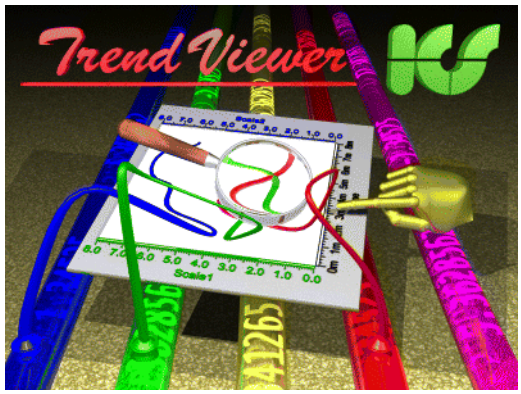
Trend Viewer



Active-X Control for the display and analysis of measured curves

User's Manual

Version 3.1

TrendViewer License Agreement

Active-X Control for the display and analysis of measured curves

The following licensing terms are valid for the for the use of TrendViewer:

1. By accepting this license agreement the licensee acquires a temporally unlimited, locally unlimited single non-exclusive right to use this product. The use of this product for the development of applications is only permitted to the licensee. The license is only valid for one workstation. The dissemination of this product as component of an application developed by the licensee is not restricted.

2. The programs contained in this product and other files may not be passed onto third parties, distributed or simultaneously used on several computers neither in their source code nor as executable programs - nor in modified form -. Also the installation of one and the same license on more than one computer or the multiple use in the network - for example in companies, schools or universities - is prohibited.

3. The *TrendViewer* developer license is delivered in form of a text file. It is named trendviewer3.lic and must not be passed on/handed out with the developed application. No license file must be delivered with the applicaton in which *TrendViewer* is integrated.

If *TrendViewer* is used in the project engineering of process displays in control systems, the license file trendviewer3.lic must only be installed on the editor's workplace.

4. All extensive/further rights are reserved to the holder of the Copyright. The delivered software must neither be diassembled, decompiled or otherwise retranslated.

5. Changes in the software and documentation are subject to change without notice. Update-Service and Support are only granted to registered licensees.

6. The program and documentation were carefully developed and checked. Unless in the event of gross negligence or intent ICS does not assume any liability for possible damages related to the use of this software or its documentation. Furthermore, unless otherwise explicitly guaranteed in writing, no warranty is given for the compatility of this software with any arbitrary programs or hardware components.

7. The license terms also extend to any possible updates or amendments - also subject to change without prior notice -.

8. Should one of the preceding articles of the agreement become invalid, this shall not affect the remaining articles.

All trademarks used in the text are accepted.

Contents

1. Introduction

Field of application

TrendViewer is an Active-X control with high usability which is embedded into applications to display and visually analyze measured data. *TrendViewer* can also be integrated into user interfaces of control systems, programs for data acquisition and analysis of measurements or in machine controls, which support the use of Active-X controls.

TrendViewer offers a so-called "Zoomable User Interface" and especially excels in highly-flexible and precise display of scales of various types. The large number of curves and scales which are processed simultaneously in *TrendViewer* almost present no limits to the requirements of reediting very large sets of measured data. The numerous additional objects and interactive properties allow the user to maintain a clear view in very complex curve graphs and display the desired data immediately and directly. Additional functions support the application in the visual evaluation of large data sets.

Above all, *TrendViewer* is suitable for interactively selecting panes with the mouse from extensive measuring curves like they are e.g. recorded in the process industry (pharmacy, chemistry) for reasons of quality control. The pane is displayed immediately in its dimensions and its ratio to the entire amount of measuring data during the mouse movement. *TrendViewer* is transparent concerning the data source. Online trend displays as well as archives can be edited just as easily in *TrendViewer*.

Two task definitions are especially supported by *TrendViewer*:

- the representations of connections between measured curves over time in phase diagrams. By interactive positioning of scales the connections can be redefined easily
- the direct comparison of measured curves of a measuring point in different time intervals. The time intervals can be changed easily with the mouse and shifted independently with the 2 x-scales.

Licensing procedure

The control is offered to software developers or project engineers initially as royalty-free test version such that the features of the control can be tested in the projected application. The functionality of the royalty-free version is not restricted.

If *TrendViewer* is to be passed on to end customers in an application, a developer license must be purchased. Otherwise a message is displayed informing the user that he is using an unlicensed version of *TrendViewer*. No additional license is required during runtime.

Installation

TrendViewer is offered as internet download and on data medium (CD). The installation file contains the trend control, the documentation and a test routine allowing to set the properties of *TrendViewer* and compare them with the previous version of this product as well as programs in source code to demonstrate the integration of *TrendViewer* in applications.

Computer and software requirements

As *TrendViewer* is a very dynamic control, particularly in its interactive use, the computer on which the finished application runs should at least meet the following performance requirements depending on the operating system so that *TrendViewer* can be used ergonomically:

Personal Computer with Pentium IV 200 MHz, 128 MByte RAM, mouse with 2 or 3 keys or a similar pointing device:

If *TrendViewer* is used in the user interface of a control system, the requirements published by the manufacturer of the control system must be respected.

The following software is required:

Microsoft Windows NT 4.0, Windows 2000, Windows XP, Windows CE or more recent Windows version.

A development environment like e.g. Microsoft Visual Studio is required to integrate *TrendViewer* into an application. Please use the editors of the respective control systems for the integration in process displays.

Licensing

Licensing is conducted by internet. The name of the person, the name and department of the company and its location are necessary to issue a license file. This information can be provided by completing a form available at http://www.icsgmbh.de or via the electronic shop. The purchaser is sent a personalized license file by e-mail.

The *TrendViewer* developer license is delivered in form of a text file. It is named trv30developer.lic and must by no means be passed on with the developed application. No license file must be delivered together with the application into which *TrendViewer* is integrated.

If *TrendViewer* is used in project engineering of process displays in control systems, the installation of license file trv30developer.lic is only required on the editor workstation.

Product Types

TrendViewer is available in 3 performance classes which can be released by different license files:

Product Type	Properties
Without license	Active-X Component for the display of up to 512 binary or analog data curves with 512 y-scales, each with linear or logarithmic scaling, 2 x-scales. The x-axes display time, duration, angle or value. Display of time, locus and phase diagrams. All functions can be tested without a license except the additional functions only contained in the Professional Edition
Standard Edition	keys to show and hide the curves, circular buffer for over 500 binary and analog measured curves, free assignment to x- and y-scales, interactively editable curve points, 2 adjustable limit lines for each curve, flood curves, flooding curve versus line, arbitrary line styles and multiple fill styles, 2 time scales for measuring period or absolute time for interactive curve comparison, more than 500 linear and logarithmic x- and y-scales, cyclic degree scales, value-related equidistant auxiliary lines for each scale to support qualititive readability, overview bar for the comparison of a curve section with a measured interval, tooltip with value information at mouse position, three-stage scale gradations, (y,t)-, (x,y)- and phase diagrams, vertical and horizontal display, automatic adaptation to the window size, adjustable curve field size, strip charts, lasso zoom, follow me“ keys for plotter functions, curve capture function, rulers for all scales and curves, adjustable angle of inscription, localized scale inscription via XML-file, scale inscription with arbitrary character sets, supports localized time zones
Professional Edition	The *TrendViewer* Professional Edition offers the same features as the *TrendViewer* Standard Edition. In addition, the following features are only available in the *TrendViewer* Professional Edition: envelope curve display, movable curves, interactive parallel or stacked scale positioning with target recognition

Support

If technical support is required for the use of *TrendViewer*, please e-mail your questions together with the license file to support-trv@icsgmbh.de. Information about new releases and patches is published on the website www.icsgmbh.de.

Information about new *TrendViewer* versions or about other products can be requested in the order or by subscribing to one of the mail lists on the website of ICS GmbH.

Address:

ICS Gmbh
Pforzheimerstr. 202c
D76275 Ettlingen
Germany

Phone: ++49/7243/7127-0
Fax: ++49/7243/7127-29

www.icsgmbh.de

2. Diagram

Basic but also very complex representations of measured curves can be parameterized in *TrendViewer*. In each diagram up to 512 curves for analog or binary measuring points can be assigned individually or in groups to as many y-scales.

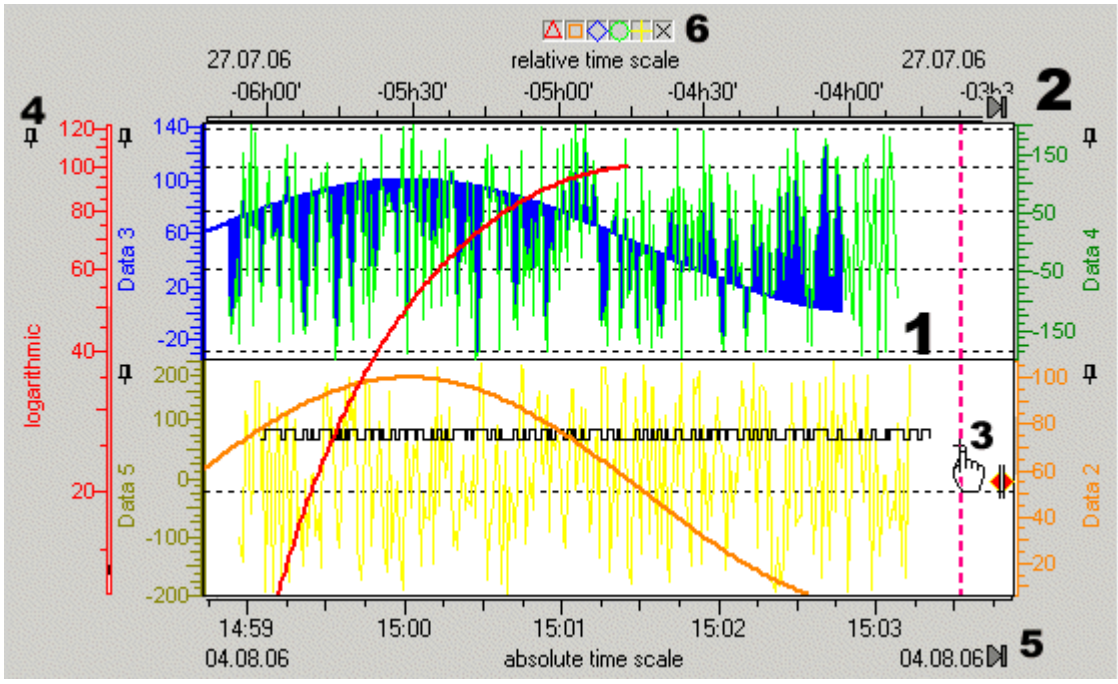
As the curves can be assigned to one of the two independent x-scales, the trends of a measuring point can be viewed and compared in one of the same diagram at various time intervals.

A phase diagram is a special type of representation in which the connection between measured curves of several measuring points is shown parallel to the time axis. The relation between the measured curves can be displayed directly by the interactive arrangement of the data scales.

Curves can be arranged in strips one above the other in the trend window or can be drawn across the entire trend window. Also combinations of both representations are possible, such that complex situations can be displayed in a diagram.

2.1 Components

The following illustration shows and designates all components of a diagram which *TrendViewer* offers on the user interface.



The numbers in the illustration have the following meaning:

1) Trend window with 2 strips. The upper strip shows the curves related to scales **Data 3** and **Data 4**, the lower strip displays curves related to scales **Data 2** and **Data 5**. The curve of scale **logarithmic** runs across the entire trend window. The auxiliary lines are predefined by the logarithmic scale. In addition, a read-line (3) used by *TrendViewer* to return curve data to the individual position of the application is shown.

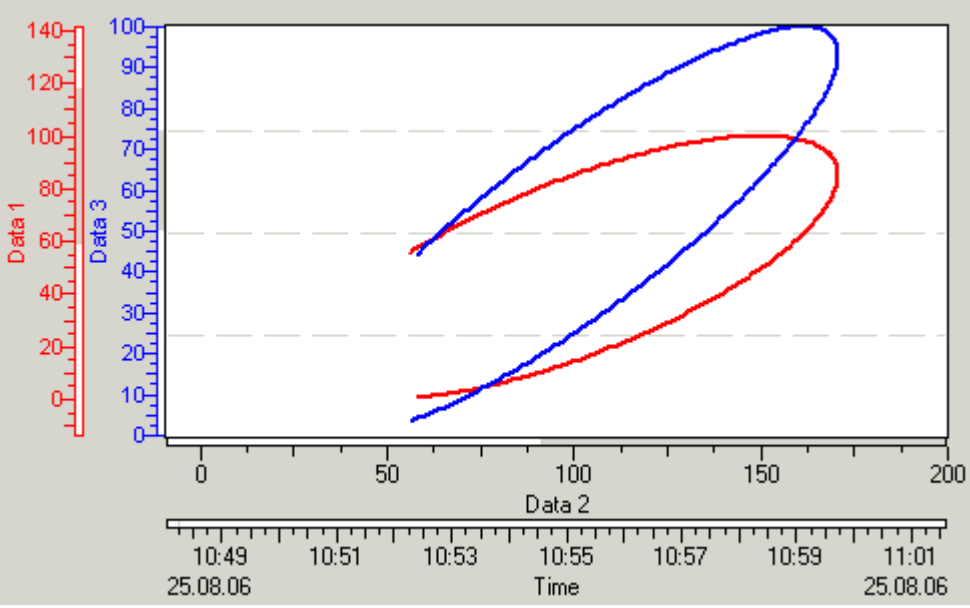
   The upper strip shows 2 curves for analog measuring points, which are flooded against one another. The bottom strip also shows a curve for a binary measuring point (black) to which no scale is assigned.

   Curves can be shown and hidden by curve buttons (6).

2) Scales are arranged around the trend window. A time diagram e.g. contains a time scale on the top and the bottom and data scales on the left and right. The data scales can be stacked. Scales can be interactively moved to other positions via scale buttons (4) (Professional License required). The "Follow-Me" buttons (5) provide that the trend window behaves like a plotter for current data.

The components and their use are described in detail in the following sections.

In a phase diagram the components are arranged as follows:



The curves associated to **Data 1** and **Data 3** are plotted against the curve of scale **Data 2**. A locus is thereby created. In addition, the time axis is displayed parallel to scale **Data 2**. When the displayed time interval is reduced or extended, the locus is reduced or extended correspondingly. If the time interval is changed, another cutout of the locus can be seen.

2.2 General Properties

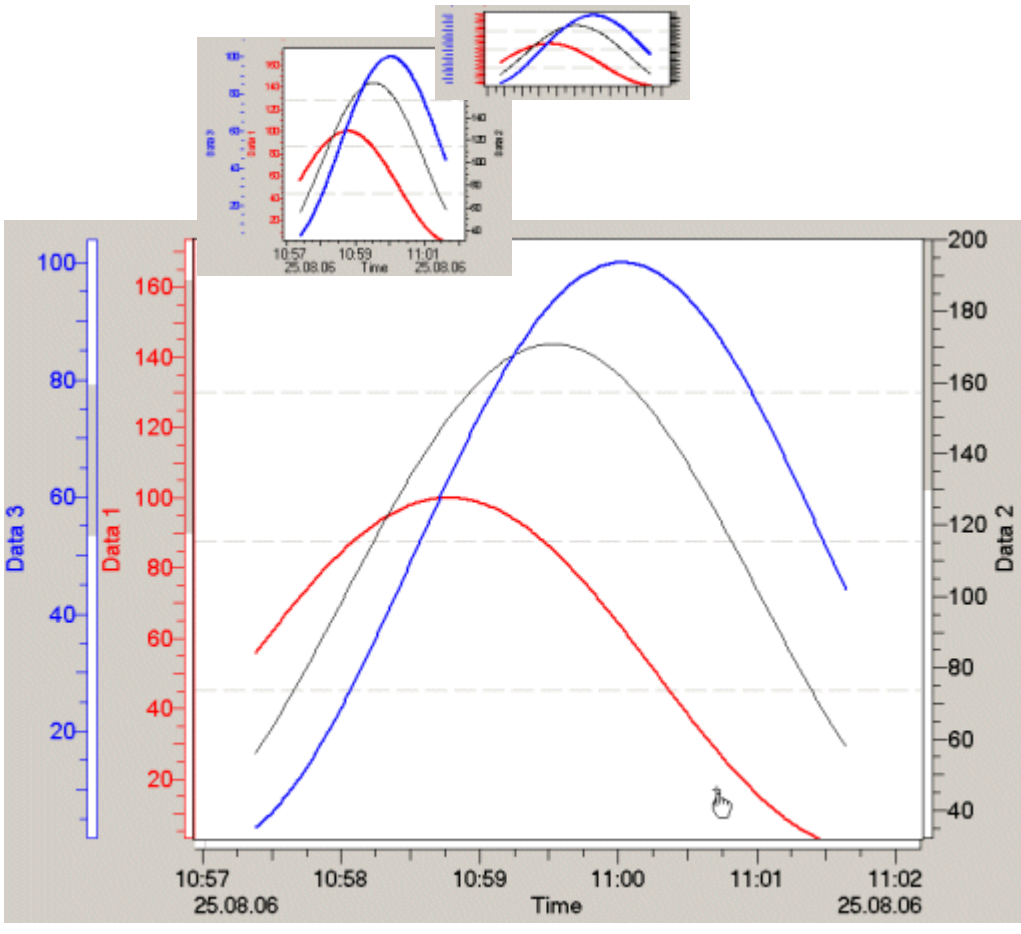
In the abundance of features offered in *TrendViewer* a number of characteristics influence the look and operability of a diagram with overall effect.

Note: In the following descriptions of *TrendViewer* components and their interactive use, horizontal alignment of the diagram is always described. The x-axes run horizontally and the y-axes vertically. In diagrams in rotated state the descriptions are to be comprehended correspondingly.

2.2.1 Change the size of a diagram

If *TrendViewer* is used in a Windows application of adjustable size, the components of the diagram are adapted as far as possible to this size. The font size of the scale inscription is automatically adapted if it was set to variable. In extreme cases, the scale inscriptions or also entire parts of the trend window - if the size was predefined directly - can be left out.

Different sizes of the same diagram:



2.2.2 Define the size and position of the trend window

There are two methods to define the size of the trend window:

indirectly

First, the size of the space required for the scales is computed from data of the layout, stroke width, font size and additional elements like overview bars or curve buttons. The remaining space is available to the trend window.

directly

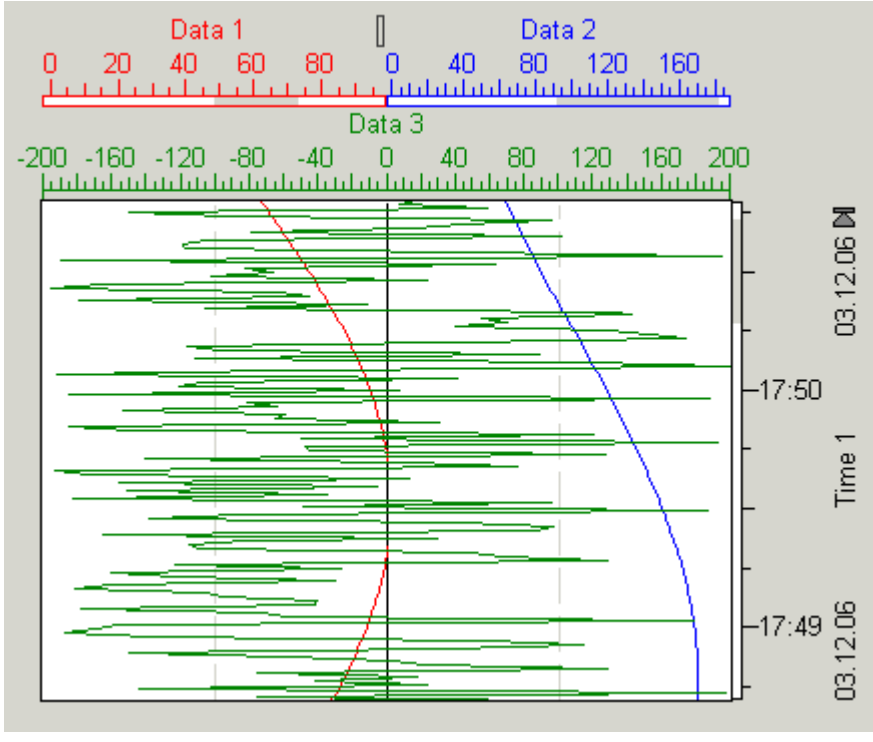
The distance between the trend window and the edge of the entire available space of the *Trend-Viewer* Control is predefined. The remaining space is therefore taken by the scales. How well readable the created diagram is to the user depends on the parameters selected for the scale inscription and the remaining scale elements.

2.2.3 Set the orientation of the diagram

The orientation of the diagram can be set to vertically or horizontally. The inscription of the scales is predefined and depends on the orientation of the entire diagram.

The alignment of the measured curves can be set from right to left, from left to right or from upward to downward direction by making additional scale settings.

A diagram can be represented rotated by 90° to reproduce the look of a mechanical curve plotter:



The curves of scales **Data 1** and **Data 2** are drawn in the left or the right strip whereas the curve for scale **Data 3** runs across the entire width of the trend window. The 'Follow-Me' button allows to stop the curves or make them run from top to bottom as new data is put in.

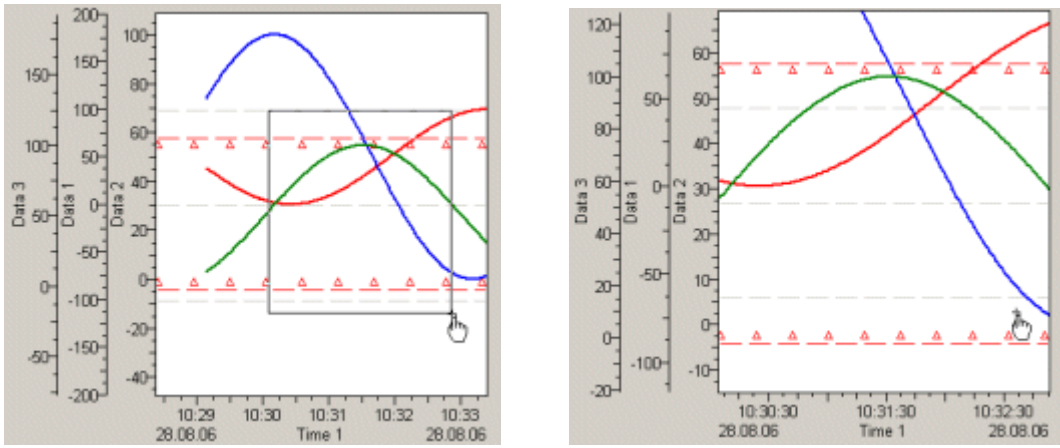
2.2.4Set zoom function

Section 2.3.1 describes how to select a cutout of measured curves. The zoom function can be deactivated to protect the set positions from inadvertent changes.

Alternatively, the lasso technique can also be used to add measured curves which are displayed in the trend window to a pane. In this zoom technique the time and data scales are also consistently updated.

The cutout selected with the lasso is released to the entire trend window. This allows to perceive more details of the measured curves. The zoom function can be used several times in succession. The zoom steps can, however, not be undone by an undo function. Instead, a range saved to the application before zooming must be reset.

The effect of the lasso zoom can be seen in the following illustration. The left figure shows the lasso selection, the right figure shows the result after releasing the mouse button.



2.2.5 Define auxiliary lines

Regardless of the measured curves auxiliary lines can be inserted to the trend window. These auxiliary lines divide the trend window into an adjustable number of equivalent ranges and such provide a rough reference point for reading measured curves. Number, starting point and appearance of the auxiliary lines are adjustable.

Additional auxiliary lines are assigned to scales and curves. Their parameterization is described in Section 3.2.2 and Section 4.3.

2.2.6 Divide trend window into strips

The trend window can be divided into several strips that are placed one on top of the other - or in plotter display in strips placed one beside the other. The data scales are therefore piled on top of each other and the assigned curves are displayed in the corresponding strips. When a curve comprises a range of values which is not displayed by the scale, the curve is truncated at the edge of the strip.

A combination of curves between the strips and curves which run across the entire trend window, is also possible. The figure in Section 2.2.3 shows a possible combination.

2.2.7 Tooltip

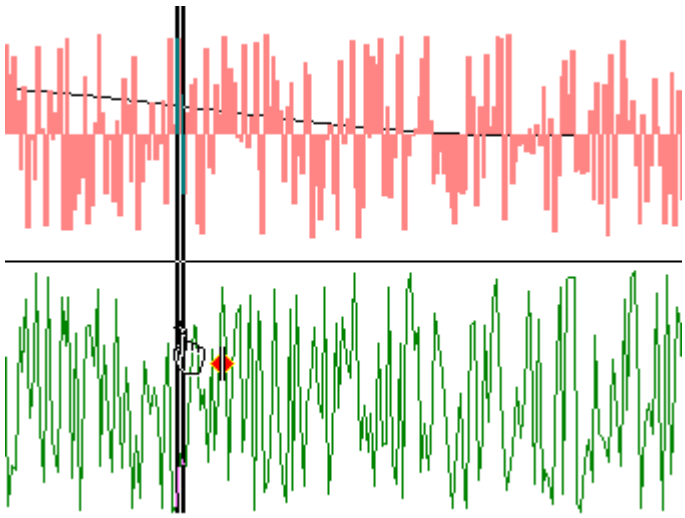
A tooltip is available for all scales and curve buttons. When sliding the mouse along a scale, the tooltip displays the current value at the current position of the mouse pointer. Thus, scale values can also be read off the main divisions line between inscriptions.

The name of the individual curve which is activated via a curve button as well as the name of the associated y-scale are displayed in a tooltip when the mouse pointer points to the curve button.

2.2.8 How to use the rulers and curve buttons

Two interactive rulers are provided for reading individual curve values. The vertical x-ruler always delivers the time value at the current position, as well as the corresponding measured data of all activated curves. The x-ruler affects all strips of the diagram.

The horizontal y-ruler delivers the scale values of all activated scales at its current position to the application. At the initial use, the rulers are contained at the edge of the trend window. The cursor shows the rulers when they are activated.



In addition to the rulers of the trend window a separate x- and y-ruler is available for each curve. A ruler can only be displayed via the interface which was designed for that purpose. Its use via mouse pointer is similar to that of Trend Window rulers.

The curve buttons are used to deactivate individual curves. They can be positioned either at the scales, to which the corresponding curves are assigned or they can be combined at the upper edge of the entire diagram for all active curves.

2.3 Interactive Use

TrendViewer provides high usability to the user through dynamically modifiable components. The components can be modified directly by the user to attain individually optimal readability and to gather different information from the same curves.

A prime feature of *TrendViewer* is the zoom function which allows to almost seamlessly extract detailed curve trends in very little time - e.g. in a few minutes - from series of measurements recorded in lengthy time intervals. This requires that the entire series of measurements has previously been loaded in *TrendViewer*.

Curves and scales also correspond during the zoom procedure. The zoom function can be applied to every scale as well as to the entire trend window. If only one scale is modified, all scales assigned to this scales are affected. If the trend window is modified, all curves and scales are affected by the zoom function.

For the analysis and documentation of measured curves additional properties of *TrendViewer* are most useful:

1) The dynamic arrangement of scales. It allows to resort diagrams in seconds and directly view connections between measured curves under new aspects (Professional License required)
2) The comparison of ranges of measured curves versus time
3) The comparison of trends of measured curves versus behavior, to be able to compare e.g. the setpoint curve and the actual value curve (Professional License required).
4) The modification of measured curves values (Professional License required)

All functions can be used via parameters of the application or interactively. Function 1 is described in Section 2.3.2, function 2 in Chapter 3, function 3 and 4 are described in Section 4.6.

2.3.1 Define a range of the measured curve

When defining a range of a measured curve by means of a data scale or time scale, no additional control elements like buttons, scroll bars or sliders are required. Instead, the scales are touched directly and moved to their range using the mouse buttons. Every modification of a scale setting affects the measured curves in the trend window such that the displays of scales and measured curves always correspond.

The use of the scales is described in Section 3.4.

In addition, the content of the trend window can be reduced, enlarged or shifted by using the mouse. The scales are automatically adapted, such that the displays of scales and measured curves are consistent again.

