22  
 Select 2-7  
 Select All 2-7  
 Select Background Bitmap 2-54  
 Selecting elements in Visualization 2-7  
 Selecting Visualization Elements 2-6  
 SELECTION 8-3  
 Send to Back 2-8  
 Send to Front 2-8  
 Settings in visualization 2-53  
 Shape 2-11  
 Simplified input handling 5-1  
 Slider 2-34  
 Slider 2-39  
 Slider size 2-27  
 STARTVISU 8-2  
 static language switching 3-1  
 Status Bar 2-9  
 status check for Web-Visualization 8-1  
 SUPPORTTOOLTIPSINALARMTABLE 8-4  
 switch images 2-57  
 SysLibTargetVisu.lib 9-4  
 SysLibTargetVisu.lib 9-1  
 System Variables 10-1

**T**

Table  
   Target-Visualization 9-6  
 Table 2-5, 2-27  
 Table element configuration 2-27  
 Tap FALSE 2-21  
 Tap variable 2-21  
 Target Settings  
   Target-Visualization 9-2  
 Target-Visualization  
   Activate 9-2  
   Implicit Variables 10-1  
   PLC\_VISU 9-1  
 Target-Visualization 2-56, 9-1  
 Template for table element 2-28  
 Text 9-5  
 Text 2-12  
 Text in Visualization 2-12  
 textlist 9-4  
 Textvariables in Visualization 2-14  
 tif-file 2-57  
 tit-Datei 2-51  
 tit-file 3-1  
 tit-file 3-2  
 Toggle variable 2-21  
 ToolTip in Visualization 2-22  
 Tooltip-display 2-34  
 TOOLTIPFONT 8-4  
 TRACE 9-5  
 TRACE 2-52, 8-7  
 translation file 3-2  
 Translation file  
   Visualization texts 2-12  
 transparent bitmaps 8-7  
 transparent visualization element 9-6  
 Trend  
   Target-Visualization 9-6  
 Trend 2-6, 2-42  
 Trend configuration 2-42

**U**

Unicode 3-1  
 UPDATETIME 8-2  
 URL 2-32  
 URL-connection for Web-Visualization 8-3  
 Use as master layout 2-56  
 Use as Target-Visualization 2-56  
 Use as Web-Visualization 2-56  
 Use VISU\_INPUT\_TASK 9-2  
 USECURRENTVISU 8-2  
 USEFIXSOCKETCONNECTION 8-2  
 User level 10-1  
 USEURLCONNECTION 8-3

**V**

Variables in Visualization 2-20  
 vis-Datei 2-51  
 vis-file 3-1  
 vis-file 3-2  
 Visualisation  
   Input possibilities for CoDeSys HMI 2-50  
 Visualization  
   Compile 2-54  
   Display 2-53

## Appendix C

- Frame 2-53
- Grid 2-54
- Insert 2-4
- Instance 2-10
- Language 2-54
- Master layout 2-56
- Placeholder 2-10
- Visualization as Visualization element 2-48
- Visualization Editor 2-1
- Visualization Elements 2-2
- visualization files 5-1, 9-2
- visualization name 2-2
- Visualization Object
  - Configure... 2-53
- Visualization without master layout 2-56
- VisuDoExecuteUserlevelInit 10-2

### W

- Web Server
  - Configuration and Start 8-4
  - Configuration file 8-5
  - File upload directory 8-5
- webserver\_conf.xml 8-5
- WebVisu.htm 8-2

- Web-Visualization
  - Auto-Reload 8-1
  - language setting 8-6
  - PLC\_VISU 8-4
  - status check 8-1
  - WebVisu.htm 8-2
- Web-Visualization 2-50, 2-56, 8-1
- width of visualization screen 8-2
- WMF file 2-5
- Write access for visualization clients 5-2
- WRITEACCESSLOCK 8-3
- WRITERECEIPT 2-51

### X

- xml-file for dynamic texts 3-4

### Z

- zip-files for Web-Visualization 8-2
- Zoom to vis. 9-6
- Zoom to Vis. 2-21, 2-49
- ZOOMTOCALLER 10-1