This function can be switched off such that the user only applies it when useful. This is to protect the entire range of the selected measured data from changes caused by accidentally moving the trend field.

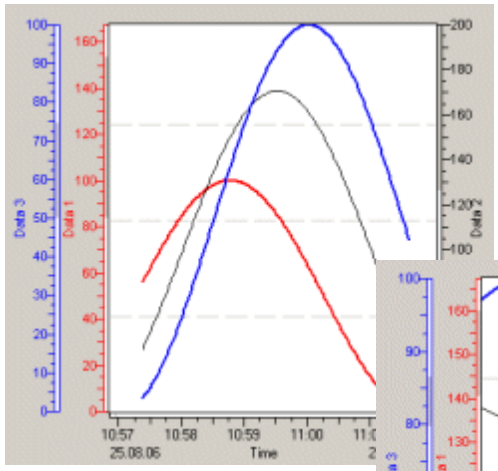
The echo of the cursor shows if the displayed range of the entire measured dataset is reduced, enlarged or moved. All possible operations are detailed in the following illustrations.

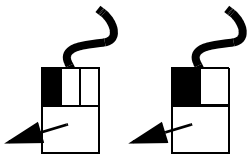
Reduce a range

Reducing a range produces an enlarged display of details of the measured curves. The range can be reduced until the smallest resolution of a scale is reached.

When the mouse is moved to the left while the left mouse button is simultaneously held down, the left lower point of the trend field is moved. The right upper point of the trend field remains unchanged.

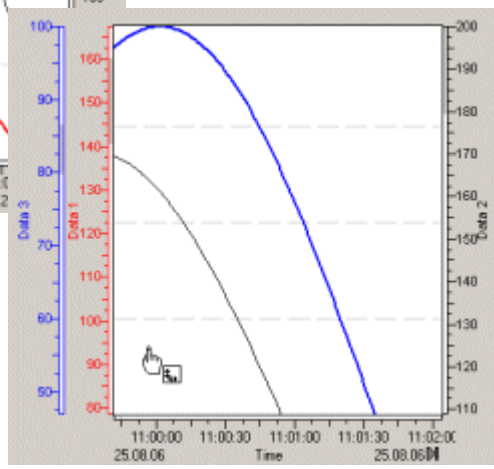
before moving

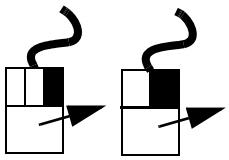




after shifting

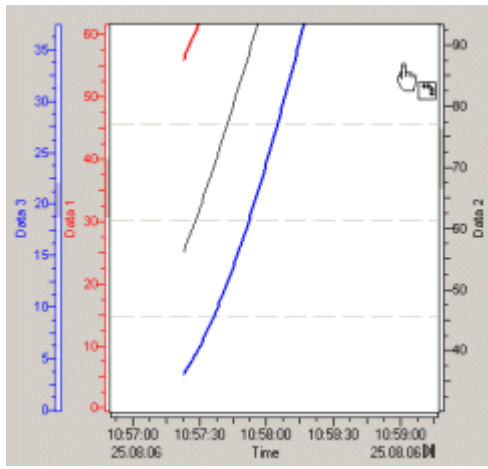
to left bottom





after shifting

to right top



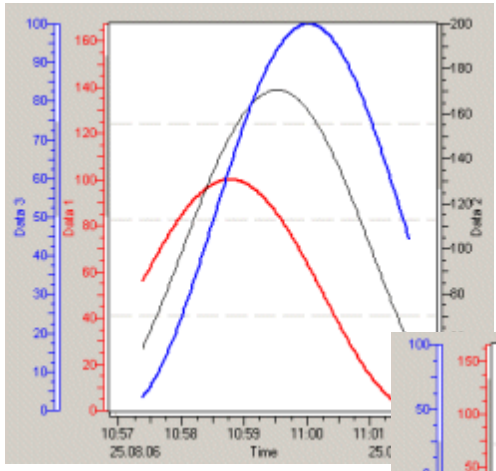
When moving the mouse to the right while simultaneously holding down the right mouse key, the right upper point of the trend field is moved. The left lower point of the trend field remains unchanged.

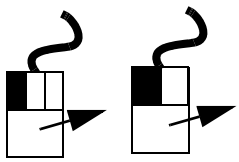
Enlarging a range

Enlarging a range results in a reduced display of details of the measured curves. The range can be enlarged until the lower or the upper limit of a scale is reached.

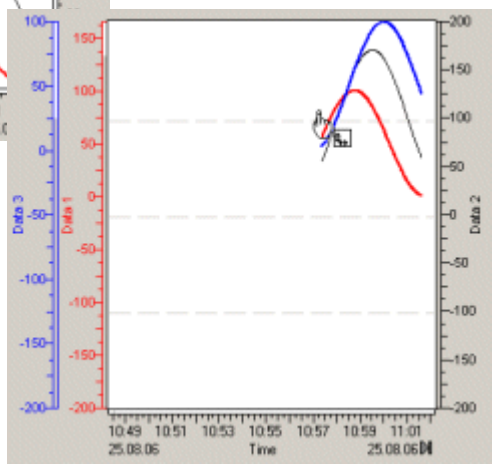
When the mouse is moved to the right while simultaneously holding the left mouse button down, the left point of the bottom trend field is moved. The right point of the top trend field is moved. The right top point of the trend field is unchanged.

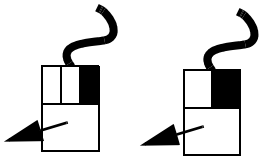
before moving



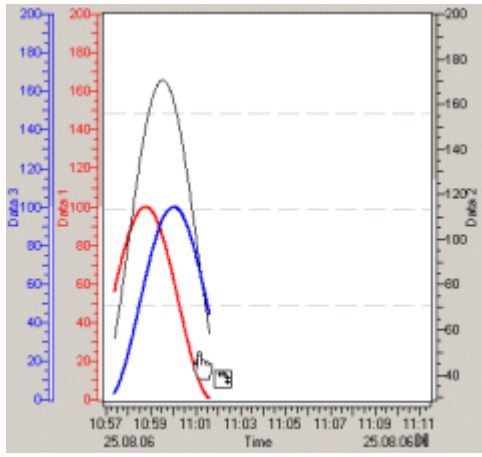


after moving to top right





after moving to bottom left



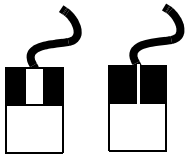
When the mouse is moved to the left while simultaneously holding the right mouse key down, the top right point of the trend field is moved. The left bottom point of the trend field remains unchanged.

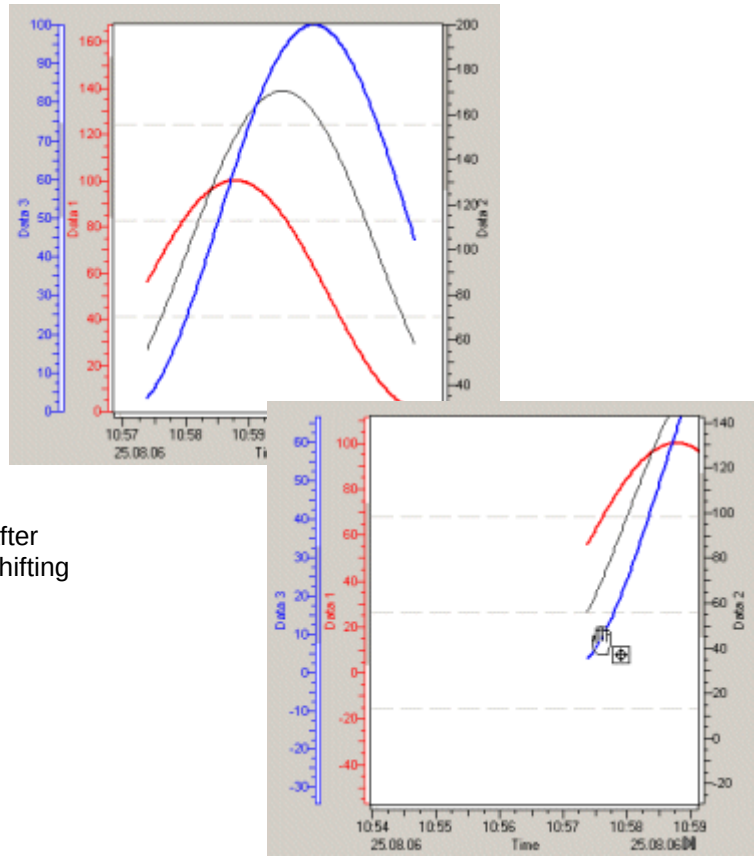
Shifting a range

When a range is shifted, another part of all measured curves is displayed without changing the sizue of the details.

The left and the right mouse keys are pressed simultaneously and the mouse is moved to the right or to the left.

before shifting





after shifting

2.3.2 Position scales (Professional License required)

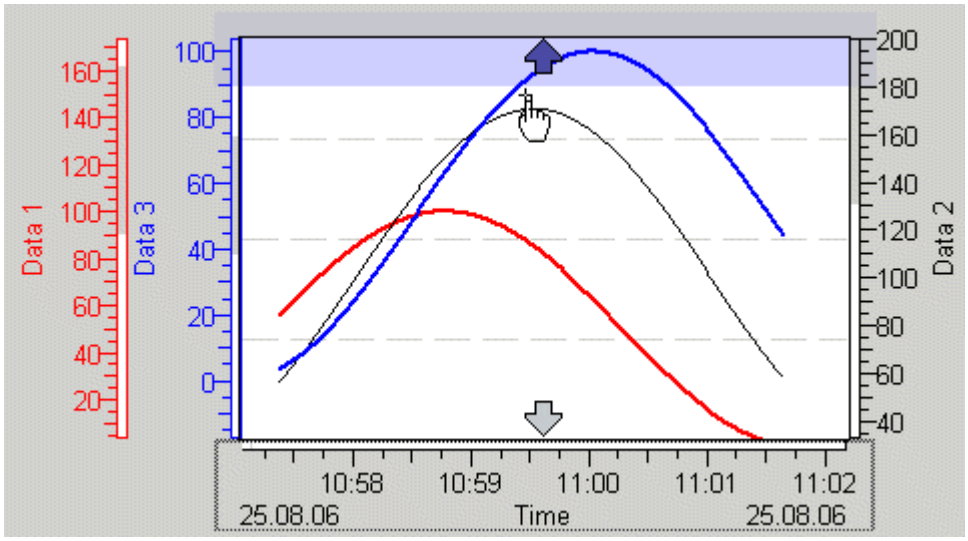
The trend control provides two x-scales and up to 512 y-scales. The x-scales are arranged on opposite sides of the trend window and can be configured as time scales, linear or logarithmic data scales or degree scales.

The y-scales are linear or logarithmic data scales or degree scales. They can be arranged in columns next to each other or also within a column on top of each other. Arbitrary combinations of positions are possible.

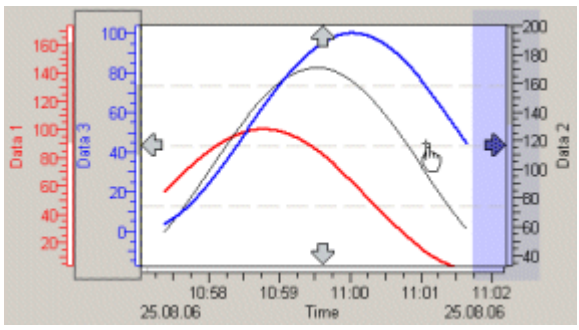
To create a x/y-chart, one of the y-scales can also be arranged parallel to an x-scale. This allows e.g. to plot the curves of a measuring point versus other measuring points (Example: pressure/ temperature dependency).

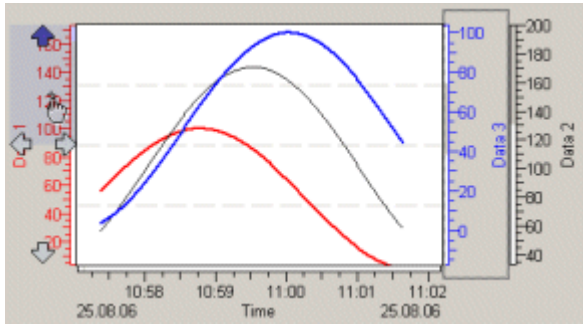
The scales are touched at the scale button or as object and moved to the desired position. *Trend-Viewer* supports this task by echos.

If the x-scale is to be positioned at the opposite side of the trend window, the echo shows the two possible positions:

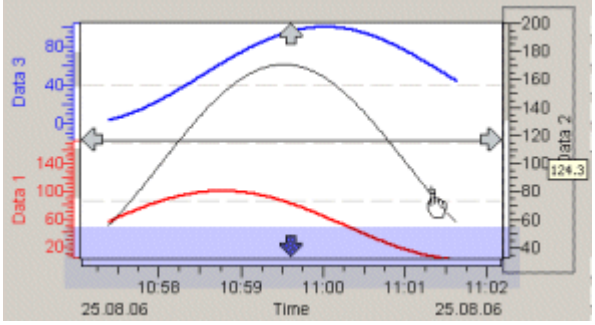


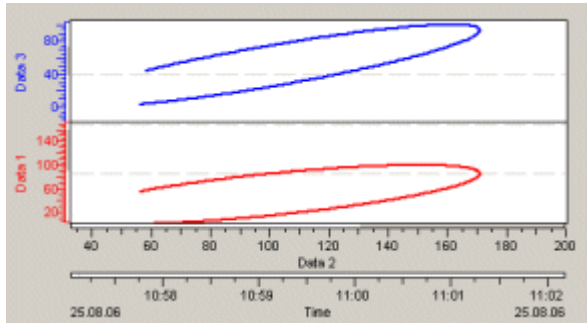
If a data scale is to be stacked on top of another one or if it is to me moved to another position, the echo shows all possible positions which can be attained using the mouse. The dark blue arrow indicates the position the scale takes when the mouse button is released.





In a phase diagram the data scale can also be arranged parallel to the time scale. The illustration on the right shows the possible result after positioning data scale **Data 2.** If the curves are to be plotted versus **Data 1** instead of versus scale **Data 2**, only scale **Data 1** must be moved to position of scale **Data 2**. **Data 1** and **Data 2** are such exchanged.

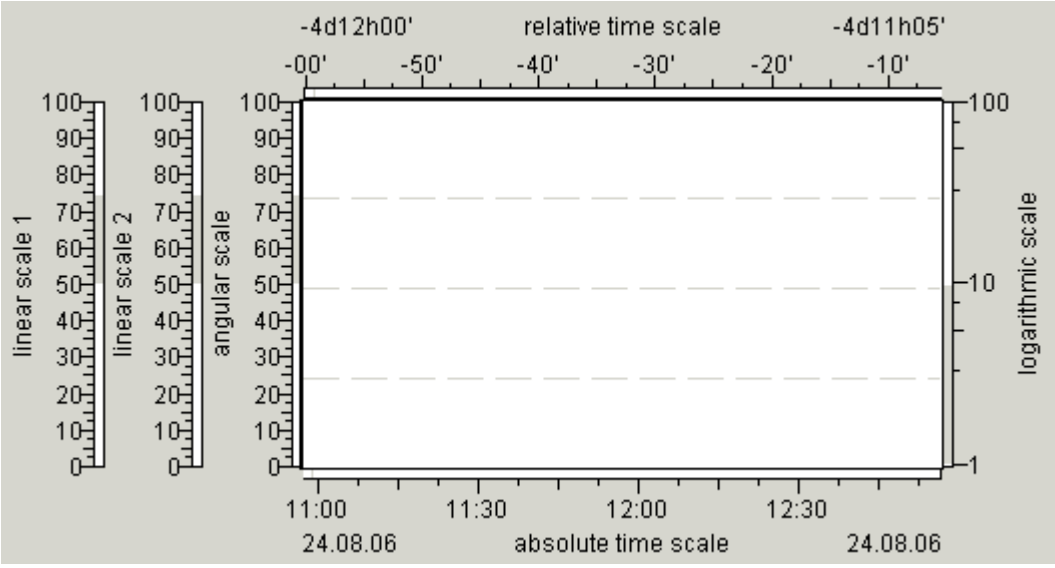




3. Scale

The most important component of a curve diagram for accurate reading of measured values is the scale, which can be adapted to any requirement. This applies above all if the singular zoom features of *TrendViewer* are used to seamlessly display a very wide spectrum of resolutions - e.g. for time scales from years to milliseconds - while zooming. In *TrendViewer* 5 scale types are available which can be positioned individually around the trend window.

The following illustration shows a layout example of all scale types:



2 x-axes are provided such that a measured curve can e.g. be displayed simultaneously in two different time intervals in a diagram. Both scales can be parameterized independently and changed interactively.

In phase diagrams a data scale is displayed parallel to a time scale, so that the ratio of several measured values to a special measured value can be monitored with respect to time-dependent behavior.

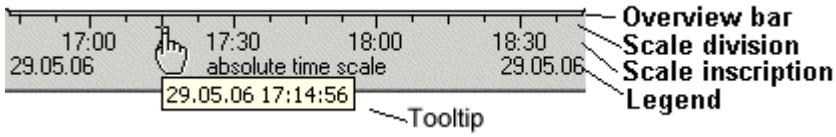
An separate data scale can be assigned to every measured curve, several measured curves can be referenced to one data scale. Each curve is connected with one of the two time scales.

The display of the measured curve always corresponds to the current scaling. If a scale or the size of the cutout displayed in the trend window is interactively modified, the measured curves are automatically adapted.

3.1 Scale components

A scale consists of 2 keys and 5 components which can be formatted individually:

- an overview bar in which the displayed section is highlighted in position and length in relation to the entire range of values of a measuring curve.
- a scale division with main grid, secondary grid and division grid. The main grid is labelled with data, the number of the secondary grid and division grids can be adjusted.
- the scale inscription, which is aligned in the main grid depending on the scale type. The step width of the main grid is defined in file TVformat.xml in proportion to the entire displayed interval. Well readable inscriptions are automatically provided. If the provided space is insufficient, the inscriptions are hidden.
- the legend, which consists of text and in time scales of the minimum and the maximum value of the scale range. When the minimum- and maximum value are displayed - e.g. date, hour and minute - the inscription of the main grid line should be minimized as much as possible using only smallest time intervals - e.g. seconds and milliseconds.
- the Tooltip which displays the value at the cursor position when pointing to the scale with the cursor.



In order to save place, the overview bar and the legend can be hidden. The legend of a diagram can be displayed by the application at any other place independently of *TrendViewer*.

The tooltip can be deactivated to improve the performance in systems with little processor performance.

3.1.1 Overview bar

The selectable overview bar of a scale roughly shows the ratio between the range of values of the assigned measured curves and the displayed range. In addition, the interval between the minimum and maximum value of the measured curves is always inserted.

3.1.2 Scale inscription

Besides the accuracy and the text format of a scale inscription, which depend on the scale type, versatile parameters like the font, alignment at the scale, font style and the scale edge adjustment can be set. When formatting the scale inscription sufficient space must be provided to ensure the complete display of inscriptions as well as optimum readability.

In conflicts between the available space and the optimal readability, each requirement must be decided by priority. Conflicts arise e.g. with directly contiguous scales (so-called stacked scales of values) or when the trend display is contained in a window of interactively changeable size. *TrendViewer* always tries to ensure the best possible representation when the window size is changed.

3.2 Scale types

There is a principal difference between data scales and time scales. Data scales are used in time-value representations (x,t-diagrams) of analog and binary signals for the x-axis of a diagram. The locus curves (x,y-diagrams) or the phase diagrams (x,y,t-diagrams) in which the data scales are also used as x-axes form an exception.

3.2.1 Data scale

A data scale can be assigned a linear, logarithmic or an angle division. The changeover between linear and logarithmic division lines is also possible in the application during the use of the scale provided that this is permitted by the scale initial values and the scale final values. The angle division can be displayed cyclically in 360 degrees or as linear representation.

In order to ensure optimal readability in every situation, the format of the scale inscription for the main grid can be defined in file TVformat.xml. A section each is contained on linear and angular division, in which the format of the inscription for the main grid, legend and tooltip are defined for each section of the measured curves which is represented by a data scale.

The following example shows an extract of file TVformat.xml:

```
<DecimalFormat>
<RangeValue>20.2</RangeValue>
<MainGrid>5</MainGrid>
<ScaleFormat>%.lf</ScaleFormat>
<LegendFormat></LegendFormat>
<TooltipFormat>%.1f</TooltipFormat>
</DecimalFormat>

<DecimalFormat>
<RangeValue>10.1</RangeValue>
<MainGrid>2</MainGrid>
<ScaleFormat>%.lf</ScaleFormat>
<LegendFormat></LegendFormat>
<TooltipFormat>%.1f</TooltipFormat>
</DecimalFormat>
```

<RangeValue> specifies the inscribed section of the range of values which is represented in the diagram.

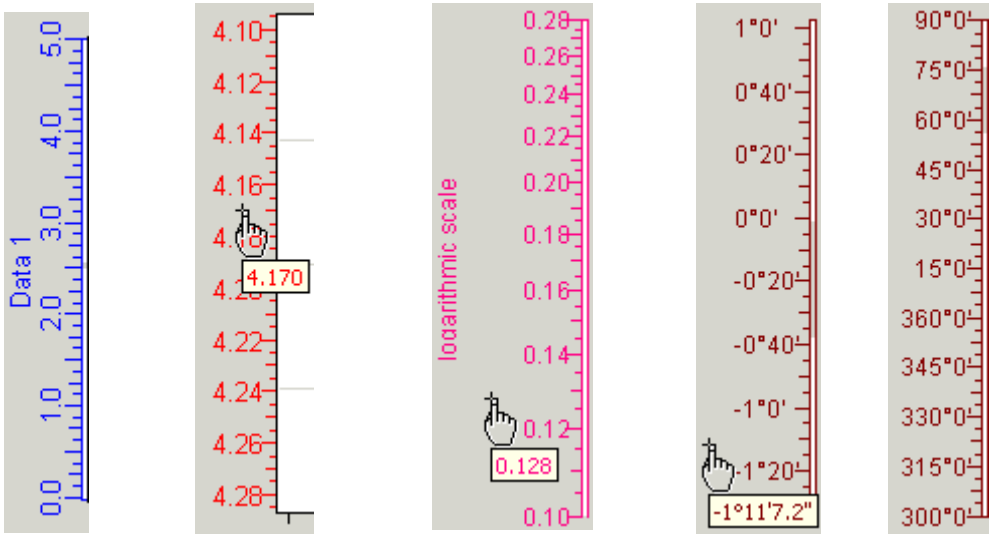
<MainGrid> specifies the range of values which corresponds to a main grid of the scale. This range of values is assigned to the <MainGrid> starting at <RangeValue> and is valid up to the next higher <RangeValue>. In the example, <MainGrid> =2 is set ranging from <RangeValue> > 10.1 to <RangeValue> <= 20.2.

The <RangeValue> was always entered 1% higher than the requested maximum value, to compensate representation inaccuracy on the monitor, especially when interactively setting the initial scale values or final scale values.

<ScaleFormat>, <LegendFormat> and <TooltipFormat> specify the format of the scale inscription, the legend and the tooltip (C-Syntax of the format statement). The <TooltipFormat> is always stated one digit more accurately than <ScaleFormat> so that a more precise value can be represented when selecting it with the cursor.

A TVformat.xml file which corresponds to the standard setting is delivered with *TrendViewer*. The file can also be modified according to the requirements of the application.

Examples of data scales:



3.2.2 Time scale

Time intervals (e.g. measured data at batch start) as well as absolute times with date and time can be displayed.

In time scales utmost precision in the inscription of the scale sections are a main goal. As the inscription of the main grid with date (day, month, year) and time (hour, minute, second, millisecond) becomes very long, it can be extended to the legend. The legend contains the minimum and maximum time value of the displayed interval.

The accuracy of the inscription can be adapted to the size of the respectively displayed interval. If the scale only shows e.g. few seconds of it, the scale division can be adjusted to milli-seconds. If years are displayed, the scale marks can be adjusted e.g. to 90 days.

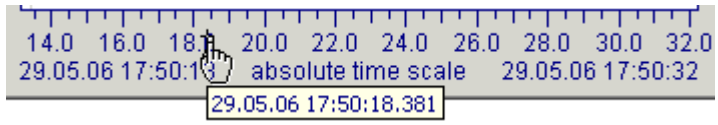
The country-specific date format is specified in the TVformat.xml file. The order of the components of the date and separators can even be defined separately for tooltip and scale inscription.

The change to daylight saving time is considered.

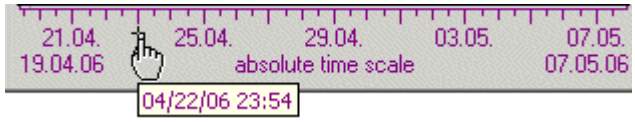
Examples of time scales

Display of minutes

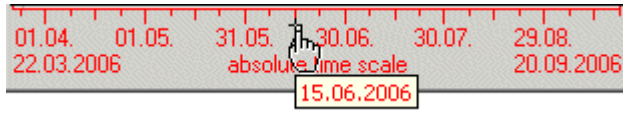
Display of seconds



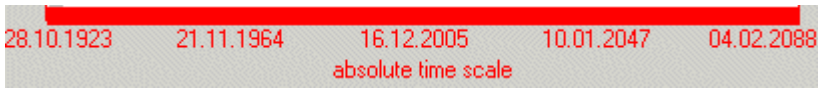
Tooltip with English data format



6 months with 30 day grid



more than 100 years



relative time scale

3.3 Scale properties

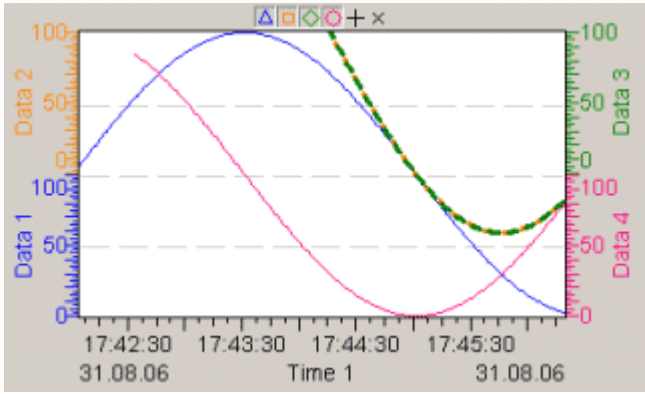
Apart from the graphic attributes like color and inscription the scales may also contain additional components like the 'Follow-Me' button for updating an online recorded measured curve and scale buttons to show and hide the associated measured curves and information on the curve names.

The alignment of the inscription is selectable for all scale types, such that e.g. online recorded measured curves can be plotted from left to right or from right to left.

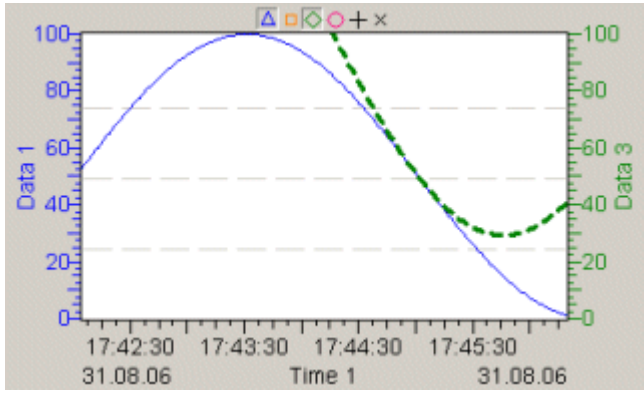
3.3.1 Scale display and positions

The position can be changed interactively as described in Section 2.3.2 or can be defined by parameters. The x-scales are to be positioned at the top and bottom of a diagram whereas both x-scales can be placed on top of each other on one side of the diagram.

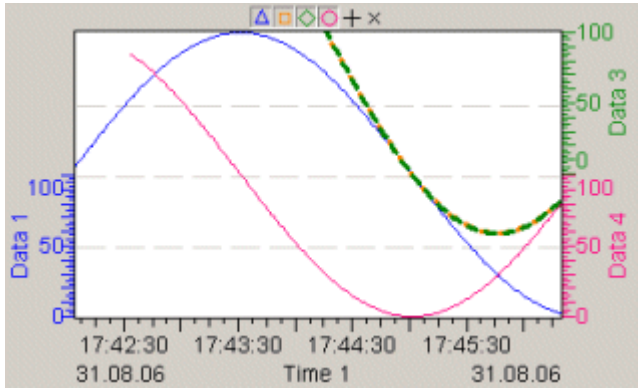
The y-scales are displayed left or right beside the diagram, whereas several y-scales can be placed beside each other or on top of each other.



If the automatic scales management is activated, only scales with assigned curves are displayed. This manner only a diagram with minimum information is received if e.g. curves are selected and deselected via the curve buttons. Curves 2 and 4 are deselected in the display:



The display of a scale is additionally controlled via the attributes 'activated' and 'visible'. If one of the scales is deactivated in scales, which are stacked one on top of the other, the adjacent scale expands to the range of the deselected scale. If a scale is activated but not visible, the space of the scale which is set to invisible remains empty. For this purpose, the automatic scale management must be deactivated.



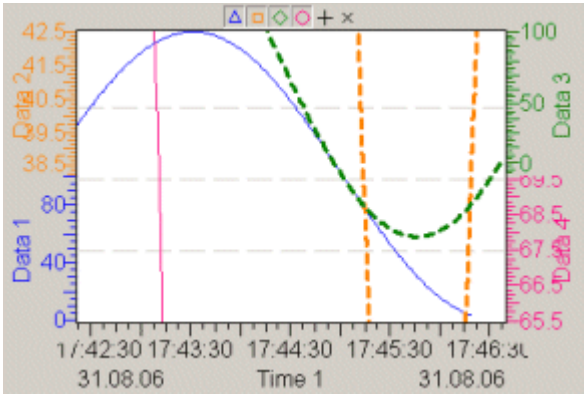
3.3.2 Graphic attributes

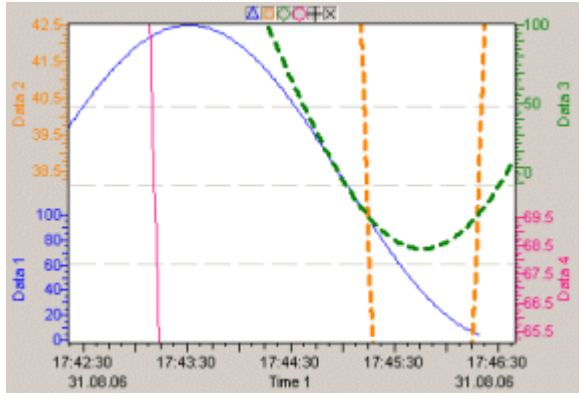
The assignment between scale and curve can be visualized by means of color. Colors can be set arbitrarily. However, sufficient contrast must be provided to the image background to ensure readability.

High demands are made on the inscription of a scale because of the amount of information which is to be displayed. Above all, if the inscription should be readable any time during the zoom procedure and if only limited space is available, special attention must be paid to good readability of the complete units of information. *TrendViewer* attempts to automatically ensure this as much as possible.

The available range for the inscription is defined by the size of the diagram, the field, which is required for the trend window, the font, the distance between the main grids of a scale and the font alignment. If the available space is initially insufficient, more room can be provided by reducing the font, enlarging the distance between scale and legend (in horizontal inscription) and margins at the scale border.

With directly bordering scales like e.g. scales that are piled one above the other the representation of the inscriptions can be optimized by inserting invisible scales and by assigning priorities. The illustration shows a diagram before and after optimizing space.





If the space between the main grids or the edge of the scale is insufficient, the inscriptions are automatically omitted. This can be adjusted by respectively formatting the inscription in file TVformat.xml.

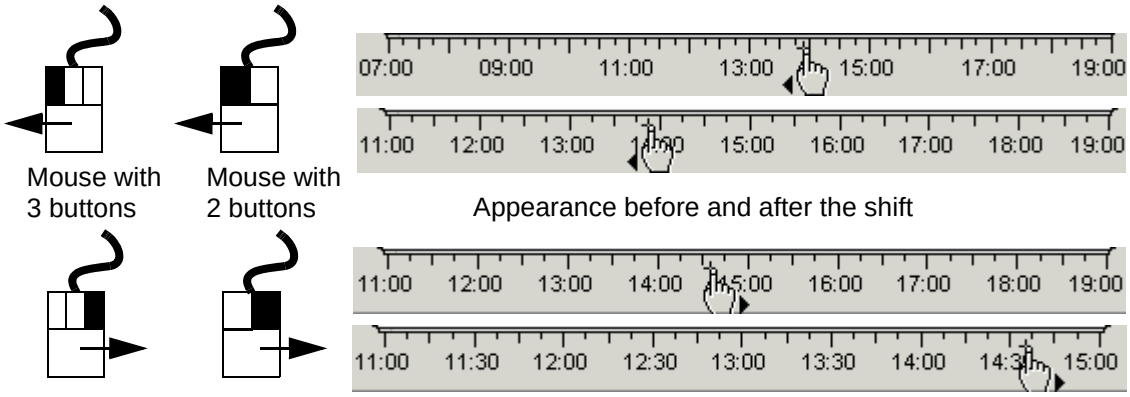
3.4 Modify scale range interactively

The scale section can be modified directly using a mouse or a comparable positioning device.

The echo of the cursor shows if the represented section of the total range of values of the measured curve is reduced, enlarged or moved. *TrendViewer* always ensures that all curves which are assigned to a scale in the trend window are changed proportionally with the scale.

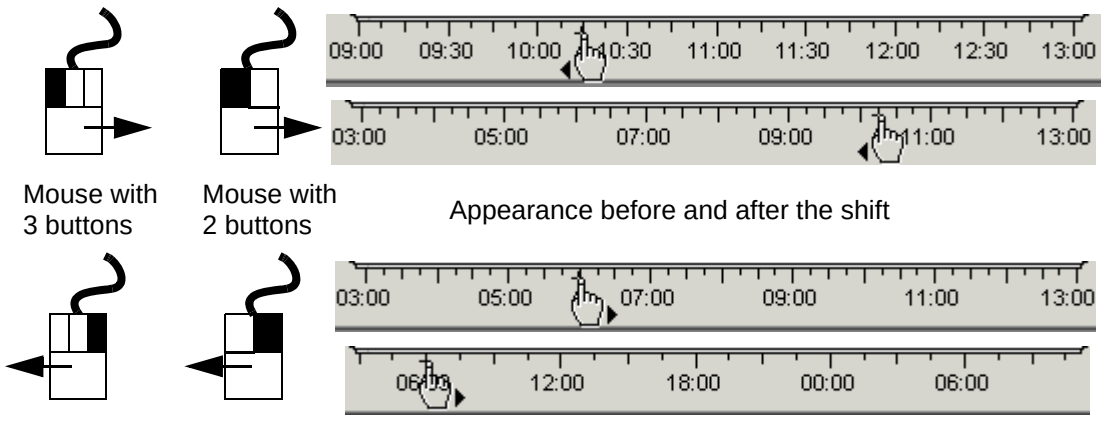
The left side of an x-scale or the bottom edge of a y-scale is changed by using the left mouse button. The right edge of an x-scale or the top edge of a y-scale is changed by using the right mouse button. Each opposite edge remains unchanged.

Reducing a range (Zoom In)



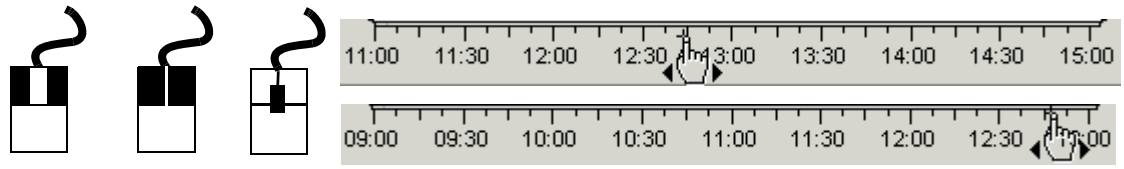
Reducing a range causes an enlarged display of details from the total range of values of the measured curves. The cutout can be reduced until the selected minimum resolution of the scale display is achieved.

Enlarging a range (Zoom Out)



Enlarging a section results in the reduced representation of details of the total range of values of the measured curves. The range can be enlarged until the maximum upper limit or the minimum lower limit of the scale display is reached.

Shifting a range (Panning):



Mouse with 3 buttons | Mouse 2 buttons | Mouse with wheel

Appearance before and after moving

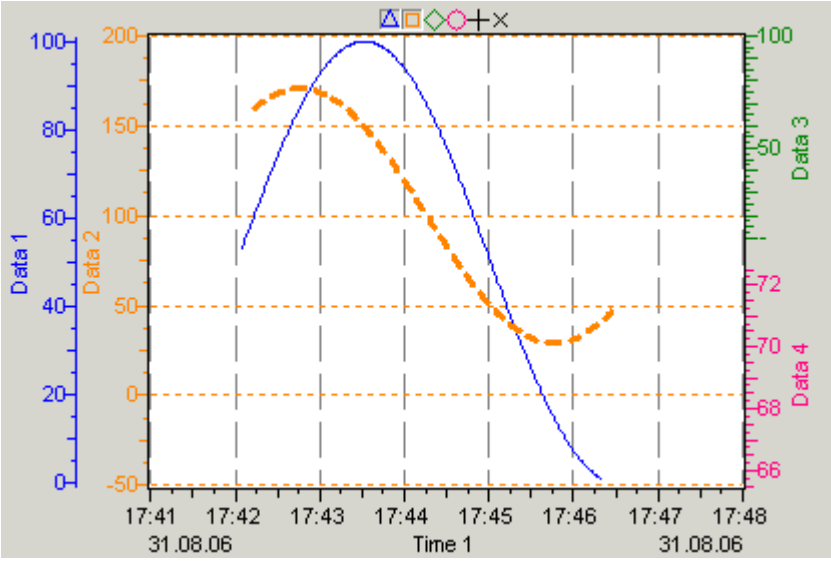
Shifting a range results in the representation of another part of the entire range of values of the measured curves, without changing the details. The range can be moved until the limits of the scale representation are reached.

The left and right mouse button are simultaneously pressed and the mouse is moved to the right or to the left. By an appropriate configuration of the scale (see **MoveOnly**) the cutout can also be shifted with any arbitrary mouse button.

In order to prevent an unintentional shifting of the scales, the control concept can be switched. To change the range, the control or shift key must be held down while moving the mouse.

3.5 View auxiliary lines

In order to improve the readability of the curves, auxiliary lines can be viewed for each scale. This property is particularly helpful if the two opposite x-scales are to be set to coincide by means of the zoom function. The comparison of curve runs at different times is very well supported by this feature.



The distance between the auxiliary lines and the number of auxiliary lines can be set. In all scales an arbitrary number (e.g. every hour) of auxiliary lines is selectable.

4. Curve

TrendViewer provides 2 curve types with different features: binary and analog curves. Attributes which influence the display e.g. in line style, color, stroke width and interpolation between interpolation points are assigned to the curves. In addition, the surface between a curve and a limit line or between two curves can be flooded. The representation of the flooded area can also be adjusted. Curves are assigned to scales to form groups which show the connection between measured curves.

TrendViewer buffers all curve values which are returned for representation. If e.g. measured data is continuously recorded and displayed in a machine control, the buffer continuously grows and affects the performance of the machine control after lengthy runtime.

The length of the curve buffer can therefore be set for each curve. The buffer is then described cyclically.

For the display of phase diagrams it is required that the same x-values are applied to all curves, such that the y-values of the measured curves can be associated. This only works on the assumption that the application ensures the measured curves to be returned to *TrendViewer* at the same grid point of the x-values - which are in most cases time values of the curves.

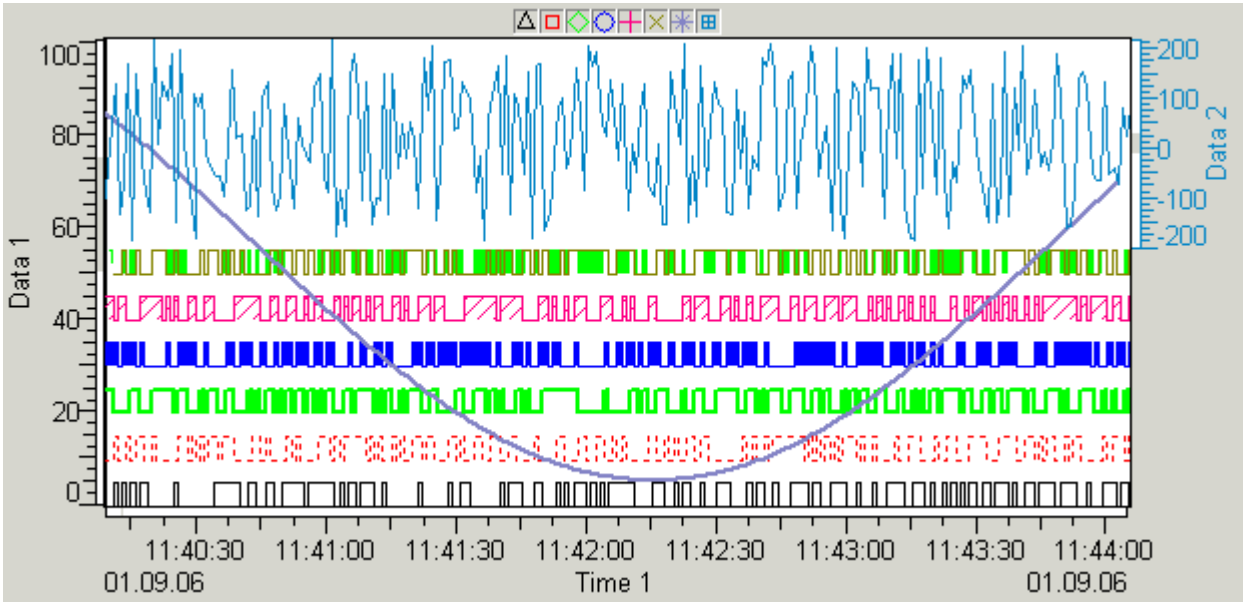
4.1 Curve types

Analog curves show trend data of analog measuring points. Every measured value is represented by a grid point of the curve. The range of values and the resolution of a measured value are predefined by the measuring point. The curve shape is displayed in accordance with the assigned scale and can such be evaluated.

Interpolation points of binary curves can only accept the values 0 and 1. Curve values < 0 are set to 0, values > 1 are set to 1.

Binary curves are placed into the trend window and are independent of analog curves. Only the allocation of a common x-axis establishes a connection between binary and analog curves.

Several analog and binary curves are combined in the following illustration.

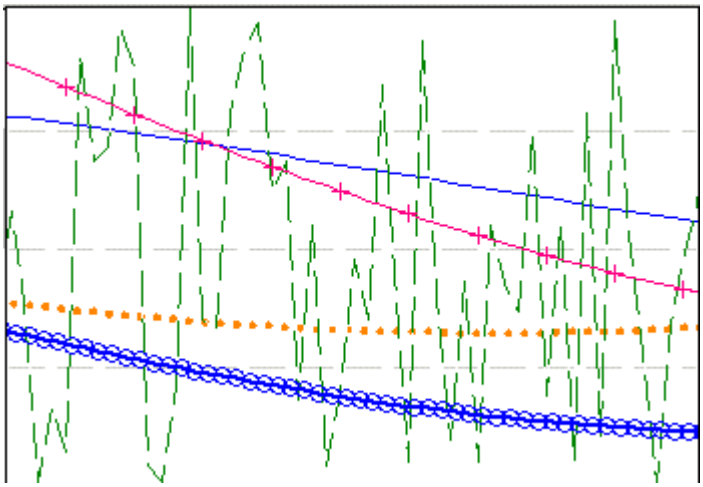


4.2 Display

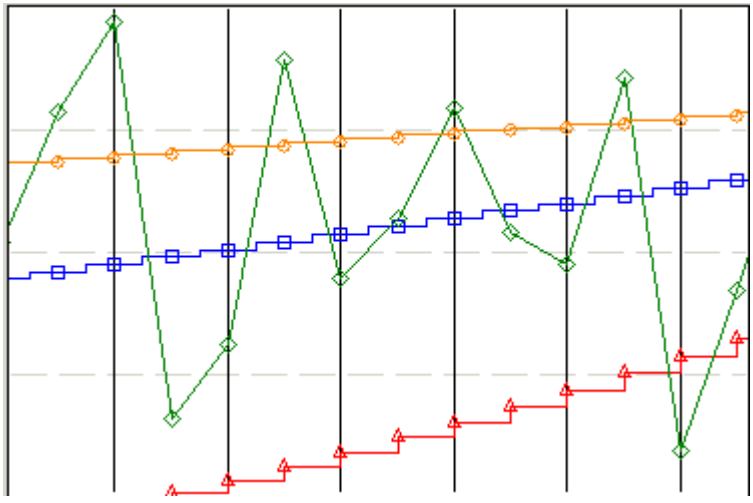
Different graphic representations of curves are used to differentiate curves in a diagram or to visually combine them. If diagrams are output on a printer that cannot map color information, good readability must be ensured by different symbols and fill styles.

4.2.1 Elementary properties

The following illustration contains all the attributes a curve may contain. How many vertices of a curve should be assigned a symbol (each one, each second one etc.) can be defined.



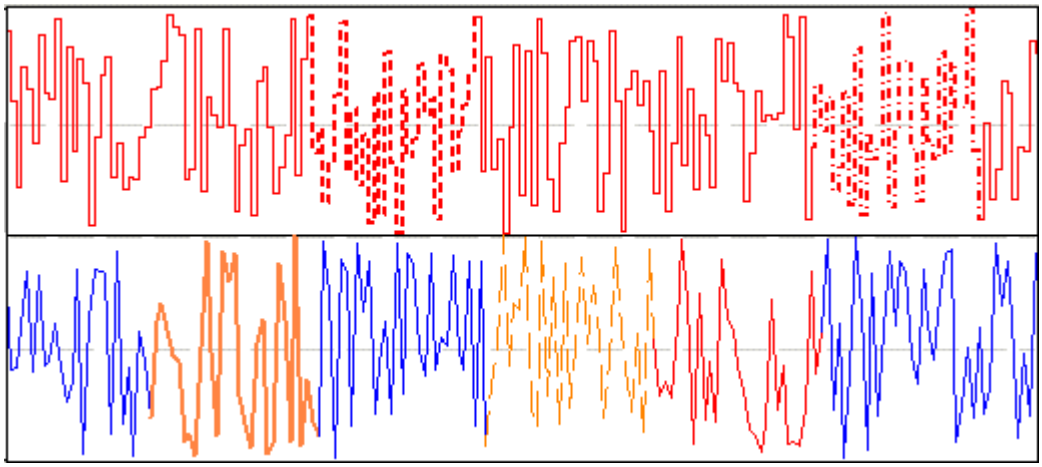
The interpolation between 2 vertices can take 4 shapes, which are displayed in the following display. In staircase curves the step can be at the left, the right or between two vertices. Each vertex can be marked with a symbol, to visualize the differences.



4.2.2 Status

Each vertex of a curve is assigned a status value, which is differentiated by a corresponding graphic display of the connecting line with color, stroke width or line style.

It is therefore possible to differentiate e.g. between correct measured values, substitute values and faulty values.

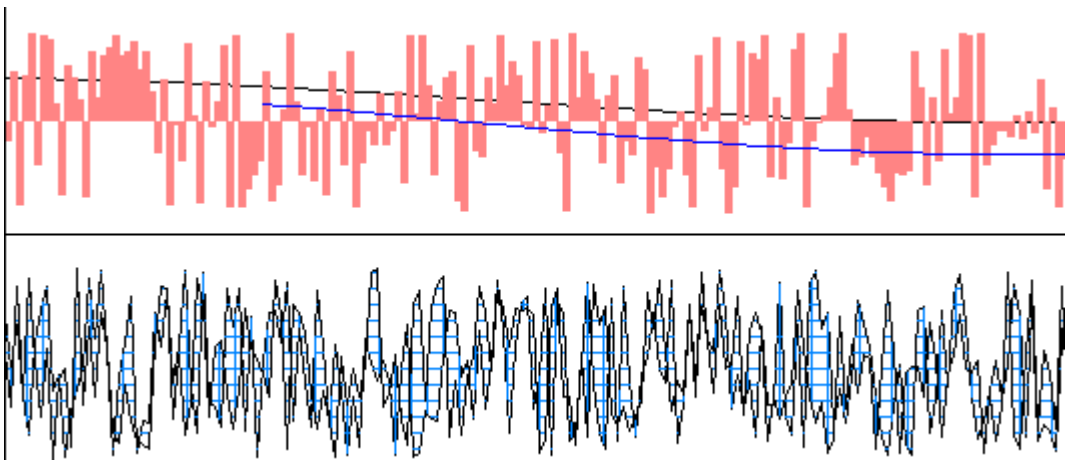


Status values can be assigned to analog and to binary curves.

4.2.3 Flooding

The flood function can be used in complex diagrams in order to better visualize the deviation between two measured curves or the distance between a measured curve and a limit value. The flooded section can be assigned a separate color and fill style to improve the distinction between the remaining graphic elements in the diagram.

In connection with flooded surfaces the definition of drawing priorities is required such that measured curves are not covered by flooded surfaces. The following figure shows an example in the upper part in which the blue curve is placed in front and the black curve behind the flooded area.

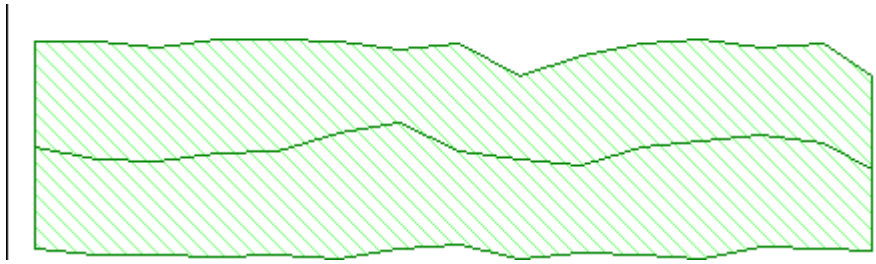


Flooding is only possible between analog and binary curves.

The curves can be assigned to the same or to different x- and y-scales.

4.2.4 Envelope curve

An envelope display can be used in special task descriptions that do not require a detailed representation of the measured curve but in which the definition of a strip in which the measured curve is contained is sufficient. In addition to the envelope curve a mean value line can be inserted.

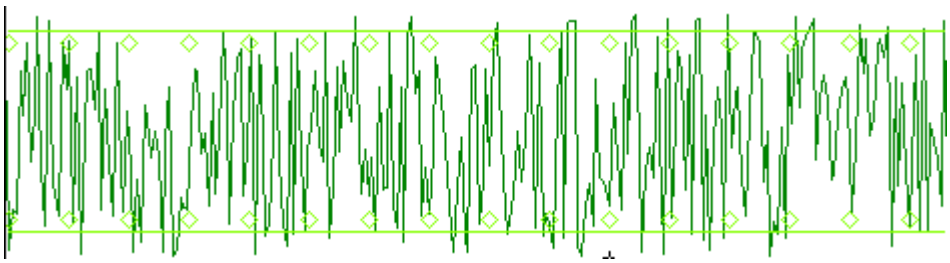


Envelope curves can only be set in analog curves.

4.3 Limit lines

Another utility for the evaluation of measured curves are limit lines which can be activated as maximum and minimum expected value for each measured curve. Values that exceed or fall below limit values can such easily be identified.

To differentiate limit lines of several curves, they are marked by the symbol associated to the curve. Symbols are drawn under the upper limit line or on top of the lower limit lines. The line attributes can be set individually.



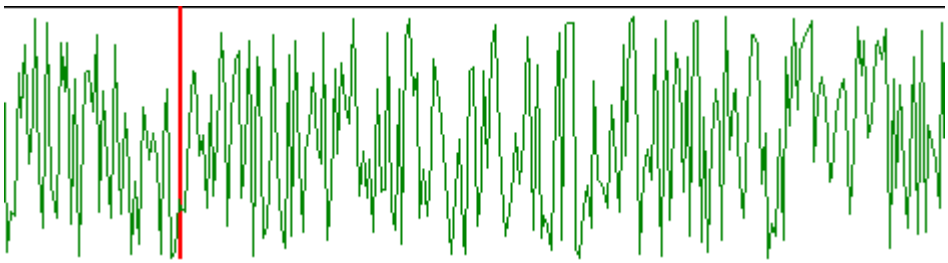
Limit lines can be shifted using the mouse.

Binary curves do not have auxiliary lines.

4.4 Rulers

Each curve is assigned a horizontal and a vertical ruler. The rulers are pulled from the edge of the trend window into the diagram using the mouse. Only the horizontal or the vertical ruler can be seen or operated. The ruler of the diagram (see section 2.2.8) must be deactivated.

The line attributes of the rules can be set individually to be able to differentiate more rulers.



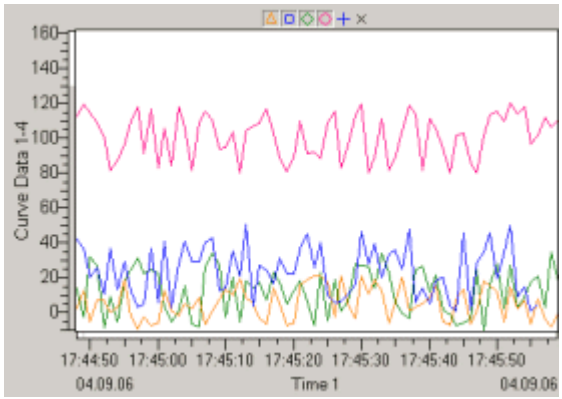
If only the vertical ruler of the mouse is moved, the application continuously gets the X-coordinate and the value of the measured curve at the ruler position. The application can represent these values to the user.

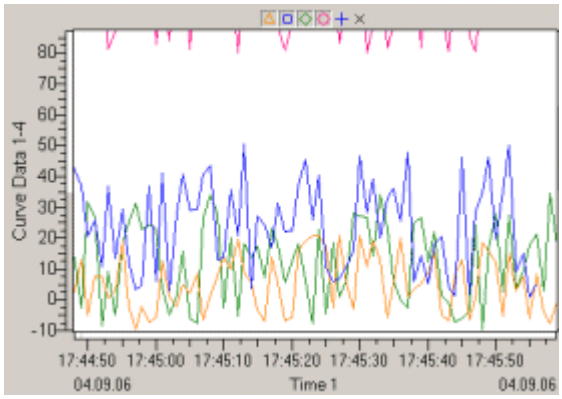
Binary curves only have a vertical ruler to receive the binary value at a certain point of time.

4.5 Scale assignment

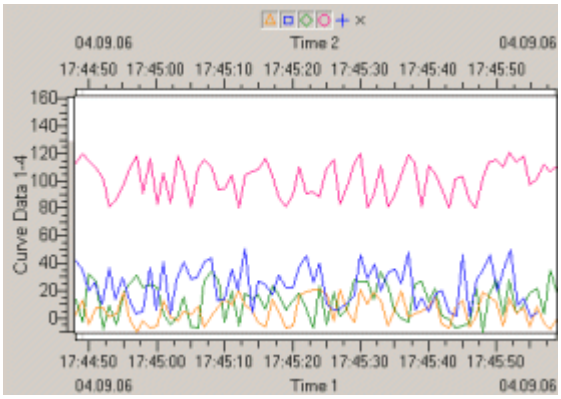
Each curve must be assigned to one of the two x-scales and to one of the 512 y-scales.

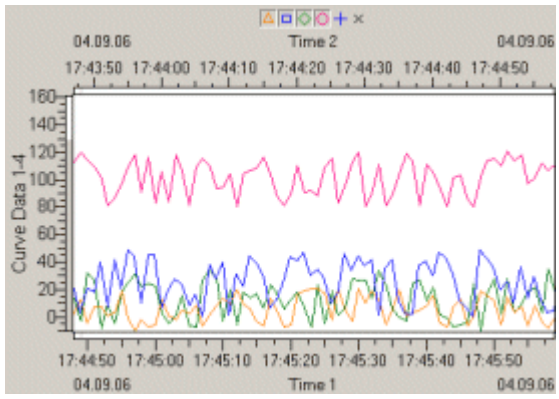
Several curves can be assigned to the same scale. Thus the comparability of the run of the curves of several measuring points is supported. The zoom function simultaneously affects all curves which are assigned to a scale. In the following figure the scale end of Curve Data 1-4 is reduced from 160 to 80.





In the following figure the blue curve was assigned to the second x-scale and moved by one minute.



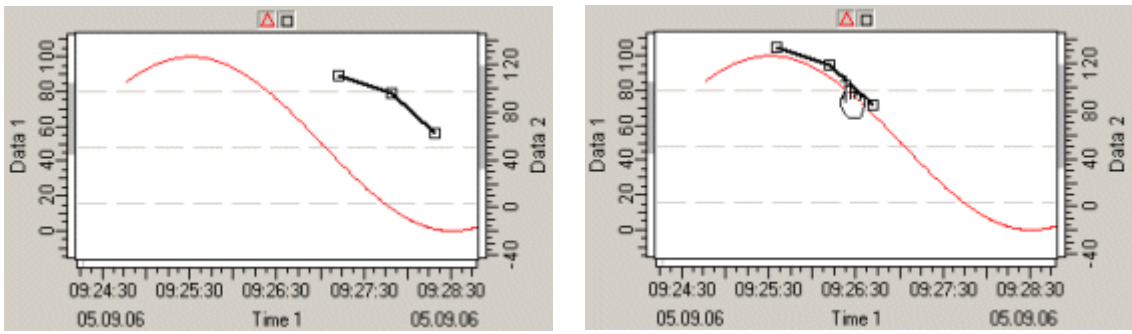


4.6 Edit curves (Professional License required)

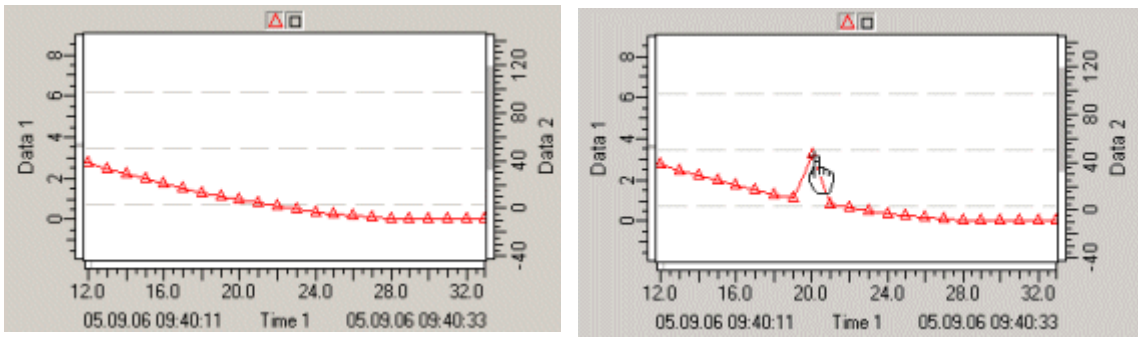
A task definition in the analysis of archived measured curves or in the monitoring of online trends is the comparison between the setpoint curve and the actual value curve. Setpoint curves are often computed for certain situations in process technical processes and predefine destination points or runs of curves for the maintenance personnel.

If curves of setpoint values are to be adapted to current given facts, they must be moved within the diagram and their vertices must be modified. Each analog curve can be enabled for interactive editing.

Note: All interactive changes of the curves also change the curve values which were loaded in *TrendViewer*. This must be kept in mind if the curve values are to be read from the data buffer of *TrendViewer*. In the following figure the black curve of setpoint values is moved.



The vertices of the red curve are changed by means of the mouse.



An application can move or stretch an analog curve along x- and y-direction using the attributes **Offset** and **Gain**. This modification does not affect the curve values in the data buffer of *TrendViewer* and can be reset any time.

5. TrendViewer Objects

5.1 Properties

The following tabular representation contains an overview of all objects of the trend control and their properties:

Object groups:

5.1.1 TrendCtrl

General properties of the trend control:

AutoScaleGrid	automatic scaling of the scale grid (TRUE)
AutoScaleVisibility	TRUE controls the automatic activation and deactivation of the scales
BackColor	background color
CrossHair	activate (TRUE) or deaktivate (FALSE) the crosshair
Curve	returns the associated curve object (TVCurve)
Curves	returns the number of the curves
CurveButtons	style for curve buttons
DataWindows	activates (TRUE) or deactivates (FALSE) the display of the data window when selecting the scales and curve buttons
DockingScales	activates (TRUE) or deactivates (FALSE) the interactive docking of scales
FixedFont	deactivates the automatic adaptation of the font size (TRUE)
Font	font used (change of the font style leads to new computation of the font size **(FontScaleFactor)**
FontScaleFactor	scaling factor for the font size resulting from the optimal display size computed by the trend control
ForeColor	foreground color of the trend control
HorizontalScText	generates only horizontal text for vertical scales in the event of TRUE
HorizontalScTextAlign	alignment of the horizontal scale text
Interactive	activates (TRUE) or deactivates (FALSE) the mouse control
LockUpdate	„freezes“ (for TRUE) the current look of the control
MarginFactor	sets the scaling factor for the margins of the trend control
Orientation	orientation of the trend window at the scales or the scales at the scale window
OVBarsBackColor	background color of the overview bars
OVBarsRangeColor	background color of the range of the overview bars
OverviewBars	shows (TRUE) or hides (FALSE) the overview bars
Picture	„read only“ property which returns a bitmap of the current control
ReleaseKey	release key to position the scales via the mouse application
Ruler	reference to a TVRuler object (ruler)
Rotated	orientation of the trend chart (FALSE: horizontal, TRUE: vertical)
ScaleLegends	shows (TRUE) or hides (FALSE) the scale legends
ScaleWidthFactor	enlargement factor for the scale width to enable the display of horizontal scales inscriptions in the text (see **HorizontalScText**)
StripChartsStyle	activates and configures the strip chart functionality
TrendAuxLineCount	number of the trend auxiliary lines of the trend window

TrendAuxLineStart	Starting value for the trend auxiliary lines of the trend window
TrendAuxLineStep	step width for the trend auxiliary lines in percentage of the trend window
TrendAuxLineStyle	reference to a **TVLineStyle**-object (defines the style of an auxiliary line)
TrendBackColor	color of the background of the trend window
TrendPicture	background picture for the trend window
XScale	returns the associated x-scale object (**TVXScale**)
YScale	returns the associated y-scale object (**TVYScale**)
YScales	returns the number of the y-scales
Zoom	enables (TRUE) or disables (FALSE) the zoom in the trend window

AutoScaleGrid

Boolean AutoScaleGrid

Description

This property is used to activate and deactivate the automatic scaling of the division lines in scales.

Range of values

TRUE	automatic scaling is activated
FALSE	automatic scaling is deactivated

Remark

If a user-defined scaling is to be executed, this property = FALSE must be selected. The respective settings must be defined in the scale objects via properties **DivGrid**, **MainGrid**, **SecGrid**, **GridTextFormat**.

A more detailed explanation of this property is contained in the description of **MainGrid (TVXScale)**.

Also see

TVXScale, TVYScale

AutoScaleVisibility

Boolean AutoScaleVisibility

Description

This property is used for activating and deactivating the automatic management for the scale visibility.

Range of Values

TRUE	automatic scale management is activated
FALSE	automatic scale management is deactivated

Remark

In the default setting only those scales are visible in the trend control that were assigned a visible curve. Scales to which no visible curves are assigned are automatically hidden.

If this automatism is deactivated, the visibility of the scales does not depend on the status of the curves and can be controlled by the container itself.

Also see

TVXScale, TVYScale, TVCurve

BackColor

`OLE_COLOR` BackColor

Description

Background color of the trend control

Range of values

is defined by the Windows color palette

Also see

ForeColor

CrossHair

`Boolean` CrossHair

Description

Activate or deactivate the trendviewer crosshair.

Range of values

TRUE	activate crosshair
FALSE	deactivate crosshair.

Also see

Ruler

Curve

`TVCurve` Curve(Short Index)

Description

Returns the associated **TVCurve**-object.

Range of values

see **TVCurve**

Also see

Curves

Curves

Long Curves()

Description

Returns the number of the generated **TVCurve**-objects.

Range of values

0..

Also see

TVCurve

CurveButtons

enumCurveButtons CurveButtons

Description

The control buttons to activate and deactivate the curves can be placed at a certain position of the trend window by using this property.

Range of values

0-cbNone	Control buttons are not visible
1-cbTop	Control buttons at the top of the trend window
2-cbScale	Control buttons at the scale objects

Remark

The control buttons are used to activate and deactivate the individual curves and to display the name and status of each curve.

Display:
An activated button shows that the associated curve is visible. A deactivated button shows that the curve is invisible.
If the mouse pointer is moved along one of the curve buttons, a small window is displayed next to the mouse (for **DataWindows** = TRUE), which displays the name of the curve and the name of the scale which is assigned to the curve.

Use:
The respective curve can be switched to visible or invisible by a single left mouse button click on a curve button.

The associated curve can be switched to visible or invisible by a single left mouse button click on a curve button.
If **CurveButtons** equals cbScales, the automatic scale management should be deactivated (see **AutoScaleVisibility**), as the deactivation of the curve also hides the associated scale and scale button.

Also see

AutoScaleVisibility, **DataWindows**

DataWindows

`Boolean` DataWindows

Description

Defines whether the data window for the display of trend data like scale values and curve descriptions is to be displayed.

Range of values

TRUE	data windows are visible
FALSE	no data window

Remark

The data window corresponds to a Windows tooltip window which follows the movement of the mouse pointer and displays information on the current mouse position.

Also see

CurveButtons

DockingScales

`Boolean` DockingScales

Description

Defines whether the interactive scale positioning can be activated or deactivated by using the mouse.

The associated control elements are visible in the scales when they are activated.

Range of values

TRUE	Scale positioning via mouse move is activated
FALSE	Scale positioning via mouse move is deactivated

Also see

TVXScale, TVYScale, ReleaseKey

FixedFont

`Boolean` FixedFont

Description

Deactivates the automatic scaling of the font size.

Range of values

TRUE	automatic font size scaling is deactivated
FALSE	automatic font size scaling is activated

Remark

For more information on the automatic font size scaling see **Font**

Also see

Font, FontScaleFactor

Font

`IFontDisp*` Font

Description

Defines the font used.

Range of values

is defined by the existing working environment.

Remark

To achieve a flexible scalability of the trend control (for **FixedFont** equals FALSE) the size of the font used is automatically fit to the size of the TrendControl. This is performed by the following procedure:

At every drawing procedure of the Control the optimal font size for the current setting is computed (depending on the scale number and scale positions). The calculated font size can such be enlarged or reduced using property **FontScaleFactor** to adapt the font to the programmer's requirements.

With every modification of property **Font** the Control tries to adjust the returned font size by adapting the **FontScaleFactor.** The look of the control is protected against changes by computing limit values for the font size which must not be exceeded. Property **FontScaleFactor** is therefore calculated accurately for the desired font size but the selected font might actually be displayed smaller because of the defined limit values.

Following an internal adaptation of property **FontScaleFactor** the font size of the current font is no longer relevant to the display of the font. The selection of the current font therefore does not output the current font size.

The following sample code should be used to modify a font type without affecting the current size:

Sample Code

```
TrendViewer1.Font.Bold = True
TrendViewer1.FontScaleFactor = 2
```

If **FixedFont** equals TRUE, the font is not modified by the Trend Control.

Also see

FontScaleFactor, **HorizontalScText**, **FixedFont**, **ScaleWidthFactor**

FontScaleFactor

`Double` FontScaleFactor

Description

Enlargement reduction factor for the computed optimal font size

Range of values

>0 and <1.0	reduction of the font
1.0	no modification of the font
>1.0	enlargement of the font

Remark and sample code

This property is only relevant to (**FixedFont** equals FALSE).

For more information on the automatic font size scaling see property **Font**.

Also see

Font, **HorizontalScText**, **FixedFont**

ForeColor

`OLE_COLOR` ForeColor

Description

Defines the forecolor of the Trend Control

Range of values

is defined by the Windows palette

Remark

The forecolor is used to draw the Trend Window as well as the standard font color for the data window.

Also see

BackColor, **DataWindows**

HorizontalScText

`Boolean` HorizontalScText

Description

Horizontal description of horizontally arranged scales.

Range of values

TRUE	all scales are inscribed in horizontal orientation
FALSE	the scales are inscribed according to their alignment

Remark

HorizontalScText is set to FALSE in the predefined setting of the Trend Control. All scales are then labelled according to their alignment. A space-saving representation of the Trend Control is such provided.

Setting **HorizontalScText** to TRUE provides the horizontal display of all scale texts (except the scale designations). This type of scale inscription, however, requires a variable scale width to display the corresponding text. The width of the scales must be adapted suitably by the user or the container. To support a variable scale width the property **ScaleWidthFactor** can be used. The scale width management is considerably simplified by deactivating the automatic font size adaptation (see **FixedFont**).

Also see

Font, **FontScaleFactor**, **ScaleWidthFactor**, **FixedFont**

HorizontalScTextAlign

`Boolean` HorizontalScTextAlign

Description

Alignment of text in horizontal inscription.

Range of values

0	Alignment at the division line
1	Left-aligned or right-aligned with respect to the text margin

Also see

Font, **HorizontalScText**, **FixedFont**

Interactive

`Boolean` Interactive

Description

All mouse interactions are deactivated in case of FALSE. Calling the user dialog by double-clicking the left mouse button is not affected by **Interactive**.

Range of values

TRUE	mouse control is activated
FALSE	mouse control is deactivated

Also see

Interactive (TVXScale), **Interactive (TVYScale)**

LockUpdate

`Boolean` LockUpdate

Description

When **Lockupdate** is set to TRUE the redrawing of the Control is disabled. Changes of the trend properties are not visible in the Control afterwards.

Range of values

TRUE	Changes to the properties do not affect the appearance of the Control
FALSE	Changes in the properties are immediately visible, default setting

Remark

The Control should have been completely redrawn at least once before **Lock-Update** is set to TRUE.

Changing the window size of the control while **LockUpdate** is set to TRUE does not affect the representation. **LockUpdate** should therefore always be set to FALSE when changing the size of the Control.

If **LockUpdate** is set to TRUE and a background process runs to draw the control, the program waits until the background process has been terminated.

Sample Code

see **ActualMax, ActualMin**

Also see

TVRange

MarginFactor

`Double` MarginFactor(enumMargins index)

Description

defines the scaling factor for the margins of the trend

Range of values

arbitrary value

Also see

MarginFactor (TVXScale), MarginFactor (TVYScale)

Orientation

`Short` Orientation

Description

defines the display mode of the trend window

Range of values

0	trend window and scales are automatically adapted to the total size of the trend window
1	fixed trend window. Scales are adapted.

Remark

In a fixed trend window, the smallest distance between trend window and trend margin is used as standard width/height for the scales.

A more detailed description of this property is contained in **SetTrendWindow**

Also see

TVXScale, TVYScale, SetTrendWindow

OVBarsBackColor

`OLE_COLOR` OVBarsBackColor

Description

background color of the bar which displays the data interval (data scale) or the time interval (time scale)

Range of values

is defined by the Windows color palette

Also see

OVBarsRangeColor, OverviewBars

OVBarsRangeColor

`OLE_COLOR` OVBarsRangeColor

Description

background color of the range for the overview bars

Range of values

is predefined by the Windows color palette

Also see

OVBarsBackColor, **OverviewBars**

OverviewBars

`Boolean` OverviewBars

Description

Activates (`TRUE`) or deactivates (`FALSE`) the overview bars of the scales.

Range of values

TRUE	the overview bars are visible
FALSE	the overview bars are hidden

Remark

The overview bars are automatically hidden, if the font size falls below a defined limit value.

Also see

OVBarsBackColor, **OVBarsRangeColor**

Picture

`IPictureDisp*` Picture

This property returns the current window content as bitmap.

Remark

The Picture-property can only return a valid bitmap if the control has been completely redrawn at least once.

The bitmap created by Picture can e.g. be used for printouts under Visual Basic. To produce a relatively high-quality printout, the **PrintTrend**-method should be used.

If **LockUpdate** is set to TRUE and a background process runs to redraw the control, the system waits until the background process has been terminated.

Sample code

```
Printer.PaintPicture Scale1.Picture, 10, 10
Printer.EndDoc
```

Also see

PrintTrend

ReleaseKey

`Boolean` ReleaseKey

Description

Activate or deactivate (TRUE) or deactivate (FALSE) release key.

Range of values

TRUE	release key activated
FALSE	release key deactivated

Remark
The release key enables to position the scales interactively using the mouse.

Caution
This property is not available in every license type.

Also see

DockingScales

Ruler

TVRuler Ruler

Description

Reference to a **TVRuler-**object (curve ruler)

Range of values

see **TVRuler**

Rotated

Boolean Rotated

Description

In general, the x-scale of a trend is horizontally oriented with a left-oriented scale minimum and a right-oriented scale maximum.

In special cases, a vertical orientation of the x-scale with the scale minimum at the bottom is requested. This type of Trend can be produced by setting **Rotated** to TRUE.

Range of values

TRUE	vertical x-scale
FALSE	horizontal x-scale (standard)

ScaleLegends

Boolean ScaleLegends

Description

Activates (TRUE) or deactivates (FALSE) the scale legends.

Range of values

TRUE	the legends are visible
FALSE	the legends are hidden

Also see

OverviewBars

ScaleWidthFactor

`Double` ScaleWidthFactor

Description

Enables to expand all vertically oriented scales by the defined factor.

Range of values

all positive values

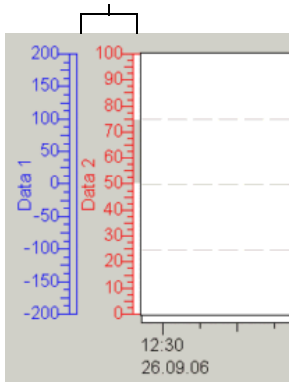
Remark

This property is required to enable the representation of horizontal scale inscriptions in vertical scales.

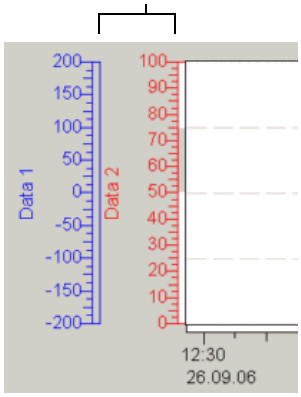
Before each redrawing the trend control calculates the position and size of all trend elements also computing the width of all scales. The scale width of all vertically oriented scales is subsequently multiplied by the value of **ScaleWidthFactor**.

The following figures illustrate the meaning of this property:

ScaleWidthFactor = 1.0



ScaleWidthFactor = 1.3



In the above-illustrated example, the value of **ScaleWidthFactor** is increased from 1.0 to 1.3 to ensure an accurate representation of the scale inscription.

Also see

Font, **FontScaleFactor**, **HorizontalScText**

StripChartsStyle

`enumStripChartsStyle` StripChartsStyle

Description

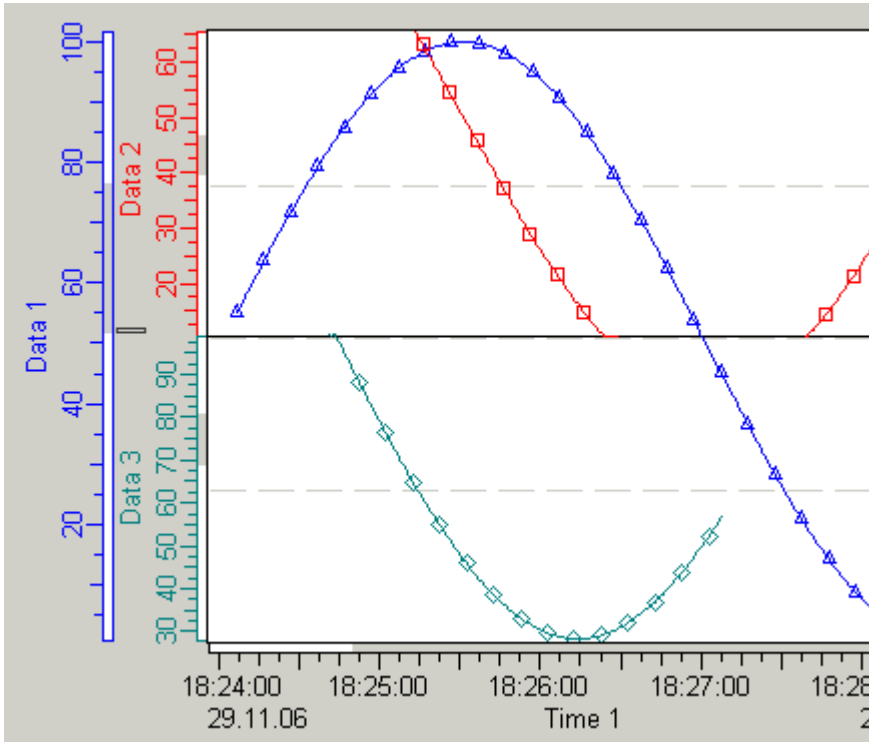
By using this property the Trend Control can be reconfigured to display strip charts.

Range of values

`0-scNoStripCharts`	no strip charts
`1-scLeftSide`	strip chart at the left margin
`2-scRigthSide`	strip chart at the right margin
`3-lsBothSides`	strip chart at both margins

Remark

When the display of strip charts is activated, the trend window is divided into strips. These strips are aligned the way the individual trend margin is divided by the scales. Only those scales are affected which are placed directly at the margin of the trend window. Curves that are assigned to theses scales are truncated as soon as they leave the respective strip. The following figure shows a specific example of a strip chart of the trend control for **StripChartsStyle** equals `scLeftSide`:



The above-illustrated example shows that scales „Data 2“ and „Data 3“ form strips and that the curves associated to these scales are truncated as soon as they exceed the limits of these strips. As curve associated to „Data 1“ is not assigned to a strip, it is displayed in the entire trend window.

TrendAuxLineCount

`Short` TrendAuxLineCount

Description

Defines the number of auxiliary lines in the trend window.

Range of values

arbitrary integer value

Also see

TrendAuxLineStart, **TrendAuxLineStep, TrendAuxLineStyle**

TrendAuxLineStart

`Short` TrendAuxLineStart

Description

Defines the die start position of the first auxiliary line.

Range of values

arbitrary value

Also see

TrendAuxLineStep, **TrendAuxLineStyle, TrendAuxLineCount**

TrendAuxLineStep

`Short` TrendAuxLineStep

Description

Defines the step width of the Trend Window by auxiliary lines in percentage to the height of the trend window

Range of values

`<=0%`	no auxiliary lines
`1% - 99%`	distance between individual auxiliary lines (`25%` by default)
`>=100%`	no auxiliary lines

Remark

For **TrendAuxLinesStep** equals `10%`, the trend window is divided into 10 ranges by auxiliary lines, for **TrendAuxLineStep** equals `25%` the trend window is quartered.

Also see

TrendAuxLineStyle, TrendAuxLineCount, TrendAuxLineStart

TrendAuxLineStyle

`TVLineStyle` TrendAuxLinesStyle

Reference to a **TVLineStyle**-object. Defines the line style of the trend auxiliary lines.

Range of values

see **TVLineStyle**

Remark

The drawing of lines of line style unequal `psSolid` and of line width unequal `0` is considerably more time-consuming.

Also see

TrendAuxLineStep, TrendAuxLineCount, TrendAuxLineStart

TrendBackColor

`OLE_COLOR` TrendBackColor

Description

Background color of the trend window

Range of values

is defined by the Windows color palette

Also see

BackColor

TrendPicture

`IPictureDisp*` TrendPicture

Description

background color of the trend window

Range of values

any Windows bitmap

Remark

The Trend Control does not execute its own color management to implement the palette of the selected trend background picture. The application must perform this.

Also see

PrintTrend

XScale

`TVXScale` XScale(short index)

Description

Returns an x-scale object (index).

Range of values

`see` **TVXScale**

Remark

The "index" can only be 0 or 1, as TrendViewer only provides a maximum of 2 x-scales.

Also see

YScale, TVXScale

YScale

`TVYScale` YScale(short index)

Description

returns or defines a y-scale object.

Range of values

`see` **TVYScale**

Remark

The "index" should be within the generated y-scale objects (0..**YScales**). The first 10 y-scales are generated standardized when applying the trend control. Up to 512 additional y-scales can be generated.

Also see

XScale, TVYScale

YScales

`Short` YScale()

Description

returns the number of the generated y-scale objects

Range of values

```
0..512              Number of generated y-scales
```

Also see

TVYScale

Zoom

Short Zoom

Description

Sets the zoom method within the trend window.

Range of values

0	deactivated zoom method
1	activated extended zoom method of the trend control
2	zoom method is activated by means of the lasso

5.1.2 TVXScale

Properties of the x-scale object:

AbsoluteTime	switches the time scale between absolute time (TRUE) and time span (FALSE) (see **Type**)
AutoFollowMode	activates (TRUE) or deactivates (FALSE) the automatic switch to follow mode
AuxCount	number of auxiliary lines
AuxLine	defines the arrangement of the auxiliary lines
AuxLineStyle	reference to a **TVLineStyle-**object
AuxStart	starting point of the first auxiliary line
AuxStep	step width of auxiliary lines in percentage
DataRangeEnabled	shows (TRUE) or hides (FALSE) the display of the current range of values within an overview bar
DivGrid	defines the division grid. The division grid defines the number of divisions within the secondary grid (**SecGrid**)
EdgeTextFormat	format for the left/right text output in the legend
GridTextFormat	text format of the value display at the main grid lines
Index	instance of the scale
FollowMode	activates (TRUE) or deactivates (FALSE) the follow mode
FollowModeButton	shows (TRUE) or hides (FALSE) the button for the follow mode at the scale
ForeColor	foreground color of the scale
Enabled	activates (TRUE) or deactivates (FALSE) the scale (changeable only for **AutoScaleVisibility**=FALSE)
Index	instance of the scale
Interactive	activates (TRUE) or deactivates (FALSE) the mouse control
LegendDateVisible	extended display of the date in the legend of the scale (TRUE) - or deactivate (FALSE)
MainGrid	defines the main grid. A main grid is the distance between two main division lines
MarginFactor	factor for the edges (left/right) of the scale
Mirrored	mirrors the scale
Modulo360Degrees	only displays values between 0° and 360° in an angular scale (always forms a positive modulo 360°)
Position	reference to a **TVPosition** object (scale positioning)
Priority	defines the priority of the scale
Range	reference to a **TVRange-**object (data range of the scale)
RangeMoveOnly	allows for TRUE only to move the range of values
Res ;	maximum display exactness (not persistent for time axes)
SecGrid	defines the secondary grid. The secondary grid defines the number of divisions within the main grid (**MainGrid**)
Shared	Defines if all curves use the same x-data (TRUE) or if

	each curve requires an own x-data set
Text	name of the scale (appears in the legend)
TextAlign	text alignment in the legend
ToolTipFormat	format for the text and value output in the tooltip (data window)
Type	defines the display format of the scale
Visible	shows (TRUE) or hides (FALSE) the scale (**AutoScaleVisibility**=FALSE)

AbsoluteTime

`Boolean` AbsoluteTime()

A time scale is either displayed with a date and time or with a time span.

Range of values

FALSE	time span (default setting))
TRUE	absolute time

Remark

The inscription of the scale either shows the date or the time in absolute time display. Time spans are indicated in seconds, minutes and days.

The zero scale mark for the display of time spans is defined by **DataMax, DataMin**. All ranges of values (see **ActualMax, ActualMin, DataMax, DataMin**) must be indicated in absolute time values to enable an interactive switching between absolute time and time span.

This property is only relevant to time scales of **Type** = `stTime`.

Also see

ActualMax, ActualMin, DataMax, DataMin, **Res ;**

AutoFollowMode

`Boolean` AutoFollowMode()

Description

Activates or deactivates the automatic follow mode (see **FollowMode**).

Range of values

TRUE	automatically activates the follow mode
FALSE	does not automatically activate the follow mode

Remark

If Auto-Follow Mode equals TRUE, the follow mode is activated as soon as the maximum of the x-values reaches the maximum of displayed range.

Also see

FollowMode, **FollowModeButton**

AuxCount

`Long` AuxCount()

Description

defines the number of auxiliary lines which are to be drawn.

Range of values

-1	unlimited number of auxiliary lines (e.g. continuous display of auxiliary lines)
<=0	no auxiliary lines
>0	number of auxiliary lines drawn

Remark

When creating x-scale auxiliary lines starting at position **AuxStart** a certain number of auxiliary lines are drawn which are defined in **AuxCount** with the step width defined in **AuxStep**.

A more detailed description of these properties is contained in the description of **AuxLine**.

Also see

AuxStart, **AuxStep**, **AuxLineStyle, AuxLine**

AuxLine

`short` AuxLine()

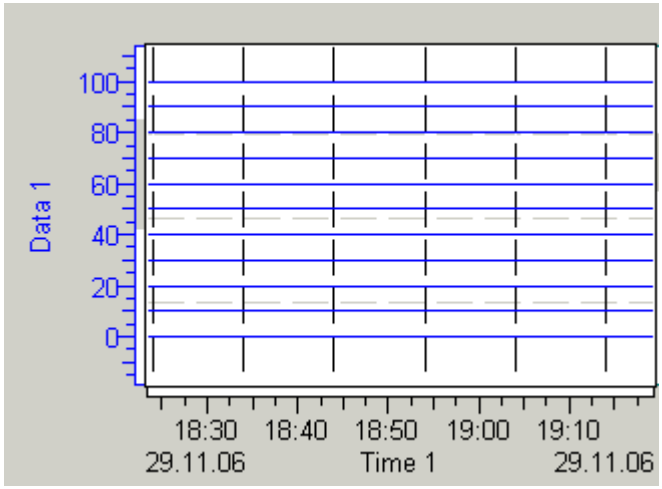
Description

Defines the number of auxiliary lines

Range of values

`0`	auxiliary lines are defined via the number of lines (**AuxCount**), start (**AuxStart**) and step (**AuxStep**)
`1`	Auxiliary lines are aligned at the division lines (**DivGrid, MainGrid, SecGrid**) of the scales using **AuxStep**

Example for AuxLine = 0 at the x- and y-scale:



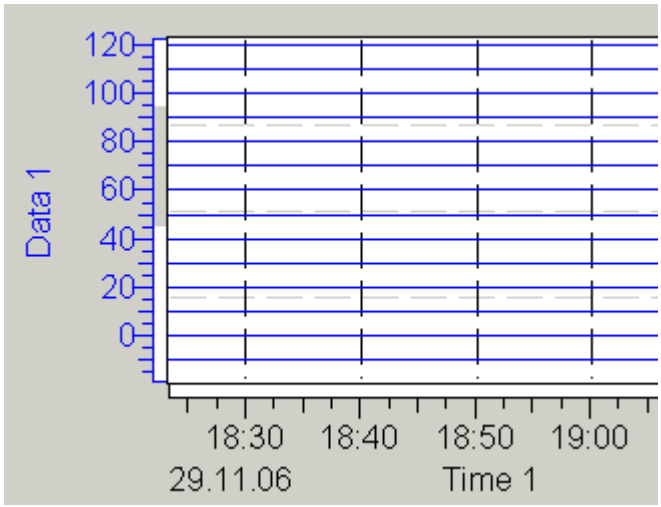
In above-shown example the following settings were made:

Setting of the x-scale auxiliary lines to continuous grid (every 10 min):
AuxStart() = 0.0
AuxStep() = 600.0
AuxCount() = -1
AuxLineStyle() = `psDash`

Setting of the y-scale auxiliary lines with:
AuxStart(`dsYScale1`) = 0.0
AuxStep(`dsYScale1`) = 10.0

AuxCount(`dsYScale1`) = 11

Example of AuxLine = 0 at the x-scale and **AuxLine** = 1 at the y-scale:



The following settings were made in the above-illustrated example:

Settings of the x-scale auxiliary lines to continuous grid (every 10 min):
AuxStart() = 0.0
AuxStep() = 600.0
AuxCount() = -1
AuxLineStyle() = `psDash`
Setting of the y-scale auxiliary lines with:
AuxStart(`dsYScale1`) = 0.0 (not relevant)
AuxStep(`dsYScale1`) = 1 (**MainGrid** and **SecGrid** orientation)
AuxCount(`dsYScale1`) = 0 (not relevant)

Remarks

If AuxLine = 1 property **AuxStart** and **AuxCount** has no meaning!

Also see

AuxStart, **AuxStep**, **AuxLineStyle, AuxCount**

AuxLineStyle

`TVLineStyle` AuxLineStyle()

Description

Reference to a **TVLineStyle**-object. Defines the line style of the x-scale auxiliary lines.

Range of values

see **TVLineStyle**

Also see

AuxStart, **AuxStep**, **AuxLine, AuxCount**

AuxStart

`Double` AuxStart()

Description

Defines the starting value for drawing auxiliary lines.

Range of values

arbitrary value

Remark

A detailed description of this property is contained in the description of **AuxLine.**

Also see

AuxCount, AuxStep, AuxLine, AuxLineStyle

AuxStep

`Double` AuxStep(`enumXScales` ScaleNumber)

Description

Defines the step width between the auxiliary lines. The step width has a different range of values depending on the **AuxLine**-type.

Range of values

for **AuxLine** = 0:

arbitrary value

for **AuxLine** = 1:

`<0`	no auxiliary lines
`0`	alignment of the auxiliary lines to **DivGrid, MainGrid, SecGrid**
`1`	alignment of the auxiliary lines to **MainGrid, SecGrid**
`2`	alignment of the auxiliary lines to **MainGrid**
`>2`	no auxiliary lines

Remark

A detailed description of these properties is contained in the description of **AuxLine**.

Also see

AuxCount, AuxStart, AuxLine, AuxLineStyle DivGrid, MainGrid, SecGrid

DataRangeEnabled

`Boolean` DataRangeEnabled()

Description

Activates the display of the current data interval in the overview bar. The data interval is defined by the **DataMax, DataMin DataMax, DataMin** of the **TVRange**-object.

Range of values

FALSE	not activated, default setting
TRUE	data interval is displayed

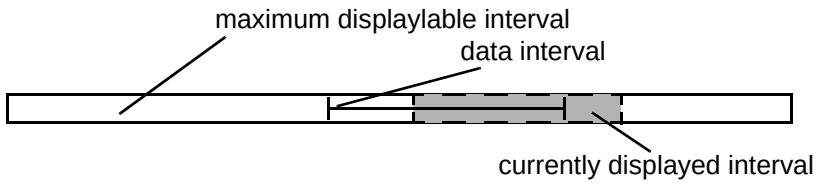
Remark

When curves of measured data are displayed it is helpful to know the maximum and the minimum value of the measured data set so that the curve of measured values can be displayed in the best resolution. The display of the data interval also helps to find out which portion of the data interval is displayed in relation to the scale.

The data range is defined during runtime. In the developing phase, a standard data range is defined to suggest the later appearance of the scale.

Example (data scale)

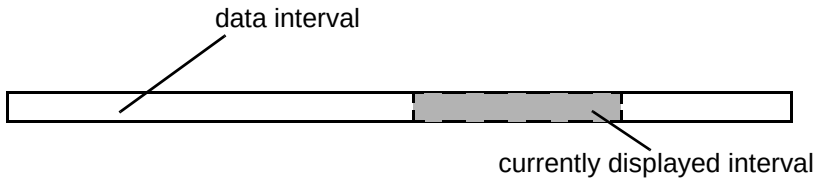
The connection between the maximum displayable interval, the currently displayed interval and the data interval is demonstrated in the following sketch:



In this constellation the scale currently shows a small extract of the entire possible graphic. Also only a section of the entire data interval can be viewed.

Example (time scale)

As the maximum displayable time range of a time scale cannot be modified, the overview bar is therefore displayed slightly different than illustrated in the following sketch:



The overview bar of the time scale corresponds to the typical display of these overview bars.

Sample code

```
Dim XScale As TRENDVIEWERLib.TVXScale
Dim XRange As TRENDVIEWERLib.TVRange

Set XScale = Trend1.XScale(0)
Set XRange = XScale.Range

XScale.DataRangeEnabled = False
XRange.DataMin = 10
XRange.DataMax = 100

XScale.DataRangeEnabled = True
```

Also see

DataMax, DataMin

DivGrid

Long DivGrid()

Description

Defines the division grid. The division grid defines the number of divisions in the secondary grid (**SecGrid**).

Range of values

arbitrary integer value

Remark

A more detailed description of this property is contained in the description of **MainGrid**.

DivGrid is only relevant if AutoScaleGrid equals FALSE, otherwise the settings from the XML-format file or the standard settings of TrendViewer are applied.

Also see

MainGrid, **SecGrid, AutoScaleGrid**

EdgeTextFormat

BSTR EdgeTextFormat()

Description

Defines the format of the value output "left and right" in the legend.

Range of values

formatting type in C-notation e.g. "%d.%m.%y"

Remarks

EdgeTextfFormat is only relevant if **AutoScaleGrid** equals FALSE otherwise the settings of the XML-format file or the standard settings of *TrendViewer* are applied.

AutoScaleGrid, LegendDateVisible

Enabled

`Boolean` Enabled()

Description

activates or deactivates an x-scale

Range of values

TRUE	Scale is activated
FALSE	Scale is deactivated

Remark

This property can only be modified if **AutoScaleVisibility** = FALSE otherwise the Trend-Control takes over the activation of the scales.

X-scales can only be switched to visible (see **Visible**) if they had previously been activated.

Also see

AutoScaleVisibility, **Visible**

FollowMode

`Boolean` FollowMode()

Description

Defines if the Control works in follow mode.

In follow mode the visible range of the x-axis is always moved to the x-axis such that the right side of the trend window corresponds to the current value of the maximum of the x-values. In common series of measurements the most current data is such always displayed in the trend window.

Range of values

TRUE	follow mode is activated
FALSE	follow mode is deactivated

Remark

This property may change during the runtime if the following events occur:

- the user has changed the maximum of the x-scale
- the user has deactivated the follow mode by a mouse click on the follow mode control button

from FALSE to TRUE:

- for **AutoFollowMode** equals TRUE the maximum of the x-data reached the right margin of the trend window
- the user activated the follow mode by a mouse click on the follow mode control button

To protect this property against changes by the user, the x-scale must be set to static by deactivating the interactive control (see **Interactive**) and the control button for the follow mode (see **FollowModeButton**) must be hidden.

Also see

AutoFollowMode, **FollowMode, FollowModeButton**, **Interactive**

FollowModeButton

Boolean FollowModeButton

Description

Defines whether the control and display buttons for the follow mode of the x-scales should be displayed or not.

Range of values

TRUE	control buttons for the follow mode are displayed
FALSE	control buttons for the follow mode are not displayed

Remark

The control buttons for the follow mode (for **FollowModeButtons** equals FALSE) are used for the display as well as for the control.

Display:
If the follow mode of an x-scale is activated, the respective button appears pressed. The follow mode symbol is then colored light green.

Use:
The follow mode of the associated x-scale can be activated or deactivated by a single left mouse button click on a control button.

Also see

AutoFollowMode, FollowMode

ForeColor

OLE_COLOR ForeColor()

Description

scale and description color of the x-scale

Range of values

is defined by the Windows color palette

Also see

BackColor

GridTextFormat

BSTR GridTextFormat()

Description

defines the format of the value display for the main grid

Range of values

formatting type in C-notation e.g. "%d.%m.%y"

Remark

A more detailed description of this property can be found in the description of **MainGrid**.

GridTextFormat is only relevant if AutoScaleGrid equals FALSE otherwise the settings of the XML-format file or the standard settings of TrendViewer are applied.

MainGrid, AutoScaleGrid

Index

`Short` Index()

Description

Delivers the index value of the x-scale.

Range of values

`0..MaxXScales`

Interactive

`Boolean` Interactive()

Description

Deactivates the interactive change of the display range of the x-scale by means of the mouse (TRUE) or (FALSE).

Range of values

TRUE	user can change the display range of the x-scale by using the mouse
FALSE	modification disabled

Remark

The zoom functions of the trend window do not affect the x-scale in the event of **Interactive** = FALSE.

Also see

Zoom (TrendCtrl)

LegendDateVisible

`Boolean` LegendDateVisible()

Description

Defines whether the extended data display of the x-scale in the legend of the scale is activated (TRUE) or deactivated (FALSE).

Range of values

TRUE	extended data display of the x-scale is displayed on the left and right of the legend
FALSE	extended data display is not displayed

Remark

Only relevant in a time scale (tsTime). If the time scale was zoomed e.g. such that only hours, minutes and seconds can be displayed, the extended data display only shows the day, month and year.

Also see

EdgeTextFormat

MainGrid

`Double` MainGrid()

Description

The main grid defines the distance between two main division lines.

Range of values

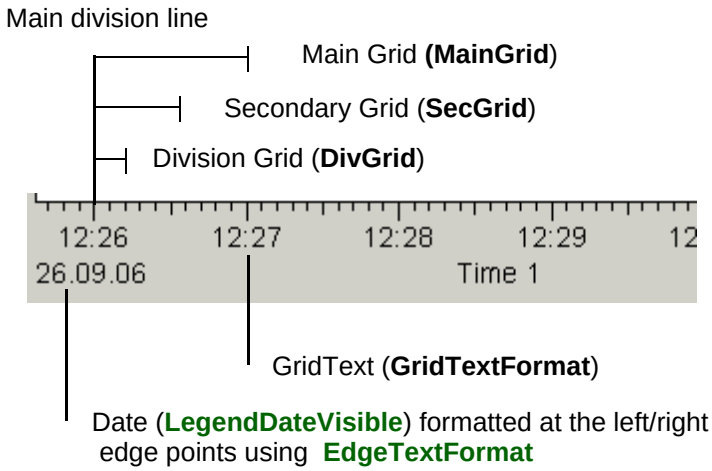
arbitrary value

Remark

The specification of the main division grid is only relevant if **AutoScaleGrid** is set to FALSE otherwise the main division grid is defined via the XML-format file or the standard setting of *TrendViewer*.

Example

The following figure displays the connection between **MainGrid**, **SecGrid** and **DivGrid** as well as **GridTextFormat, EdgeTextFormat** and **LegendDateVisible**:




```
Dim xScale1 As TRENDVIEWERLib.TVXScale
Dim yScale1 As TRENDVIEWERLib.TVYScale
Dim Range1 As TRENDVIEWERLib.TVRange

Set xScale1 = TrendViewer1.XScale(0)
Set yScale1 = TrendViewer1.YScale(0)

TrendViewer1.AutoScaleGrid = False

' define y-Scale
yScale1.MainGrid = 50
yScale1.SecGrid = 2
yScale1.DivGrid = 2

yScale1.GridTextFormat = "%.lf"
yScale1.ToolTipFormat = "%.1f"

' define x-Scale
Set Range1 = xScale1.Range

' five minutes before and 5 minutes after
Range1.ActualMax = Now + 300 / 86400
Range1.ActualMin = Now - 300 / 86400


' each minute a main-line
xScale1.MainGrid = (60 / 86400)
xScale1.SecGrid = 2
xScale1.DivGrid = 5
```

```
xScale1.GridTextFormat = "%H:%M"
xScale1.ToolTipFormat = "%d.%m.%y %H:%M:%S"

xScale1.LegendDateVisible = True
xScale1.EdgeTextFormat = "%d.%m.%y"
```

Also see

SecGrid, DivGrid, GridTextFormat, AutoScaleGrid, GridTextFormat, Edge-TextFormat

---

MarginFactor

`Double` MarginFactor()

Description

Defines the left and right margin factor of the x-scale.

Range of values

arbitrary value

Remark

This property is used for later extensions. In this *TrendViewer* version it has no functionality as parallel x-scales are not permitted.

Also see

MarginFactor (TrendCtrl), Priority, MarginFactor (TVYScale)

Mirrored

`Boolean` Mirrored()

Description

mirrors the x-scale

Range of values

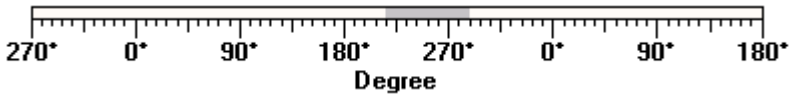
TRUE	scale is mirrored
FALSE	scale is displayed in normal orientation

Modulo360Degrees

`Boolean` Modulo360Degrees()

Description

The degree scale is normally displayed with arbitrary degree values. When this property is set to TRUE, all degree values are mapped to a degree range of 0° and 360° by means of a modulo function as illustrated in the following example:



Range of values

TRUE	Display between 0° and 360°
FALSE	normal degree scale

Remark

This setting is only relevant to degree scales.

Also see

Type

Position

`TVPosition` Position()

Description

Reference to a **TVPosition**-object. Defines where the x-scale is positioned.

Range of values

`see` **TVPosition**

Priority

`Short` Priority()

Description

Display priority of the x-scale.

Range of values

arbitrary integer value

Remark

The display priority has a special effect on the margins. Depending on the *TrendViewer* setting overlapping of individual scales can be avoided using the **MarginFactor (TrendCtrl)**.

Also see

MarginFactor (TrendCtrl)

Range

`TVRange` Range()

Description

Reference to a **TVRange** object. Defines the limits of the individual scale ranges.

Range of values

see **TVRange**

RangeMoveOnly

`Boolean` RangeMoveOnly()

Description

The display range of the x-scale can be moved by means of the mouse control.

Range of values

TRUE	moving only enabled
FALSE	moving, enlarging and reducing of the range enabled, default setting

Remark

For **RangeMoveOnly** = TRUE the zoom functions of the trend window have no effect on the x-scale. The display range can be modified via the extended zoom functions of the trend window by moving the trend section.

Also see

Interactive, **Zoom**

Res ;

`Double` Res()

Description

Minimum resolution which can be displayed through the scale division.

If the x-scale is defined as time scale, only property **RangeRes** can be modified, properties **Max, Min** are assigned defined preset values which are specified by the internal Windows time format.

Predefined settings (data scale)

	data scale	time scale
Max, Min	200, -200	preset value, preset value
Res	0,001	0,001

Also see

TVRange

SecGrid

`Long SecGrid()`

Description

The secondary grid defines the number of divisions within the main grid (**MainGrid**)

Range of values

arbitrary integer value

Remarks

A more detailed description of these properties is contained in the description of **MainGrid**.

SecGrid is only relevant if AutoScaleGrid equals FALSE, otherwise the settings of the XML-Format file or the standard settings of TrendViewer are applied.

Also see

MainGrid, **DivGrid, AutoScaleGrid, MainGrid (TVXScale)**

Shared

`Boolean Shared`

Description

Determines if all curves use the same x-data (TRUE) or if each curve requires an own x-data set (FALSE).

Range of values

TRUE	all curves use the same x-data
FALSE	all curves possess own x-data

Remark

This property defines a fundamental property of the internal data storage:

If **Share** equals TRUE only a set of x-data is stored, the data points of all activated curves together form a data point of type (X,Ya,Yb,...,Yn). For this data management it is possible to display the trend as phase diagram.

If **Share** equals FALSE, an own data set is created for each curve. In this event, the curves are all independent and each curve has independent data points of type (Xa,Ya).

When processing curve data this property must always be considered, as inconsistency in the data set might easily result from maloperations. This problem is detailed in the remarks on the individual properties and methods.

When changing the property IdenticalXData all internal data sets are automatically deleted.

Also see

Range, Res ;, TVData

Text

BSTR Text()

Description

scale text for the description of the x-scale

Range of values

arbitrary text

Remark

If the space provided in the text display field is inefficient, the inscription is truncated and marked by three points.

Also see

LegendDateVisible, EdgeTextFormat, ScaleLegends, TextAlign

TextAlign

`Short` TextAlign()

Description

text alignment of the scale text in the legend

Range of values

`0-taCenter`	centered text
`1-taLeft`	left-aligned text
`2-taRight`	right-aligned text

Also see

Text, ScaleLegends

ToolTipFormat

BSTR ToolTipFormat()

Description

determines the format of the value display for the ToolTip (**DataWindows**)

Range of values

formatting type in C-notation, e.g. "%d.%m.%y"

Also see

DataWindows

Type

`enumScaleTypes` Type()

Description

This property allows to select the data type of the x-scale (decimal, time-, degree or logarithmic display).

Range of values

`0 -stDecimal`	decimal scale
`1 -stTime`	time scale
`2 -stDegree`	degree scale
`3 -stLogarithmic`	logarithmic scale

Also see

enumScaleTypes, **AbsoluteTime**, **Modulo360Degrees**

Visible

`Boolean` Visible()

Description

This property is used to activate and deactivate the x-scale.

Range of values

TRUE	visible
FALSE	not visible

Remark

This property can only be modified if **AutoScaleVisibility** = FALSE otherwise the Trend Control itself activates the scales.

Only previously activated x-scales can be set to visible (see **Enabled**).

Also see

AutoScaleVisibility, **Enabled**, **TVPosition**

5.1.3 TVYScale

Properties of the y-scale object:

AuxCount	number of the auxiliary lines
AuxLine	style of the auxiliary line (normal display or alignment at the main division line)
AuxLineStyle	reference to a **TVLineStyle-**object (defines the style of an auxiliary line)
AuxStart	starting point of the first auxiliary line
AuxStep	step width of the auxiliary lines in percentage
DataRangeEnabled	shows (TRUE) or hides (FALSE) the display of the current range of values within an overview bar (not modifiable for **AutoScaleVisibility**=TRUE)
DivGrid	defines the division grid. The division grid determines the number of the divisions within the secondary grid (**SecGrid**)
EdgeTextFormat	format for the left/right text output in the legend
Enabled	activates (TRUE) or deactivates (FALSE) the scale (modifiable only for **AutoScaleVisibility**=FALSE)
ForeColor	foreground color of the scale
GridTextFormat	text format of the value display at the main grid lines
Index	entity of the scale
Interactive	activates (TRUE) or deactivates (FALSE) the mouse control
MainGrid	defines the main grid. A main grid is the distance between two main grid lines.
MarginFactor	factors for the margins (top/bottom) of the scale
Mirrored	mirrors the scale
Modulo360Degrees	displays in an angular scale only values between 0° and 360° (always forms a positive modulo 360°)
Position	reference to a **TVPosition-**object (positioning of the scale)
Priority	defines the priority of the scale
Range	reference to a **TVRange-**object (data range of the scale)
RangeMoveOnly	allows (TRUE) to move the range of values only
Res	maximum display accuracy (not persistent in time axes)
SecGrid	defines the secondary grid. The secondary grid defines the number of divisions in a main grid (**Main-Grid**)
Text	name of the scale (displayed in the legend)
TextAlign	text alignment in the legend
ToolTipFormat	format for the text and value output in the tooltip (data window)
Type	defines the display format of the scale
Visible	shows (TRUE) or hides (FALSE) the scale (only for **AutoScaleVisibility**=FALSE)
XYScaleEnabled	activate scale as XY-scale (TRUE)

AuxCount

`Long` AuxCount()

Description

defines the number of auxiliary lines which are to be marked

Range of values

-1	unlimited number of auxiliary lines (e.g. for continuous display of auxiliary lines)
<=0	no auxiliary lines
>0	number of drawn auxiliary lines

Remark

When generating y-scale auxiliary lines, starting at position **AuxStart** a number of auxiliary lines defined in **AuxCount** with step width **AuxStep** are drawn.

A more detailed description with examples of this property is contained in the description of **AuxLine (x-scale)**.

Also see

AuxStart, AuxStep, AuxLine, AuxLineStyle, AuxCount (TVXScale)

AuxLine

`short` AuxLine()

Description

Defines the arrangement of the auxiliary lines

Range of values

0	Auxiliary lines are defined via the number of the lines (**AuxCount**), Start (**AuxStart**) and Step (**AuxStep**)
1	Auxiliary lines are aligned via the step at the division lines of the scales

Remark

If AuxLine = 1 properties **AuxStart** and **AuxCount** have no meaning

A more detailed description with samples of this property is contained in the description of **AuxLine (TVXScale)**.

Also see

AuxStart, AuxStep, AuxCount, AuxLineStyle, AuxLine (TVXScale)

AuxLineStyle

`TVLineStyle` AuxLineStyle()

Description

Reference to a **TVLineStyle**-object. Defines the line style of the y-scale auxiliary lines.

Range of values

see **TVLineStyle**

Also see

AuxStart, AuxStep, AuxCount, AuxLine, TVLineStyle

AuxStart

`Double` AuxStart()

Description

defines the start value for drawing the auxiliary lines

Range of values

arbitrary value

Remark

A more detailed description of this property is contained in the description of **AuxLine (TVXScale)**.

Also see

AuxStep, AuxCount, AuxLine, AuxLineStyle, AuxLine (TVXScale)

AuxStep

`Double` AuxStep()

Description

defines the step width between the auxiliary lines

Range of values

arbitrary value

Remark

a more detailed explanation of this property is contained in the description of **AuxLine (TVXScale)**.

Also see

AuxStart, AuxCount, AuxLine, AuxLineStyle, AuxLine (TVXScale)

DataRangeEnabled

`Boolean` DataRangeEnabled()

Description

Activates the display of the current data interval in the overview bar. The data interval is defined by the properties **DataMax, DataMin**.

Range of values

FALSE	not activated, default setting
TRUE	data interval is displayed

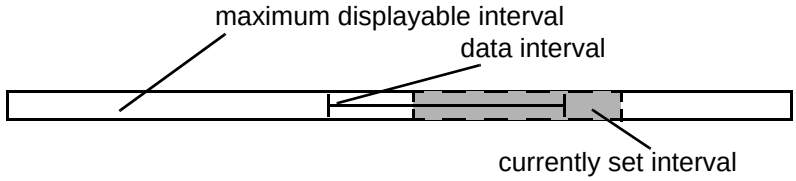
Remark

In the display of measured curves it is useful to know the largest and the smallest value of the measured data set, such that the curve of measured values can be displayed in the best possible resolution. The display of the data interval also helps to find out which section of the data interval is displayed in relation to the scale.

The data range can only be defined during the runtime. In the developing phase a standard data range is defined to suggest the later appearance of the scale.

Example (Data scale)

The connection between the maximum displayable interval, the currently displayed interval and the data interval is shown in the following figure:



In this constellation the scale currently only displays a small section of the entire possible graphic display. Likewise, only a small portion of the data interval can be seen.

Sample program

```
Dim YScale As TRENDVIEWERLib.TVYScale
Dim YRange As TRENDVIEWERLib.TVRange

Set YScale = Trend1.YScale(0)
Set YRange = YScale.Range

YScale.DataRangeEnabled = False
YRange.DataMin = 10
YRange.DataMax = 100

YScale.DataRangeEnabled = True
```

Also see

TVRange

DivGrid

Long DivGrid()

Description

Defines the division grid. The division grid specifies the number of divisions within the secondary grid (**SecGrid**).

Range of values

arbitrary integer value

Remark

A detailed description of this property is contained in the description of **MainGrid (TVXScale)**.

DivGrid is only relevant if AutoScaleGrid equals FALSE otherwise the settings of the XML-format file or the standard settings of TrendViewer are applied.

Also see

MainGrid, SecGrid, AutoScaleGrid, MainGrid (TVXScale)

EdgeTextFormat

BSTR EdgeTextFormat()

Description

Defines the format of the value output "left and right" in the legend.

Range of values

formatting type in C-notation e.g. "%d.%m.%y"

Remark

EdgeTextFormat is only relevant if **AutoScaleGrid** equals FALSE otherwise the settings of the XML-format file or the standard settings of *TrendViewer* are applied.

A more detailed description of these properties in contained in the description of **MainGrid (TVXScale)**.

AutoScaleGrid, LegendDateVisible, MainGrid (TVXScale)

Enabled

Boolean Enabled()

Description

activates or deactivates a y-scale

Range of values

TRUE	scale is activated
FALSE	scale is deactivated

Remark

This property can only be modified if **AutoScaleVisibility** = FALSE otherwise the Trend Control handles the activation of the scales itself.

Only previously activated y-scales can be set to visible (see **Visible**).

Also see

AutoScaleVisibility, **Visible**

ForeColor

`OLE_COLOR` ForeColor()

Description

scales and description color of the x-scale

Range of values

is defined by the Windows color palette

Also see

BackColor

GridTextFormat

`BSTR` GridTextFormat()

Description

defines the format of the value display for the main grid

Range of values

formatting type in C-notation e.g. "%.1f"

Remark

A more detailed description of these properties is contained in the description of the **MainGrid (TVXScale).**

GridTextFormat is only relevant if AutoScaleGrid equals FALSE otherwise the settings of the XML-format file or the standard settings of *TrendViewer* are applied.

MainGrid (TVXScale), AutoScaleGrid

Index

`Short` Index()

Description

returns the index value of the y-scale

Range of values

`0..MaxYScales`

Interactive

`Boolean` Interactive()

Description

switches the interactive modification of the display range of the y-scale on (TRUE) or off (FALSE) by using the mouse

Range of values

TRUE	user can change the display range of the y-scale by using the mouse
FALSE	modification disabled

Remark

the zoom functions of the trend window have no effect on the y-scale in the event of **Interactive** = FALSE

Also see

Zoom (TrendCtrl)

MainGrid

`Double` MainGrid()

Description

the main grid defines the distance between two main grid lines

Range of values

arbitrary value

Remark

The definition of the main division grid is only relevant if **AutoScaleGrid** is set to FALSE otherwise the main division grid is specified via the XML-format file or the standard settings of *TrendViewer*.

A more detailed description of this property is contained in the description of **MainGrid (TVXScale)**.

Also see

SecGrid, DivGrid, GridTextFormat, AutoScaleGrid

MarginFactor

`Double` MarginFactor()

Description

defines the left and right margin factor of the y-scale

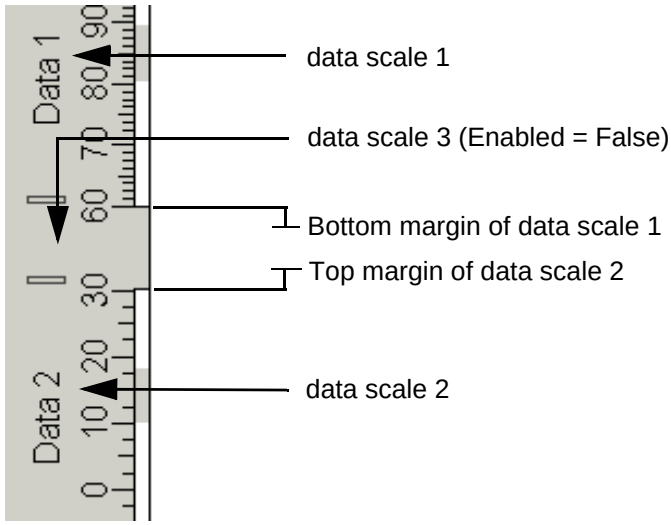
Range of values

arbitrary value

Remark

This property is used amongst others to avoid masking of the adjacent x- or y-scales. This functionality only affects superimposed scales.

The following figure demonstrates this property:




```
Dim yScale1 As TRENDVIEWERLib.TVYScale
Dim yScale2 As TRENDVIEWERLib.TVYScale
Dim yScale3 As TRENDVIEWERLib.TVYScale
Dim Position2 As TRENDVIEWERLib.TVPosition
Dim Position3 As TRENDVIEWERLib.TVPosition

Set yScale1 = TrendViewer1.YScale(0)
Set yScale2 = TrendViewer1.YScale(1)
Set yScale3 = TrendViewer1.YScale(2)
Set Position2 = yScale2.Position
Set Position3 = yScale3.Position

Position2.SetPosition 2, 0, 1, 50
Position3.SetPosition 2, 0, 1, 30

' y-Scale3 visible = False
yScale3.Visible = False

' bottom margin y-Scale 1
yScale1.MarginFactor(3) = 2
yScale1.Priority = 1

' top margin y-Scale 2
```

```
yScale2.MarginFactor(1) = 2
yScale2.Priority = 1
```

Also see

MarginFactor (TrendCtrl), Priority

Mirrored

`Boolean` Mirrored()

Description

mirrors the y-scale

Range of values

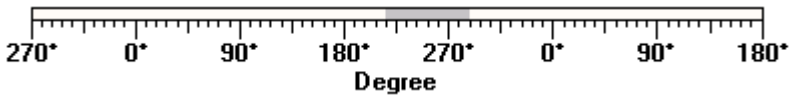
TRUE	scale is mirrored
FALSE	scale is displayed in normal orientation

Modulo360Degrees

`Boolean` Modulo360Degrees()

Description

The degree scale is normally only displayed with arbitrary degree values. When this property is set to TRUE, a modulo function is used to map the degree values to a degree range of 0° and 360° as illustrated in the following example:



Range of values

TRUE	Display between 0° and 360°
FALSE	normal degree scale

Remark

This setting is only relevant to degree scales.

Also see

Type

Priority

`Short` Priority()

Description

Display priority of the y-scale.

Range of values

arbitrary integer value

Remark

The display priority especially affects the margins. Depending on the *Trend-Viewer* settings **MarginFactor** may vary to avoid overlapping of individual scales.

Also see

MarginFactor

Position

`TVPosition` Position()

Description

Reference to a **TVPosition** object. Defines the position of the x-scale.

Range of values

`see` **TVPosition**

Range

TVRange Range()

Description

Reference to a **TVRange**-object. Defines the limits of the different scale ranges.

Range of values

`see` **TVRange**

RangeMoveOnly

`Boolean` RangeMoveOnly()

Description

The display range of the y-scale can only be moved by means of the mouse control.

Range of values

TRUE	moving only enabled
FALSE	moving, enlarging and reducing of the section enabled, default setting

Remark

If **RangeMoveOnly** = TRUE the zoom functions of the trend window have no effect on the y-scale. The display range can be modified via the extended zoom functions of the trend window only by moving the trend section.

Also see

Interactive, **Zoom**

Res

`Double` Res()

Description

Minimum resolution which can be displayed by the scale division.

Default setting (data scale)

Max, Min	200, -200
Res	0,001

Range of values

arbitrary value

Also see

TVRange

SecGrid

`Long` SecGrid()

Description

The secondary grid defines the number of divisions of the main grid (**MainGrid**)

Range of values

Arbitrary integer value

Remark

A more detailed description of this property is contained in the description of **MainGrid (TVXScale)**.

SecGrid is only relevant if AutoScaleGrid equals FALSE otherwise the settings of the XML-format file or the standard settings of *TrendViewer* are applied.

Also see

MainGrid, **DivGrid, AutoScaleGrid, MainGrid (TVXScale)**

Text

BSTR Text()

Description

scale text for the description of the y-scale

Range of values

arbitrary text

Remark

If the space provided for the display of the text is insufficient, the inscription is truncated and marked by three dots.

Also see

ScaleLegends, TextAlign

TextAlign

`Short` TextAlign()

Description

Text alignment of the scale text in the legend

Range of values

`0-taCenter`	centered text
`1-taLeft`	left-aligned text
`2-taRight`	right-aligned text

Also see

Text, ScaleLegends

ToolTipFormat

BSTR ToolTipFormat()

Description

Defines the format of the value display for the ToolTip (**DataWindows**)

Range of values

formatting type in C-notation e.g "%.1f"

Remarks

A more detailed description of these properties is contained in the description of **MainGrid (TVXScale)**.

Also see

DataWindows

Type

`enumScaleTypes` Type()

Description

This property allows to select the data type of the y-scale (degree, decimal or logarithmic display).

Range of values

`0 -stDecimal`	decimal scale
`2 -stDegree`	degree scale
`3 -stLogarithmic`	logarithmic scale

Remark

The time display is disabled for the y-scale.

Also see

enumScaleTypes, **Modulo360Degrees**

Visible

`Boolean` Visible()

Description

This property is used to show or hide the y-scale.

Range of values

TRUE	visible
FALSE	not visible

Remark

This property can only be modified if **AutoScaleVisibility** = FALSE otherwise the trend control activates the scales itself.

Only previously activated y-scales can be switched to visible (see **Enabled**).

Also see

AutoScaleVisibility, **Enabled**, **TVPosition**

XYScaleEnabled

`Boolean` XYScaleEnabled()

Description

Defines whether the individual y-scale can be used as x-basis for a phase diagram, which means that it may be placed directly on top of the x-scale.

Range of values

TRUE	the y-scale may be used as x-basis
FALSE	the y-scale must not be used as x-basis

Remark

This attribute is only active if **Shared (TVXScale)** = TRUE.

Also see

Shared (TVXScale), **TVPosition**

5.1.4 TVRange

Properties of the range object:

Max, Min	maximum, minimum possible scale value (not persistent in time axes)
ActualMax, ActualMin	maximum, minimum of the currently visible scale range
DataMax, DataMin	maximum, minimum of the current range of values of the curves which are assigned to the scale

Max, Min

`Double` Max()

`Double` Min()

Description

Maximum or minimum value which can be assigned to the upper scale range value or the lower scale range value.

Default setting

	Data scale	Time scale
Max, Min	200, -200	Preset value, Preset value

Remark

The limits of the scale display are defined using this property. The user can only define settings within the application which are within the defined ranges.

To change the range of values please read the procedure described in the remarks on **ActualMax, ActualMin**.

Also see

ActualMax, ActualMin

ActualMax, ActualMin

`Double` ActualMax()

`Double` ActualMin()

Description

Limit of the currently displayed scale range. A value or a point of time is entered depending on the scale type.

Range of values

Only values within the maximum display range that are defined by **Max, Min** can be assigned.

Remark

All ranges of values of the scales are checked for consistency and adapted if required before each drawing of the control. It may therefore be necessary to stop a redrawing of the control when changing the ranges of values.

In few cases, changes to the ranges of values could lead to false results by processing the message queue of Windows. This problem can easily be solved by locking the drawing by means of **LockUpdate** = TRUE.

Sample program

```
Dim Range As TRENDVIEWERLib.TVRange
Dim YScale As TRENDVIEWERLib.TVYScale

Set YScale = TrendViewer1.YScale(0)
Set Range = YScale.Range

Range.ActualMin = 10
Range.ActualMin = 100
```

Also see

Max, Min, **DataMax, DataMin**

DataMax, DataMin

`Double` DataMax()

`Double` DataMin()

Description

Maximum of minimum value of the data interval

Range of values

Min	minimum value
Max	maximum value

Remark

As already described in the remarks on **ActualMax, ActualMin**, the ranges of values are checked in every drawing of the control. The data range is only checked if it has already been activated by **DataRangeEnabled (x-scale)** or **DataRangeEnabled (y-scale)**.

Sample program

see **DataRangeEnabled (x-scale)** or **DataRangeEnabled (y-scale)**

Also see

Max, Min, **ActualMax, ActualMin**

5.1.5 TVPosition

Properties of the positioning object:

Group	defines the group to which a scale is to be placed
Col	defines the column to which the scale is to be positioned
Row	defines the row to which the scale is to be positioned
Size	defines the size of the scale in percentage

Group

`Short` Group()

Description

Defines the group which is displayed in the scale.

Range of values

`0-sgTop`	scale is placed on the trend top
`1-sgBottom`	scale is placed on the trend bottom
`2-sgLeft`	scale is placed left
`3-sgRight`	scale is placed right

Remark

A more detailed description of this property is contained in the description of method **SetPosition**.

Also see

SetPosition, Col, Row, Size

Col

`Short` Col()

Description

Defines the column which is displayed in the scale.

Range of values

`0-31`	1st to 32nd scale

Remark

A detailed description of this property is contained in the description of the method **SetPosition**.

Also see

SetPosition, Group, Row, Size

Row

`Short` Row()

Description

Defines the row which is displayed in the scale

Range of values

`0-10`	1st to 8th row

Remark

A more detailed description of this property is contained in the description of the method **SetPosition**.

Also see

SetPosition, Group, Col, Size

Size

`Double` Size()

Description

Defines the size of the scale in percent.

Range of values

`0-`	scale not visible
`>0..100`	percentage size of a scale

Remark

A more detailed description of this property is contained in the description of the method **SetPosition**.

Also see

SetPosition, Group, Col, Row

5.1.6 TVCurve

Properties of the curve object:

AuxLineStyle	reference to a **TVLineStyle-**object (defines the style of an auxiliary line)
AuxValue	define the position of the curve auxiliary lines
AuxVisible	activate curve auxiliary lines (TRUE)
Average	activates (TRUE) or deactivates (FALSE) the mean value display for the compressed curve display
Capture	capture curve(s)
Data	reference to a **TVData**-object (contains all curve points)
Enabled	activates (TRUE) or deactivates (FALSE) the curve
Envelope	activates (TRUE) envelope curve
EnvelopeThreshold	threshold value for the envelope curve (value, starting at which a compressed trend display is plotted)
FillStyle	reference to a **TVFillStyle**-object (defines the FillStyle of a curve)
FloodCurve	reference to a **TVCurve** (references the curve versus which another curve is to be flooded)
FloodStyle	flood style
FloodYValue	value (straight line on the y-scale) versus which another value is to be flooded
Index	instance of the curve
LineStyle	reference to a **TVLineStyle-**object (defines the line style of the curve)
Moveable	curves/curve points can be moved (TRUE)
Priority	priority of the curve
Ruler	reference to a **TVRuler** object (curve ruler) priority of the curve
RulerViewSync	Sichtbarkeit des Kurvenlineal mit der Sichtbarkeit der Kurve koppeln
StateStyle	reference to a **TVStateStyle** object (state of a curve point)
StateStyleCount	status number of the curve points (< 512)
Symbol	number of the curve symbol (enumsSymbols)
SymbolStep	distance between two curve symbols
Text	name of the curve
Type	defines the curve type (e.g. linear, binary etc.)
Visible	shows (TRUE) or hides (FALSE) the curve
XScale	reference to a **TVXScale**-object (assignment of the curve to an x-scale)
YScale	reference to a **TVYScale**-object (assignment of the curve to a scale)

AuxLineStyle

`TVLineStyle` AuxLineStyle()

Description

Reference to a **TVLineStyle** object. Defines the line style of the auxiliary lines of the curve.

Range of values

see **TVLineStyle**

Also see

AuxValue, AuxVisible

AuxValue

`Double` AuxValue(`short` index,)

Description

Defines the position of the two auxiliary lines for the individual curve.

Range of values

The value should be within the range of values of the scale which is assigned to the curve.

Parameter

index	number of the auxiliary line of the curve whose value is to be set or returned

Remark

The auxiliary lines of the curves are among others used to identify a range of values for the curve. Small symbols along the auxiliary lines characterize if the auxiliary line identifies the maximum or the minimum of a range. The symbols (the current curve symbol) are always placed in the range of values next to the auxiliary lines.

Also see

AuxLineStyle, AuxVisible

AuxVisible

`Boolean` AuxVisible(`short` index,)

Description

defines whether the auxiliary line(s) of the individual curve are to be shown

Range of values

TRUE	shows the auxiliary line
FALSE	hides the auxiliary line

Parameter

index	number of the curve auxiliary line which is to be displayed

Also see

AuxValue, AuxLineStyle

Average

`Boolean` Average()

Description

Defines if a mean value curve is to be displayed instead of the normal curve starting at a certain amount of displayed data points.

Range of values

TRUE	mean value curve is displayed starting at a certain amount of data
FALSE	mean value curve is not displayed

Remark

Starting at a certain number of data points of a curve the curve details are no longer visible owing to the restricted resolution of the output device and because data points of e.g. different x-values are mapped to the same x-coordinates in the output context. In this event, it is no longer useful to draw the individual points of the curve.

In order to create useful graphs with large bulks of data the trend control supports two ways for the compressed display of curves:

1) Mean value curve:
   The displayed bulk of data is divided into sections. For each section the mean values of the x and of the y-data are formed and plotted versus each other.
2) Envelope curve:
   The displayed data bulk is divided into sections. For each of these sections the maximum and minimum value is determined for the y-data. The maximum and minimum values are plotted as envelope curve versus the x-mean values. The resulting defined range can be marked by a fill style/color (**FillStyle**).

The data model of the trend control permits the calculation of the required minimum, maximum and mean values without nameable expenditure of time.

The compressed display of the curve data is activated as soon as one of the two compression methods has been activated. Both compression methods can also be displayed simultaneously for a curve.

If large amounts of data are edited with the trend control, curve compression should definitely be selected as otherwise the drawing of the control would take too much time. The number of curve points at which a curve compression should be activated can be set using property **EnvelopeThreshold**.

In order to be able to fully use the interactive control of the trend control, the time required for drawing the trend should be less than one second.

For curve type "binary" (see **Type**) property **Envelope** is deactivated.

Also see

EnvelopeThreshold, Envelope

Data

`TVData` Data()

Description

Reference to a **TVData**-object. Contains all curve values and status values.

Range of values

see **TVData**

Remark

A more detailed description of this property is contained in the description of **TVData**.

Also see

XGain, XOffset, XValue, YGain, YOffset, YValue, State

Enabled

`Boolean` Enabled()

Description

Activates or deactivates the curve.

Range of values

TRUE	curve is activated
FALSE	curve is deactivated

Remark

Only an active curve can be switched to visible by setting **Visible** to TRUE. If curve buttons are visible, only curve buttons for active curves are displayed.

The activation of curves should be handled especially careful in the event that **Shared** (TVXScale) was set to True and the curve data was returned to the trend control via the attribute **Data**. In this event only curve points are accepted, if all active curves were assigned a y-value or an error value.

If trend property **AutoScaleVisibility** equals TRUE, the activation of the y-scales is controlled together with the activation of the curves.

Also see

Shared, **Data**

Envelope

`Boolean` Envelope()

Description

Defines if at a certain amount of displayed data points an envelope curve is to be displayed instead of a normal curve.

Range of values

TRUE	envelope curve is displayed starting at a certain amount of data (**EnvelopeThreshold**)
FALSE	envelope curve is not displayed

Remark

A more detailed description of this property is contained in the description of **Average**.

If an envelope curve is drawn in the background and possible masking e.g. by the fill style must be avoided **Priority** can be used in addition to define the drawing priority of the curve. For curve type "binary" (see **Type**) property **Envelope** is deactivated.

Also see

Average, **EnvelopeThreshold, Priority**

EnvelopeThreshold

`long` EnvelopeThreshold()

Description

defines the threshold value at which an envelope curve is to be displayed instead of the normal curve

Range of values

arbitrary integer value

Remark

a more detailed description of this property is contained in the description of **Average**

Also see

Average, **EnvelopeThreshold**, **Envelope**

FillStyle

`TVFillStyle` FillStyle()

Description

defines the fill style for the display of the individual flood (**TVFillStyle**) or envelope curve (**Envelope)**

Range of values

see **TVFillStyle**

Remark

a more detailed description of this property is contained in the description of **TVFillStyle**.

Also see

Envelope

FloodCurve

`TVCurve` FloodCurve()

Description

reference to a (another) **TVCurve**-object. The reference object corresponds to the curve that is to be flooded.

Range of values

see **TVCurve**

Remark

If the curves contain a large amount of data, the display may be slowed down when flooding is activated and depending on the flood style, especially when zooming or moving a curve.

Also see

FloodYValue, FloodStyle, TVFillStyle

FloodStyle

`Short` FloodStyle()

Description

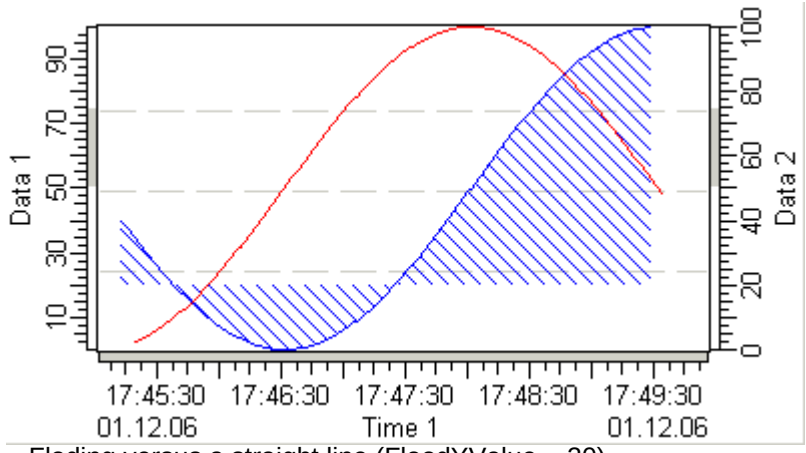
Activates/deactivates flooding of a curve and defines the flood style.

Range of values

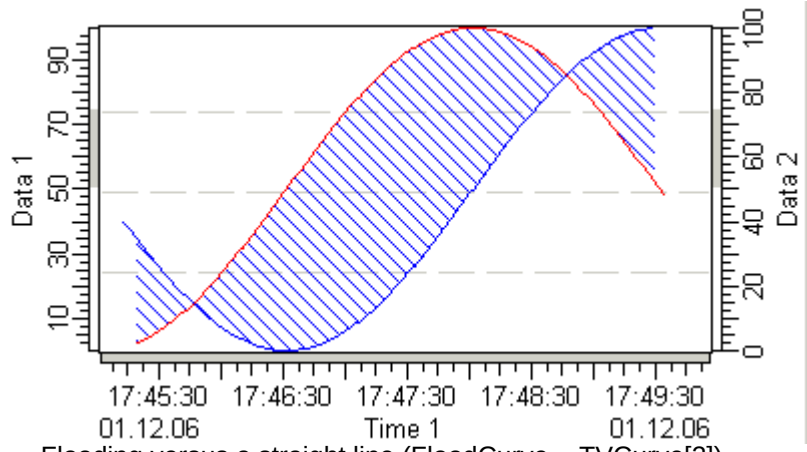
`0-`	=	flooding deactivated
`1-`	=	flooding versus a straight line
`2-`	=	flooding against a curve

Remark

The predefined setting is "0", i.e. flooding is deactivated. Property **Priority** allows to draw flood curves in the foreground or background to avoid masking of other curves.



Floding versus a straight line (FloodYValue = 20)



Flooding versus a straight line (FloodCurve = TVCurve[2])

Also see

FloodCurve, FloodYValue, TVFillStyle, Priority

FloodYValue

`Double` FloodYValue()

Description

defines the y-value of a straight line versus which will be flooded.

Range of values

arbitrary value

Remark

If the value is larger/smaller than the range limit of the assigned y-scale, flooding is carried out up to this range limit.

Also see

FloodCurve, FloodStyle, TVFillStyle

Index

`Short` Index()

Description

Returns the index (curve number).

Range of values

`0..MaxCurves`

LineStyle

`TVLineStyle` LineStyle()

Description

reference to a **TVLineStyle**-object. Defines the line style of a curve.

Range of values

see **TVLineStyle**

Moveable

`Boolean` Moveable()

Description

Defines whether a curve or a single curve point can be moved interactively via mouse control.

Range of values

TRUE	curve/curve point can be moved
FALSE	curve/curve point cannot be moved

Remarks

Depending on whether the mouse pointer is placed on top of a curve point or between two curve points, a corresponding symbol is displayed next to the mouse pointer. It shows if a point or the entire curve can be moved with the right mouse key (see display below).

This property and functionality is restricted or not available depending on the licence model!

Priority

`Short` Priority()

Description

Defines the priority for drawing the curve.

Range of values

`0..MaxCurve`

Remark

This property enables targeted drawing of curves in the foreground or background. It can such be avoided that other curves are masked by the filled surface e.g. when flooding curves or in an envelope curve.

The availability of this property and functionality depends on the license type and may be restricted or not available at all !

Also see

FloodStyle, Average, Envelope

Ruler

`TVRuler` Ruler()

Description

Reference to a **TVRuler** object. A separate ruler can be assigned to each curve.

Range of values

`see` **TVRuler**

RulerViewSync

`Boolean` RulerViewSync()

Description

The visibility of a curve ruler is linked to the visibility of the curve - thus optionally enabling separate or joint control over the ruler and curve display.

Range of values

`see` **LineStyle**

StateStyle

`TVLineStyle` StateStyle(long idx)

Description

defines the state of the index returned with "idx"

Range of values

`see` **TVLineStyle**

Remarks

Each curve can contain a specified fixed number of status values (**StateStyle-Count**). The different status values can be assigned to a curve point via **Data** or via the **TVData** object. When drawing the curve, the status and such the style of the curve line is implemented until another status value arrives.

A more detailed description of this property is contained in the description of **AddYData**.

Also see

StateStyleCount, Data, TVStateStyle, State (TVData), TVLineStyle, AddY-Data

StateStyleCount

`Long` StateStyleCount()

Description

defines the number of the status values for the curve

Range of values

`0..512`

Remark

a more detailed description of this property is contained in the description of **AddYData**.

Also see

StateStyle, TVStateStyle, State (TVData), AddYData

Symbol

`enumSymbols` Symbol()

Description

defines the curve symbol of the individual curve

Range of values

see **enumSymbols**

Also see

enumSymbols, SymbolStep, TVSymbol

SymbolStep

`Long` SymbolStep()

Description

defines the distance between two curve symbols

Range of values

`0`	no symbols are drawn
`>0`	step width in curve points between two symbols

Also see

Symbol

Text

BSTR Text()

Description

description for the individual curve

Range of values

arbitrary text

Remark

The curve description should be kept as short as possible as otherwise the representation as well as the printout of the legend (see **PrintTrend**) would take too much space. The curve description is not shortened automatically contrary to the scale description.

Also see

DataWindows, **PrintTrend**

Type

`enumCurveTypes` Text()

Description

defines the type of a curve (linear, binary, staircase steps)

Range of values

see **enumCurveTypes**

Remark

If a staircase display is selected, ranges of values and mean value curves are also displayed in staircase steps.

Special cases for binary curve display

If binary display is selected for a curve, the following definition applies to the assigned y-scale and all other curves, which are assigned to the same y-scale:

- all curves of the same y-scale automatically become binary curves
- envelope and mean value curve are deactivated

- y-scale zooming deactivated
- curve auxiliary lines are deactivated
- Limit range of the y-scale is automatically set to 0..1

Also see

Average, Envelope, AuxVisible

Visible

`Boolean` Visible(`enumCurves` CurveNumber)

Description

Curve is visible or hidden.

Range of values

TRUE	curve is visible
FALSE	curve is visible

Remark

Only active curves (see **Enabled**) can be visualized. If trend property **AutoScaleVisibility** equals TRUE, the visibility of the curves also controls the visibility of the scales.

Also see

AutoScaleVisibility, Enabled, XScale, YScale

XScale

`TVYScale` XScale()

Description

Assigns one of the two x-scales to the curve.

Range of values

`0-dsXScale1`	1st x-scale
`1-dsXScale2`	2nd x-scale

Remark

The predefined setting assigns the first x-scale to each curve.

Also see

Enabled, **Visible, AutoScaleVisibility, YScale**

YScale

`TVYScale` YScale()

Assigns one of the max. 512 y-scales to a curve.

Range of values (10 standard y-scales)

`0-dsYScale1`	1st y-scale
`1-dsYScale2`	2nd y-scale
`2-dsYScale3`	3rd y-scale
`3-dsYScale4`	4th y-scale
`4-dsYScale5`	5th y-scale
`5-dsYScale6`	6th y-scale
`6-dsYScale7`	7th y-scale
`7-dsYScale8`	8th y-scale
`8-dsYScale9`	9th y-scale
`9-dsYScale10`	10th y-scale

Remark

If trend property **AutoScaleVisibility** equals `TRUE` the visibility of the y-scales is also controlled with the visibility of the curves.

Also see

Enabled, **Visible, AutoScaleVisibility, XScale**

5.1.7 TVData

Properties of the curve value object:

lMaxDataSet	defines the maximum number of curve points which are contained in a data set for each curve
State	status value of the curve point
XGain	gain of the x-value
XOffset	offset of the x-value
XValue	x-value
YGain	gain of the y-value
YOffset	offset of the y-value
YValue	y-value

lMaxDataSet

Long MaxDataSet()

Description

Defines the maximum number of curve points which are contained in a data set for a curve.

Range of values

-1	the number of curve data is unlimited, in this event the application must control the main memory itself to prevent it from memory overflow.
0 and < 2^30-1	the number of the maximum contained curve points is limited to this value

Remark

For the observation of extensive series of measurements it is necessary to restrict the amount of data to a maximum number of curve points. If the number of curve points reaches this limit and more curve points are added, the same number of curve points added is deleted at the beginning of the data set such that the total number of the contained curve points remains almost constant.

If x-scale property **Shared** equals TRUE, the **MaxDataSet** must be the equal for all curves. Therefore, with each modification of **MaxDataSet** the returned value of a curve is also set for all other curves.

If the new value of **MaxDataSet** is smaller than the number of the already existing data sets, an associated number of data points is deleted at the beginning of the data set.

Also see

Shared, TVData

State

Long State()

Description

Defines the status value of the curve point. The value that is specified here references the status line which defines the associated curve in **StateStyle**.

Range of values

arbitrary integer value

Remark

The status and thus the drawing of the line always changes with the value pair, i.e. if the status value changes in the associated curve point (e.g. "B"), the line of the preceding curve point (e.g. "A") is drawn with the status value of curve point "B".

A more detailed description of this property is contained in the description of **AddYData**.

Also see

StateStyle (curve), StateStyleCount (curve), TVStateStyle

XGain

`Double` XGain()

Description

Defines the gain for the x-values of a curve.

Range of values

arbitrary value

Remark

The gain factor is multiplied by all x-values of the associated curve. The standard setting is 1.

A more detailed description of this property is contained in the description of **YOffset**.

Also see

XValue, XOffset, YOffset

XOffset

`Double` XOffset()

Description

Defines the offset point of the curve for the x-values.

Range of values

arbitrary value

Remark

The definition of the offset value allows to move a curve arbitrarily into x-direction to the left or to the right. The default setting is 0.

A more detailed description of this property is contained in **YOffset**.

Also see

XValue, XGain, YOffset

XValue

`Double` XValue()

Description

Returns the x-value of the last data point or sets the x-value of a new data point.

Range of values

arbitrary value

Remark

The curve properties **State**, **XValue** and **YValue** differ from other properties regarding the different use of their read and write features. This behavior results from the special features of the internal data model of the Trend Control.

Remark on the internal data model:

A new measuring point is created successively by means of the properties **State**, **XValue** and **YValue**. If all relevant values of the data point are defined, the data point is saved to intermediate storage. The buffered data is only transferred to the main memory at the next buildup of the trend display.

Creation of a new data point:

A data point must be defined completely before it is returned to intermediate storage and the creation of a new data point can start. As already described in detail in the Remark on trend property **Shared (TVXScale)**, the values of which a data point consists depends on the trend property **Shared**:

For **Shared** = `TRUE` a data point consists of a single x-value and the y-value or status values of all activated curves (see **Enabled (TVCurve)**).

For **Shared** = `FALSE` the data points of the curve are independent of each other. Each data point of a curve is therefore completely defined by its x-value as well as its y-value or its error value.

Advantage over the **AddXData**-, **AddYData**-methods:

Contrary to the **AddXData**-, **AddYData**-methods a programmer must not provide the consistency of the data sets. The logic of the therein described method is relatively complex, but if correctly used the procedure is relatively simple as shown in the subsequent sample program.

Disadvantage over the **AddXData**-, **AddYData**-methods:

If many data are to be returned to the control simultaneously, the OLE-communication brakes the transfer tremendously. In this event, the data should be returned in blocks by means of **AddXData** and **AddYData**.

The following sample program shows for both states of **Shared** how data points are returned to the control by means of **State**, **XValue** and **YValue**. In this sample only curves 1 to 3 are assumed to be active.

Sample program

```
Dim xScale1 As TRENDVIEWERLib.TVXScale
Dim yScale1 As TRENDVIEWERLib.TVYScale
Dim curve As TRENDVIEWERLib.TVCurve
```

```
Dim data As TRENDVIEWERLib.TVData

Set xScale1 = TrendViewer1.xScale(0)
Set yScale1 = TrendViewer1.YScale(0)
Set curve = TrendViewer1.curve(0)
Set data = curve.data

'assuming only Curve 1 to Curve 3 are active
If (xScale1.Shared = True) Then
          'for IdenticalXData = TRUE the x-value must be set
          data.xValue = newXValue

          For i = cnCurve1 To cnCurve3
              Set curve = TrendViewer1.curve(i)
              Set data = curve.data

              data.yValue = newYValue
              data.State = newState
          Next i


'data point completed

Else
          'for IdenticalXData = FALSE the x-value must be _
set
          'for every curve

          For i = cnCurve1 To cnCurve3
             Set curve = TrendViewer1.curve(i)
             Set data = curve.data

             data.xValue = newXValue(i)
             data.yValue = newYValue(i)
             data.State = newState(i)
          Next i
End If
```

Also see

XGain, XOffset, YValue, AddXData, AddYData, State, Shared (TVXScale), Enabled (TVCurve)

YGain

Double YGain()

Description

defines the gain for the y-value of a curve.

Range of values

arbitrary value

Remark

The gain is multiplied by all y-values of the associated curve. The standard setting is 1.

A more detailed description is contained in the description of **YOffset**.

Also see

YValue, YOffset

YOffset

`Double` YOffset()

Description

defines the offset of the value for the y-values

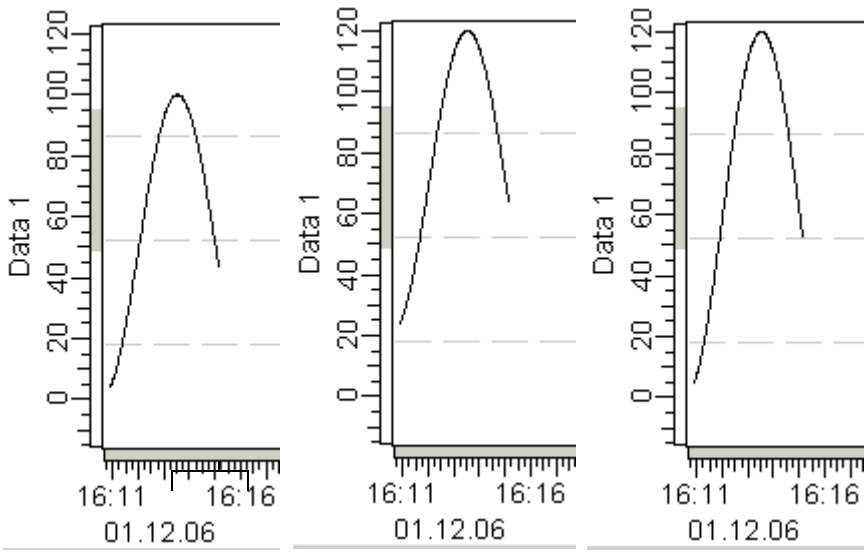
Range of values

arbitrary value

Remark

When defining the offset a curve can be moved arbitrarily into y-direction to the top or the bottom. The standard setting is 0.

The following figure is to illustrate the settings for the gain and the offset:



YOffset = 0
YGain = 1

YOffset = 20
YGain = 1

YOffset = 0
YGain = 1.2

Also see

YValue, YGain

YValue

`Double` YValue()

Description

returns the y-value of the last data point of sets the y-value of a new data point

Range of values

arbitrary value

Remark

A more detailed description of this property is contained in the description of **XValue**.

Also see

YGain, YOffset, XValue

5.1.8 TVLineStyle

Properties of the line style object:

Color	color of the line
Style	line style of the line (see **enumLineStyle**)
Width	width of the line

Color

`OLE_COLOR` Color()

Description

Defines the color of the line to draw.

Range of values

is predefined by the Windows color palette

Also see

Style, Width

Style

`Long` Style()

Description

defines the style of the line

Range of values

see **enumLineStyle**

Remark

It may require considerably more time to draw lines which are wider than stroke width 1 (**Width**).

Also see

Color, Width, enumLineStyle

Width

`Short` Width()

Description

Defines the width of the line to draw.

Range of values

1	a thin line is drawn (standard setting)
2...	a line of the corresponding width is drawn

Remark

It may require considerably more time to draw lines which are wider than stroke width 1.

Also see

Color, Style

5.1.9 TVFillStyle

Properties of the fill style object:

Color	defines the fill style
Style	fill style (see **enumFillStyle**)

Color

`OLE_COLOR` Color()

Description

Defines the fill color

Range of values

is defined by the Windows color palette

Also see

Style

Style

`OLE_COLOR` Style()

Description

defines the fill style for e.g. flood curve, envelope curve.

Range of values

see **enumFillStyle**

Remark

Depending on the selected fill style charting may require considerably more time.

Also see

Color, enumFillStyle

5.1.10 TVStateStyle

Properties of the status object:

LineStyle	reference to a **TVLineStyle**-object.
OnlyCurveColor	uses only the color of the curve for the status (TRUE)
State	status value

LineStyle

`TVLineStyle` LineStyle()

Description

Reference to a **TVLineStyle** object. Defines the line style of the status line to be drawn.

Range of values

`see` **TVLineStyle**

Also see

State, OnlyCurveColor

OnlyCurveColor

`Boolean` OnlyCurveColor()

Description

Only uses the curve color (**LineStyleTVCurve**) (TRUE) to draw the status line. The color of the status line is defined via **LineStyle** as default setting (FALSE.

Range of values

TRUE	status line is drawn in the curve color
FALSE	status line is drawn in the color which is defined in the **LineStyle**

Remark

Usually the color of the status line is **LineStyle**. If curve color **LineStyle (TVCurve**) is to be used generally for drawing, this can be simplified by setting this property.

Also see

State, LineStyle, LineStyle (TVCurve)

State

`Long` State()

Description

This property can be defined using an arbitrary integer value. It references the status value which is entered to **State (TVData)**.

Range of values

arbitrary integer value

Sample program

A more detailed description of this property is contained in the description of **AddYData**.

Also see

State (TVData), LineStyle

5.1.11 TVRuler

Properties of the ruler object:

CustomStyle	activates (TRUE) or deactivates (FALSE) the user-defined ruler style
LineStyle	reference to a **TVLineStyle**-object (line style of the user-defined ruler)
ViewType	display type of the ruler (vertical or horizontal)
Visible	activates (TRUE) or deactivates (FALSE) the interactive control of the ruler

---

CustomStyle

`Boolean` CustomStyle()

Description

activation or deactivation of the user-defined ruler style

Range of values

TRUE	user-defined ruler is activated
FALSE	user-defined ruler is deactivated

The default setting of the TrendControls uses the inverting standard ruler (**CustomStyle** = FALSE). As this ruler cannot be used in all device contexts (printers generally do not support inverting) the Trend-Control also allows the use of a user-defined ruler. Its appearance can be defined in the properties **LineStyle**.

Also see

LineStyle, ViewType, Visible

LineStyle

`TVLineStyle` LineStyle

Description

describes the line style of the user-defined (**CustomStyle**) ruler

Range of values

see **TVLineStyle**

Remark

This property is only evaluated if the user-defined line style has been activated (**CustomStyle** = TRUE).

Also see

ViewType, CustomStyle, TVLineStyle

ViewType

`Short` ViewType()

Description

Defines the display mode (horizontal/vertical) of the ruler.

Range of values

0-	ruler is hidden
1-	vertical ruler
2-	horizontal ruler

Remark

The line style does not depend on the orientation of the Trend Control (see **Rotated**)

Also see

LineStyle, Visible, Rotated

Visible

`Boolean` Visible()

Description

Defines whether the ruler is to be displayed (visible).

Range of values

TRUE	ruler is displayed
FALSE	ruler is not displayed

Also see

ViewType

5.1.12 TVSymbol

Properties of the symbol object:

Value	activates (TRUE) or deactivates (FALSE) the interactive control of the ruler

Value

`Long` Value()

Description

Defines the identification number for the symbol (**enumSymbols**).

Range of values

`see` **enumSymbols**

Also see

enumSymbols

5.2 Methods

The following tabular display contains an overview of all objects of the trend control and methods:

Object groups:

5.2.1 TrendCtrl

Methods of the trend control:

GetTrendWindow	returns the coordinates of the trend window
PrintTrend	prints the trend in an arbitrary print context (all variables must be defined with reference to y-resolution)
SetCursor	changes the mouse cursor
SetTrendWindow	sets the coordinates of the trend window

GetTrendWindow

`Void` GetTrendWindow(`Long*` Left, `Long*` Top, `Long*` Width, `Long*` Height)

Description

determines the position of the trend window in the trend control

Return value

none

Parameter

Left, Top, Width, Height	position of the trend window in pixel coordinates of the Trend-Control

Also see

OwnerDrawUpdateCall, **SetPos**

PrintTrend

`Void` PrintTrend(`OLE_HANDLE` hDC, `Long` Left, `Long` Top, `Long` Width, `Long` Height, `Bool` Rotate, `enumPrintLegendStyles` Legend)

Description

prints a trend of high quality in the target context

Return value

none

Parameter

hDC	handle the target context
Left, Top, Width, Height	rectangle of the target context in logic pixel coordinates to which the trend is to be printed
Rotate	prints the trend in rotated state (TRUE)
Legend	style for an additional legend for the description of curves (see **enumPrintLegendStyles**)

Sample program

```
Dim destleft As Long, destwidth As Long, desttop As Long,_
         destheight As Long

destleft = 0: destwidth = 0: desttop = 0: destheight = 0

'start print
Printer.Print "" '"Trend-Print"

'we want pixels
Printer.ScaleMode = 3

'calculate destination frame
destleft = Printer.ScaleWidth / 10
destwidth = 6 * destleft
desttop = Printer.ScaleHeight / 10
destheight = 6 * desttop

'print destleft, desttop, destwidth, destheight, FALSE, _
         BOTTOM

TrendViewer1.PrintTrend Printer.hDC, destleft, desttop, _
         destwidth, destheight, 0, 3

Printer.EndDoc
```

Also see

LockUpdate, enumPrintLegendStyles

SetCursor

OLE_HANDLE SetCursor(enumTrendCursor CursorIndex,
OLE_HANDLE NewCursor)

Description

changes the mouse cursor

Return value

the handle of the previously set cursor

Parameter

CursorIndex	index for the cursor form that is to be changed
NewCursor	handle of the cursor to reset. If NewCursor equals `NULL`, the standard cursor of the scale control is used.

Remark

Before a new cursor can be assigned, it must be loaded to the system (e.g. using the SDK-function LoadCursor).

Sample program

```
'NewCursor is a global defined picture object
'which holds a cusor image
TrendViewer1.SetCursor tcStandard, NewCursor.Handle
```

Also see

`enumTrendCursor`

SetTrendWindow

`Void` GetTrendWindow(`Long*` Left, `Long*` Top, `Long*` Width, `Long*` Height)

Description

sets the position of the trend window in the trend control

Return value

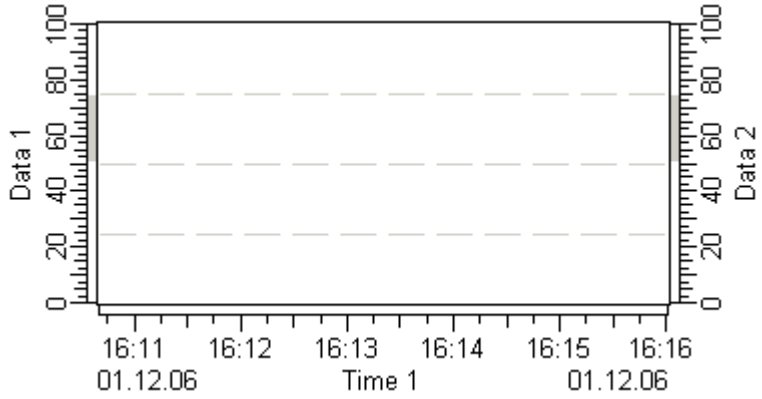
none

Parameter

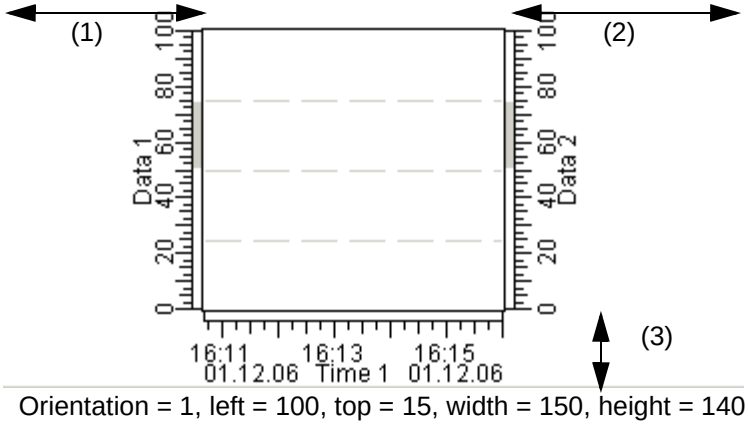
Left, Top, Width, Height	position of the trend window in pixel coordinates of the trend control

Remark

This method can only be called if the alignment (**Orientation**) has previously been set to value 1 (fixed curve window).



Orientation = 0



Orientation = 1, left = 100, top = 15, width = 150, height = 140

(1) range between left side of the control and left side of the curve window

(2) range between right side of the control and right curve window

(3) range between control bottom and curve window bottom

The scale width for all scales is determined by the smallest distance between the control to and the curve window. In above-illustrated example this is distance (3). All scales are drawn using width (3).

Also see

Orientation, GetTrendWindow

5.2.2 TVXScale

Methods of the x-scale:

CalculatePoint	computes a window coordinate for the defined x-scale value
CalculateValue	computes an x-scale value for a defined window coordinate
Capture	captures all curves on the x-coordinate

CalculatePoint

`Boolean` CalculatePoint(`Double` Value, `VARIANT*` pX, `VARIANT*` pY)

Description

Calculates the window coordinate in pixels which corresponds to the returned scale value of the respective x-scale.

Return value

`TRUE`	if the value could be returned correctly
`FALSE`	if the value is outside the maximum display range (defined by **Range**)

Parameter

Value	scale value to be edited
pX, pY	pointer to the variables to which the event is returned

Sample program

```
Dim xScale1 As TRENDVIEWERLib.TVXScale
Dim pX As Variant, pY As Variant
Dim result As Boolean

Set xScale1 = TrendViewer1.XScale(0)
result = xScale1.CalculatePoint(10.0, pX, pY)
```

Also see

CalculateValue, Range

CalculateValue

`Boolean` CalculateValue(`Long` X, `Long` Y, `VARIANT*` pValue)

Description

Calculates the associated scale value of the respective x-scale in pixels to any arbitrary window coordinate.

Return value

`TRUE`	if window coordinate was mapped correctly
`FALSE`	if the event value is outside the maximum display range (defined by **Range**)

Parameter

X, Y	window coordinate in pixels
pValue	pointer to a variable to which the event is returned

Sample program

```
Dim xScale1 As TRENDVIEWERLib.TVXScale
Dim pValue As Variant
Dim result As Boolean

Set xScale1 = TrendViewer1.XScale(0)
result = xScale1.CalculateValue(100, 100, pValue)
```

Also see

CalculatePoint, Range

Capture

`void` Capture()

Enables to "capture" all curves of an x-coordinate that are assigned to the individual x-scale.

Remark

The display of the scale is adapted such that the smallest x-value appears at the left edge of the curve window and the largest x-value of all curves at the right edge of the trend window.

Sample program

```
Dim xScale1 As TRENDVIEWERLib.TVXScale
Dim strValue As String

Set xScale1 = TrendViewer1.XScale(0)

xScale1.Capture
```

Also see

Capture (TVCurve)

5.2.3 TVYScale

Methods of the y-scale objects:

CalculatePoint	computes a window coordinate for the defined x-scale value
CalculateValue	computes an x-scale value for a defined window coordinate
Capture	captures all curves on the y-coordinate

CalculatePoint

`Boolean` CalculatePoint(`Double` Value, `VARIANT*` pX, `VARIANT*` pY)

Description

calculates the window coordinate in pixels which corresponds to the returned scale value of the individual x-scale

Return value

TRUE	if the value could be returned correctly
FALSE	if the value is outside the maximum display range (defined by **Range**)

Parameter

Value	scale value to edit
pX, pY	pointer to the variables to which the event is returned

Sample program

```
Dim yScale1 As TRENDVIEWERLib.TVYScale
Dim pX As Variant, pY As Variant
Dim result As Boolean

Set yScale1 = TrendViewer1.YScale(0)
result = yScale1.CalculatePoint(10.0, pX, pY)
```

Also see

CalculateValue, Range

CalculateValue

`Boolean` CalculateValue(`Long` X, `Long` Y, `VARIANT*` pValue)

Description

Calculates the suitable scale value of the individual x-scale for any arbitrary window coordinate in pixels.

Return value

TRUE	if the windows coordinate was mapped correctly
FALSE	if the result value is outside the maximum display range (defined by **Range**)

Parameter

X, Y	window coordinate in pixel
pValue	pointer to a variable to which the event is returned

Sample program

```
Dim yScale1 As TRENDVIEWERLib.TVYScale
Dim pValue As Variant
Dim result As Boolean
```

```
Set yScale1 = TrendViewer1.YScale(0)
result = yScale1.CalculateValue(100, 100, pValue)
```

Also see

CalculatePoint, Range

Capture

`void` Capture()

Enables to "capture" all curves on the y-coordinate to which the individual y-scales are assigned.

Remark

The display of the scale is adapted such that the smallest y-value appears at the left edge of the curve window and the largest y-value of all curves at the right edge of the trend window.

Sample program

```
Dim yScale1 As TRENDVIEWERLib.TVYScale
Dim strValue As String

Set yScale1 = TrendViewer1.YScale(0)

yScale1.Capture
```

Also see

Capture (TVCurve)

5.2.4 TVPosition

Methods of the positioning object:

GetPosition	returns the current position of a scale
SetPosition	sets the current position of a scale

GetPosition

`Boolean` GetPosition(`VARIANT*` pGroup, `VARIANT*` pColumn, `VARIANT*` pRow, `VARIANT*` pSize)

Description

Determines the logic position of a y-scale.

Return value

`TRUE`	the searched scale was found, the logic position was returned
`FALSE`	the searched scale is not active (see **Enabled TVXScale**, **Enabled TVYScale**).

Parameter

pGroup	pointer to a variable which is to receive the determined scale group (see `enumScaleGroup`)
pColumn	pointer to a variable which is to receive the determined scale column
pRow	pointer to a variable which is to receive the determined scale row
pSize	pointer to a variable which is to receive the determined scale dimension in percentage of the trend window edge

Sample program

see **SetPosition**

Also see

Enabled (TVXScale), **Enabled (TVYScale)**, **SetPosition, enumScaleGroup**

SetPosition

`void` SetPosition(`Short` Group, `Short` `Column`, `Short` Row, `Double` Size)

Description

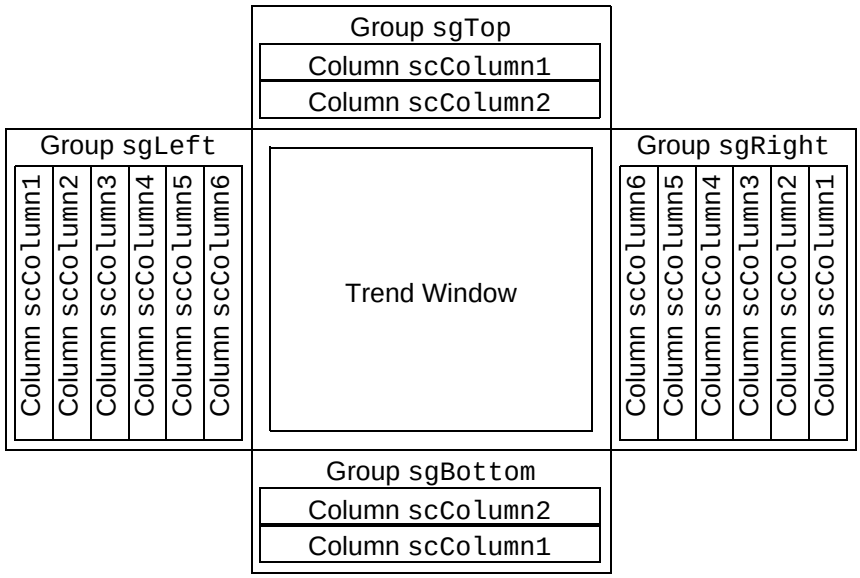
sets the individual scale to a new logic position

Parameter

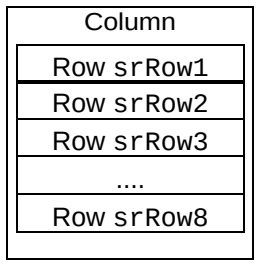
Group	new scale group (see `enumScaleGroup`)
Column	new scale column
Row	new scale row
Size	new scale size in percentage of the trend window edge

Remarks

The logic position of a scale is defined by its group, its column, its row and its relative size compared to the trend window edge. The attributes of the scale position are strictly hierarchical in the above mentioned order. The position of the groups and columns within the control are detailed in the following drawing:



Each column has eight rows except the columns within the group `sgBottom u. sgTop`:



The size of each scale must be between 0% and 100%.

The control checks the scale positions before each drawing and modifies the size and position, if required, to ensure the correct position of all scales. If a scale cannot be placed correctly to its position, it is automatically reset to its standard position (see **YScDefaultGroup**, **YScDefaultColumn**).

The control does not permit any gaps within the individual scale attributes and, if required, therefore places the scales to other positions than the ones defined as demonstrated in the following examples:

Example 1: Only columns 1 and 2 of a group are allocated, another scale should be assigned to column 4, but the scale is actually placed to column 3.

Example 2: If columns 1 to 3 are allocated and if the scale which is the only one that occupies column 2 is removed from its position, the scales in column 3 are automatically moved to column 2.

Example 3: If only rows 1 and 2 are allocated within a column and an additional scale should be placed into row 4, the scale is actually placed to row 3.

Example 4: If rows 1 to 3 are allocated and if the scale in row 2 is removed from its position, scale in row 3 is automatically moved to row 2.

If a scale is placed to a row which is already occupied by another scale, this scale and all scales in the subsequent rows are moved one row to the bottom.

If a scale is placed to an already occupied column and if the new position should be in the first row with a size of 100%, the scales in the respective column and all scales of the following columns are moved by one column such that a new scale can be inserted to the desired column.

When repositioning a scale it should be kept in mind that an already positioned scale is removed before it is repositioned as shows the following example:

Example: Columns 1 to 3 are allocated. The scale that completely occupies column 1 should now be positioned to column 2. When removing the scale, columns 2 and 3 are moved to column 1 and 2. Subsequently, the scale is moved to column 2, whereas column 2 is moved to column 3. Column 1 and column 2 have changed their positions on completion of the procedure.

The following examples show how to place the scales correctly with respect to the position management.

Sample program

```
Dim Position1 As TRENDVIEWERLib.TVPosition
Dim Position2 As TRENDVIEWERLib.TVPosition
Dim yScale1 As TRENDVIEWERLib.TVYScale
Dim yScale2 As TRENDVIEWERLib.TVYScale


Set yScale1 = TrendViewer1.YScale(0)
Set Position1 = yScale1.Position

Set yScale2 = TrendViewer1.YScale(1)
Set Position2 = yScale2.Position

Dim Group As Variant
Dim Row As Variant
Dim Col As Variant
```

```
change Y-scale 1 with Y-Scale 2
Position1.GetPosition Group, Col, Row, Size
Group = sgRight

Position1.SetPosition Group, Row, Col, Size
Group = sgLeft

Position2.SetPosition Group, Row, Col, Size
```

Also see

GetPosition, Group, Col, Row, Size

5.2.5 TVCurve

Methods of the curve object:

Capture	captures the curve on the x- and y-coordinate

Capture

`void` Capture()

Description

Allows to "capture" a curve. All data values (**DataMax, DataMin**) of a curve are displayed in the trend window.

Remarks

The display of the assigned scales is adapted such that the maximum y-value is displayed at the top edge and the minimum y-value at the bottom or the minimum x-value is displayed at the left edge and the maximum x-value at the right edge.

Also see

Data, DataMax, DataMin

5.2.6 TVData

Methods of the curve data object:

AddXData	adds x-values to a curve
AddYData	adds y-values to a curve
DeleteData	deletes a data set
FindFirstXValue	finds a value in the x-data of a curve
FindFirstYValue	finds a value in the y-data of a curve
FindNextXValue	searches for an x-value starting at a defined value
FindNextYValue	searches for an y-value starting at a defined value
GetDataRange	returns the current ranges of values of a curve
GetDataXCount	returns the number of all x-values of a curve
GetDataYCount	returns the number of all x-values of a curve
GetXData	reads a block of x-values
GetYData	reads a block of y-values
ReplaceXData	replaces a block of x-values
ReplaceYData	replaces a block of y-values

AddXData

`Long` AddXDataSA(`VARIANT` xValues, `Long` DataCount)

Description

Returns one or several x-values for the curve points of the individual curve to the Trend Control.

Return value

none

Parameter

xValues	`VARIANT` of type `Array of Double` or `Array of Date`
DataCount	number of x-values that are to be returned

Remarks

To support the different data models that are selected by property **Shared** of the **TVXScale**-object, x- and y-data fields must be returned separately. To ensure the consistency of the data sets, the following rules must be observed.

- For **Shared** = `TRUE` the associated x-data field must only be returned once.
- For **Shared** = `FALSE` the x-data set must also be returned for each curve.

During data transfer no redrawing of the control should be conducted (e.g. by processing the Windows messages to update the windows with progress bars), as the curve data is partially inconsistent. During this time, **LockUpdate** should be set to `TRUE` as a precaution.

Sample program

see **AddYData**

Also see

AddYData, LockUpdate, Shared, TVXScale

AddYData

Long AddYDataSA(VARIANT YValues, VARIANT Status, Boolean bAddStatus, Long DataCount)

Description

Returns one or several y-values for the curve points of the individual curve to the Trend Control.

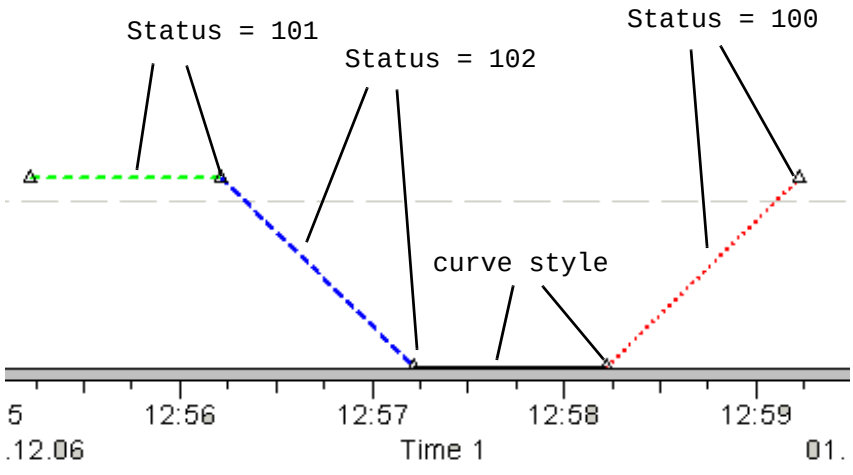
The y-value of a curve is formed from a valid curve value or an error code. If the error code unequals 0, the returned y-value is rejected and replaced by the error code.

Return value

none

Parameter

YValues	VARIANT of type `Array of Double`
Status	VARIANT of type `Array of Long` status values (see **TVStateStyle**)
bAddState	TRUE: the returned status values are integrated into the data set FALSE: the y-values are returned without status values
DataCount	number of the data sets that are to be returned



Sample program

```
Dim Data1 As TRENDVIEWERLib.TVData
Dim Curve1 As TRENDVIEWERLib.TVCurve
Dim xScale1 As TRENDVIEWERLib.TVXScale
Dim Range1 As TRENDVIEWERLib.TVRange

TrendViewer1.LockUpdate = True

' 1. Curveobject
Set Curve1 = TrendViewer1.Curve(0)
Set Data1 = Curve1.data

' define 3 states
Curve1.StateStylesCount = 3

Set Style = Curve1.StateStyle(0)
Style.State = 100

Set LineStyle = Style.LineStyle
LineStyle.Color = RGB(255, 0, 0)
LineStyle.Style = 2
LineStyle.Width = 2

Set Style = Curve1.StateStyle(1)
Style.State = 101


Set LineStyle = Style.LineStyle
LineStyle.Color = RGB(0, 255, 0)
LineStyle.Style = 1
LineStyle.Width = 2

Set Style = Curve1.StateStyle(2)
Style.State = 102

Set LineStyle = Style.LineStyle
LineStyle.Color = RGB(0, 0, 255)
LineStyle.Style = 1
LineStyle.Width = 2

' define range
Set xScale1 = Curve1.XScale
Set Range1 = xScale1.Range
Range1.ActualMax = Now + 300 / 86400
Range1.ActualMin = Now - 300 / 86400

' define Curvedata
Dim xArray(5) As Double
Dim yArray(5) As Double
Dim sArray(5) As Long

n = Now
```

```
yArray(0) = 10
xArray(0) = n + 60 / 86400
sArray(0) = 100

yArray(1) = 10
xArray(1) = n + 120 / 86400
sArray(1) = 101

yArray(2) = 0
xArray(2) = n + 180 / 86400
sArray(2) = 102

' State not defined !
yArray(3) = 0
xArray(3) = n + 240 / 86400
sArray(3) = 4711

yArray(4) = 10
xArray(4) = n + 300 / 86400
sArray(4) = 100

Data1.AddXData xArray, 5
Data1.AddYData yArray, sArray, True, 5

' Curve symbol
Curve1.SymbolStep = 1

TrendViewer1.LockUpdate = False
```

Also see

TVXScale, TVYScale, TVRange, TVCurve, TVStateStyle, TVData, Shared, LockUpdate

---

DeleteData

`Void` DeleteData()

Description

deletes the data set of the curve

Return value

none

Parameter

CurveNumber	number of the curve whose data set is to be deleted

Remarks

If **Shared** = `TRUE`, the x-data set is only deleted when all y-data sets of the active curve are deleted.

The deletion of large data bulks may possibly require considerable time. In this event, the application should display a status bar. Therefore, the data sets should be reduced step by step using curve property **IMaxDataSet** until at last the rest of the data set is deleted by using **DeleteData**. **IMaxDataSet** should subsequently be reset to the old value.

Also see

Shared, **IMaxDataSet**

FindFirstXValue

`Long` FindFirstXValue(`Double` xValue, `Double` Tolerance)

Description

Finds the first x-value of the curve which equals xValue with the defined tolerance or that exceeds xValue for a negative tolerance.

Return value

>0	Number of the found curve point
<=0	not found

Parameter

xValue		value to find
Tolerance	`>=0`:	maximum deviation of the value to find
	`<0`:	finds the first x-value which exceeds xValue

Remarks

No modifications should be made when with the data set by adding more data points as especially for limited data set lengths (see **IMaxDataSet**) data set changes may change the number of the data points between two search procedures.

Sample program

see **FindFirstYValue**

Also see

FindFirstYValue, IMaxDataSet

FindFirstYValue

`Long` FindFirstYValue(`Double` YValue, `Double` Tolerance)

Finds the first y-value of the curve which equals YValue with the defined tolerance or exceeds YValue in the event of a negative tolerance.

Return value

>0	Number of the found curve point
<=0	not found

Parameter

YValue		value to be found
Tolerance	>=0:	maximum deviation of the value to be found
	<0:	finds the first y-value, which exceeds the YValue

Remarks

While working with the data set no modifications should be made by adding more data points, as especially with limited data set lengths (see.**IMaxDataSet**) data set changes may change the number of the data points between search procedures.

Sample program

```
Dim xScale1 As TRENDVIEWERLib.TVXScale
Dim yScale1 As TRENDVIEWERLib.TVYScale
Dim Data1 As TRENDVIEWERLib.TVData
Dim Curve1 As TRENDVIEWERLib.TVCurve
Dim Range1 As TRENDVIEWERLib.TVRange
Dim Ruler1 As TRENDVIEWERLib.TVRuler


'find a data point and position the near this point
'declaring variables

Dim pXData As Variant, xmin As Double, xmax As Double
Dim pYData As Variant
Dim pStatus As Variant
Dim X As Long, Y As Long
Dim Position As Long

Set Curve1 = TrendViewer.Curve(0)
Set Data1 = Curve1.data

'definition outside: Public Position As Long
'search for y-values of curve 2 passing 20.0

If (Position > 0) Then
      Position = Data1.FindNextYValue(Position, _
            20#, -1#)
End If
If (Position < 1) Then
      Position = Data1.FindFirstYValue(20#, -1#)
End If
If (Position < 1) Then
      MsgBox ("No data found !")
Else
```

```
            'we want the full data point

            Data1.GetXData Position, pXData, 1
            Data1.GetYData Position, pYData, pStatus, False,
1

            'if the found value is not in the range of the
actual
            'display we will move the display, such that the
            'data-point is near the center of the trend window
            Set xScale1 = Curve1.XScale
            Set Range1 = xScale1.Range

            xmax = Range1.ActualMax()
            xmin = Range1.ActualMin()

            TrendViewer.LockUpdate = True ' lock display
            Range1.ActualMin() = pXData(0) - _

            (xmax - xmin) / 2#

            Range1.ActualMax() = pXData(0) + _

            (xmax - xmin) / 2#

            TrendViewer.LockUpdate = False ' we want to see it

            'we need the related point to set the  position
            Dim pX As Variant
            Dim pY As Variant
            l = xScale1.CalculatePoint(pXData(0), pX, pY)

            'set the ruler to the related point
            Set Ruler1 = TrendViewer.Ruler
            Ruler1.SetPosition pX, pY
End If
```

Also see

IMaxDataSet, **FindFirstXValue, FindNextYValue, TVRuler, TVData, TVCurve, TVRange, TVYScale, TVXScale**

FindNextXValue

`Long` FindNextXValue(`Long` Start, `Double` XValue, `Double` Tolerance)

Description

Starts searching as of curve point with Id Start for the next x-value of the individual curve, which equals Xvalue with the defined tolerance or exceeds XValue for a negative tolerance.

Finds the first y-value of the curve which equals YValue with the defined tolerance or exceeds YValue in the event of a negative tolerance.

Return value

>0		number of the found curve point
<=0		not found

Parameter

Start		Number of the curve point from which the search is to be started
XValue		value to be searched
Tolerance	>=0:	maximum deviation of the value to be found
	<0:	finds the next x-value which exceeds the XValue

Sample program

see **FindFirstXValue**

Also see

FindFirstXValue

FindNextYValue

`Long` FindNextYValue(`Long` Start, `Double` YValue, `Double` Tolerance)

Description

Searches as of curve point with the number start for the next y-value of the individual curve which equals YValue with the defined tolerance or exceeds YValue for a negative tolerance.

Return value

>0		number of the found curve point
<=0		not found

Parameter

Start		number of the curve point at which the search is to be started
YValue		value to be searched
Tolerance	>=0:	maximum deviation of the value to be searched
	<0:	searches for the next y-value which exceeds the YValue

Sample program

see **FindFirstYValue**

Also see

FindFirstYValue

GetDataRange

`Boolean` GetDataRange(`Double*` pXMin, `Double*` pXMax, `Double*` pYMin, `Double*` pYMax)

Description

determines the ranges of values of the individual curve

Return value

`TRUE`	ranges of values are returned
`FALSE`	the curve possesses no data set and therefore no valid ranges of values

Parameters

pXMin	pointer to a variable which is to receive the minimum of the x-range of values
pXMax	pointer to the variable which is to receive the maximum of the x-range of values
pYMin	pointer to the variable which is to receive the minimum of the y-range of values
pYMax	pointer to the variable to which the maximum of the y-range of values is to be received

Sample program

```
'fitting curve 1 to the trend window

Dim xRange1 As TRENDVIEWERLib.TVRange
Dim yRange1 As TRENDVIEWERLib.TVRange
Dim Curve1 As TRENDVIEWERLib.TVCurve
Dim xScale1 As TRENDVIEWERLib.TVXScale
Dim yScale1 As TRENDVIEWERLib.TVYScale
Dim Data1 As TRENDVIEWERLib.TVData
Dim pXMin As Variant, pXMax As Variant
Dim pYMin As Variant, pYMax As Variant


Set Curve1 = TrendViewer.Curve(0)
Set Data1 = Curve1.data

Set xScale1 = Curve1.XScale
Set yScale1 = Curve1.YScale

Set xRange1 = xScale1.Range
Set yRange1 = yScale1.Range
```

```
Data1.GetDataRange pXMin, pXMax, pYMin, pYMax
TrendViewer.LockUpdate = True

xRange1.ActualMax = pXMax
xRange1.ActualMin = pXMin

yRange1.ActualMax = pYMax
yRange1.ActualMin = pYMin

TrendViewer.LockUpdate = False
```

GetDataXCount

Long GetDataXCount()

Description

determines the number of the x-values of the respective curve

Return value

number of the x-values

Remarks

this method is used for consistency checks of the data sets

Sample program

see **ReplaceYData**

Also see

ReplaceYData

GetDataYCount

Long GetDataYCount()

Description

determines the number of the y-values of the respective curve

Return value

number of y-values

Remarks

this method is used for the consistency check of the data sets

Sample program

see **ReplaceYData**

Also see

ReplaceYData

GetXData

`Long GetXData(Long Start, VARIANT pXValues, Long DataCount)`

Description

reads one or several x-data sets of the respective curve

Return value

number of x-data sets read

Parameter

Start	number of the curve point at which the data are to be read
pXValues	pointer to a data range that is to receive the x-values
DataCount	number of the curve points to be read

Sample program

see **FindFirstYValue**

GetYData

`Long GetYData(Long Start, VARIANT pYValues, VARIANT Status, Boolean bGetStatus, Long DataCount)`

Description

reads one or several data sets of the individual curve

The y-value of a curve is generated from a valid curve value or an error code. If the error code unequals 0, the returned y-value is rejected and replaced by the error code.

Return value

Number of the read data sets

Parameter

Start		number of the curve points from which the data are to be read
pYValues		pointer to the data range that is to receive the y-values
Status		pointer to the data range that is to receive the error values
bGetStatus	TRUE:	the status values are entered to the returned status field
	FALSE:	the returned status field is not modified (any arbitrary pointer can be returned)
DataCount		number of the curve points to be read

Sample program

see **FindFirstYValue**

ReplaceXData

Long ReplaceXData(Long Start, VARIANT xValues, Long DataCount)

Description

replaces the x-values of the curve points of the respective curve

Return value

>0	number of the replaced values
<=0	an error has occurred

Parameter

CurveNumber	number of the curve whose data is to be replaced
Start	starting curve point, from which on the data is to be replaced
xValues	pointer to one or several x-values of the new data points
DataCount	number of the curve points which are to be replaced

Remarks

The methods **ReplaceXData**, **ReplaceYData** can only replace values in existing data sets. No new data sets are added.

If **Shared** = TRUE when exchanging x-values of the data points the x-values of the remaining curves are also exchanged.

During the data transfer a redrawing of the control should not be conducted (e.g. by processing the Windows messages to update windows with status bars) as the curve data are partially inconsistent. During this time, **LockUpdate** should be set to TRUE as a precaution.

Sample program

see **ReplaceYData**

Also see

ReplaceYData, Shared, LockUpdate

ReplaceYData

Long ReplaceYData(Long Start, VARIANT yValues, VARIANT Status, Boolean bAddStatus, Long DataCount)

Description

replaces the y-values of the curve points of the curve

The y-value of a curve is generated from a valid curve value or an error code. If the error code unequals 0, the returned y-value is rejected and replace by the error code.

Return value

>0	number of the replaced values
<=0	an error has occurred

Parameter

Start		starting curve point from which on data is replaced
yValues		pointer to one or several y-values to be replaced
Status		pointer to one or several status values to be replaced
bAddStatus	TRUE:	evaluates the returned states and enters them to the data set
	FALSE:	the replaced y-values contain no status values
DataCount		number of the curve points which are to be replaced

Sample program

```
'replace ydata
'maximum 10000 points declarations

Dim Data1 As TRENDVIEWERLib.TVData
Dim Curve1 As TRENDVIEWERLib.TVCurve
Dim Data2 As TRENDVIEWERLib.TVData
Dim Curve2 As TRENDVIEWERLib.TVCurve
Dim pYData1 As Variant, pYData2 As Variant, pYData3 As
Variant

Dim pStatus1 As Variant, pStatus2 As Variant, pStatus3 As
Variant

Dim MaxCount As Long, Count As Long

MaxCount = 10000

Set Curve1 = TrendViewer.Curve(0)
Set Data1 = Curve1.data

Set Curve2 = TrendViewer.Curve(1)
Set Data2 = Curve2.data

' notice curve data
Set Curve3 = TrendViewer.Curve(511)
Set Data3 = Curve3.data

'calculate data to replace
Count = Data1.GetYDataCount()
If (MaxCount > Count) Then MaxCount = Count
Count = Data2.GetYDataCount()

If (MaxCount > Count) Then MaxCount = Count

TrendViewer.LockUpdate = True
```

```
'get data to replace
Data1.GetYData 0, pYData3, pStatus3, True, MaxCount
Data2.GetYData 0, pYData2, pStatus2, True, MaxCount

'replace data
Data1.ReplaceYData 0, pYData2, pStatus2, True, MaxCount
Data2.ReplaceYData 0, pYData3, pStatus3, True, MaxCount

TrendViewer.LockUpdate = False
```

Also see

TVYScale, TVCurve, TVData, GetDataYCount, LockUpdate

5.2.7 TVRuler

Methods of the ruler object:

GetPosition	returns the current position of the ruler
GetScValue	returns the current ruler with reference to a scale
SetPosition	sets the ruler to a new position
SetScValue	sets the ruler in reference to a scale to a new position

eine neue Position

GetCurveValue

`Boolean` GetPos(`LPDISPATCH` pCurve, `VARIANT*` pValue, `Short` Index)

Description

Returns the exact value of the ruler position on a curve.

Return value

TRUE	Curve value could be determined.
FALSE	Curve value could not be determined.

Parameter

pCurve	Reference to a curve object.
pValue	Pointer to a variable of the current curve value.

GetPosition

`Boolean` GetPos(`VARIANT*` pX, `VARIANT*` pY)

Description

determines the current position of the ruler

Return value

TRUE	ruler position was determined
FALSE	ruler position could not be determined

Parameter

pX	pointer to a variable which is to receive the current x-position of the ruler
pY	pointer to a variable which is to receive the current y-position of the ruler

Remark

If (-1,-1) is returned as position, the ruler is outside the visible range.

Sample program

```
Dim Ruler As TRENDVIEWERLib.TVRuler
Dim pX As Variant, pY As Variant

Set Ruler = TrendViewer.Ruler
pos = Ruler.GetPosition(pX, pY)
```

Also see

SetPosition

GetScValue

`Double` GetScValue(`LPDISPATCH` Scale, `Short` Index)

Description

determines the current position of the ruler on a scale

Return value

`Double`	value on the returned scale

Parameter

Scale	reference to an x- or y-scale (**TVXScale**, **TVYScale**)
Index	not yet used ! (-1)

Remark

If (-1,-1) is returned as position, the ruler is outside the visible range (ruler is not visible).

Sample program

```
Dim Ruler As TRENDVIEWERLib.TVRuler
Dim yScale1 as TRENDVIEWERLib.TVYScale
Dim pX As Variant, pY As Variant
Dim Value as Double

Set yScale1 = TrendViewer.YScale (0)
Set Ruler = TrendViewer.Ruler
Value = Ruler.GetScValue(yScale1, -1)
```

Also see

SetScValue, SetPosition, TVXScale, **TVYScale**

SetPosition

`Boolean` SetPos(`Long` X, `Long` Y)

Description

Sets the ruler to a new position

Return value

TRUE	the ruler is visible after executing the method
FALSE	setting the ruler failed or the ruler was deleted

Parameter

X,Y	new position of the ruler in pixel coordinates of the trend window

Remarks

If the relevant coordinate is outside the trend window, the ruler is not visible.

Sample program

```
Dim Ruler As TRENDVIEWERLib.TVRuler
Set Ruler = TrendViewer.Ruler
pos = Ruler.SetPosition(300, 300)
```

Also see

GetPosition

SetScValue

`Void` SetScValue(`LPDISPATCH` Scale, `Double` Value `Short` Index)

Description

sets the ruler to the position of "Value" on the returned scale

Parameter

Scale	reference to an x- or y-scale (**TVXScale**, **TVYScale**)
Value	value on the scale to which the ruler is to be positioned
Index	not yet used ! (-1)

Remark

If (-1,-1) is returned as position, the ruler is outside the visible range. (Ruler is not visible.)

Sample program

```
Dim Ruler As TRENDVIEWERLib.TVRuler
Dim yScale1 as TRENDVIEWERLib.TVYScale
Dim pX As Variant, pY As Variant
Dim Value as Double
```

```
Set yScale1 = TrendViewer.YScale (0)
Set Ruler = TrendViewer.Ruler
Value = Ruler.SetScValue(yScale1, 50.0, -1)
```

Also see

GetScValue, SetPosition, TVXScale, **TVYScale**

5.2.8 TVCrosshair

Methods of the crosshair object:

GetCurvePos	returns the current position of a curve value
GetScValue	returns the current value of a scale

GetCurvePos

`long` GetCurvePos(`LPDISPATCH` Curve)

Description

Returns the next curve value.

Return value

`>=0`	next curve value
`-1`	curve value not found

Parameter

Curve	TVCurve objekt from the curve.

Sample program

`see` **GetScValue**

GetScValue

`double` GetScValue(`LPDISPATCH` Scale)

Description

Returns the actual Scale value.

Return value

`actual scale value of returned scale`

Parameter

Scale	TVXScale or TVYScale object of the scale from which the value is to be determined

Sample program

```
Private Sub TrendViewer1_CrosshairPosChanged(ByVal pCros-
shair As Object)
    Dim crosshair As TRENDVIEWERLib.TVCrosshair
    Set crosshair = pCrosshair
```

```
    Dim scValue As Double
    Dim pos as long

    scValue = crosshair.GetScValue(TrendViewer1.YScale(0))
    pos = crosshair.GetCurvePos(TrendViewer1.Curve(0))

End Sub
```

5.3 Events

The following tabular display contains an overview of all events of the Trend Control:

Event Groups
5.3.1 Change Event
5.3.2 Drawing Events
5.3.3 Standard-Events

Events

5.3.1 Change Event

CrosshairPosChanged	is sent when the position of the crosshair is changed
CurveAuxChanged	is sent when the position of a curve auxiliary line is changed
CurveChanged	is sent when a curve key is activated
CurveMoved	is sent when the position of a curve point or the position of a curve is changed
CurveRulerPosChanged	is sent when the position of a curve ruler is changed
FontChanged	is sent when the user changes the font
XScActualRangeChange	is sent when the display range of the x-scale is changed
XScActualRangeChanging	is sent when the display range of the x-scale is changing
XScActualRangeChanging	is sent when an attribute of an x-scale is changed
XScalePosChanged	is sent when the position of the x-scale is changed
YScActualRangeChange	is sent when the display range of the y-scale is changed
YScActualRangeChanging	is sent when the display range of the y-scale is changing
YScActualRangeChanging	is sent when an attribute of a y-scale is changed
YScalePosChanged	is sent when the position of the y-scale is changed

---

CrosshairPosChanged

CurveAuxChanged(**TVCrosshair** pCrosshair)

Description

Occurs after the user has changed the position of the crosshair.

Parameter

pCrosshair	associated crosshair object.

Also see

TVCrosshair

CurveAuxChanged

CurveAuxChanged(**TVCurve** Curve)

Description

Occurs after the user has changed the position of the curve auxiliary line.

Parameter

Curve	associated curve object of the auxiliary line

Also see

TVCurve

CurveChanged

CurveChanged(**TVCurve** Curve)

Description

Occurs when the user applies a curve key.

Parameter

Curve	associated curve object

Also see

TVCurve, CurveMoved

CurveMoved

CurveMoved(**TVCurve** Curve)

Description

Occurs when the user has modified the position of a curve point or the position of the curve.

Parameter

Curve	associated curve object

Also see

TVCurve, CurveChanged

CurveRulerPosChanged

CurveRulerPosChanged(**TVRuler** Ruler, **TVCurve** Curve)

Description

Occurs when the user as changed the position of a ruler which is associated to a curve.

Parameter

Ruler	reference to a ruler object
Curve	associated curve object

Also see

TVCurve, **TVRuler**

TVCurve, **LineStyle**

FontChanged

FontChanged(`LPFONTDISP*` pFont)

Description

Is displayed when the user changes the font

Parameter

LPFONTDISP	character set

Also see

Font

XScActualRangeChange

XScActualRangeChange(**TVXScale*** pXScale, `VARIANT` OldMin, `VARIANT` OldMax)

Description

this event is triggered after every change of the current x-scale range

Parameter

pXScale	reference of the x-scale
OldMin	old value of the property (**ActualMax, ActualMin**)
OldMax	old value of the property (**ActualMax, ActualMin**)

Also see

TVXScale, **ActualMax, ActualMin**

XScActualRangeChanging

XScActualRangeChange(**TVXScale*** pXScale, `VARIANT` OldMin, `VARIANT` OldMax)

Description

this event is triggered at every changing of the current x-scale range

Parameter

pXScale	reference of the x-scale
OldMin	old value of the property (**ActualMax, ActualMin**)
OldMax	old value of the property (**ActualMax, ActualMin**)

Also see

TVXScale, **ActualMax, ActualMin**

XScaleChanged

XScaleChanged(**TVXScale*** pXScale)

Description

Occurs when the style or a range of values of the x-scale is changed.

Parameter

pXScale	reference to an x-scale

Remarks

Also see

TVXScale, XScalePosChanged

XScalePosChanged

XScaleChanged(**TVXScale*** pXScale)

Description

Is displayed after the position of the x-scale has changed

Parameter

pXScale	reference to the x-scale

Remarks

Also see

TVXScale, XScActualRangeChanging

YScActualRangeChange

YScActualRangeChange(**TVYScale*** pYScale, `Double` OldMin, `Double` OldMax)

Description

this event is triggered after every change to a current y-scale range

Parameter

pYScale	reference to the y-scale
OldMin	old value of the property
OldMax	old value of the property

Also see

TVYScale, ActualMax, ActualMin, XScActualRangeChange

YScActualRangeChanging

YScActualRangeChange(**TVYScale*** pYScale, `Double` OldMin, `Double` OldMax)

Description

this event is triggered at every change of a current y-scale range

Parameter

pYScale	reference to the y-scale
OldMin	old value of the property
OldMax	old value of the property

Also see

TVYScale, ActualMax, ActualMin, XScActualRangeChange

YScaleChanged

YScaleChanged(**TVYScale*** pYScale)

Description

Occurs when the style of a range of values of the individual y-scale changes.

Parameter

pYScale	reference to the y-scale

Also see

TVYScale, YScalePosChanged

YScalePosChanged

YScaleChanged(**TVYScale*** pYScale)

Description

occurs after the position of the x-scale has changed

Parameter

pYScale	reference to the y-scale

Remarks

Also see

TVYScale, YScActualRangeChanging

5.3.2 Drawing Events

PostDrawCurve	is sent before drawing a curve
PostDrawCurve	is sent before drawing the curves
PreDrawCurve	is sent after drawing a curve
PreDrawCurves	is sent after drawing the curves

PostDrawCurve

PostDrawCurve(0LE_HANDLE hDC,**TVCurve** pCurve Long Left, Long Top, Long Width, Long Height);

Description

Is sent within the drawing procedure after the associated curve has been drawn.

Parameter

hDC	Handle of the device context in which the trend is drawn
pCurve	associated curve
Left, Top, Width, Height	coordinates of the range of the target window (in pixel), that must be redrawn

Remarks

This event can be used to draw the trend window to the foreground. This event is also sent during the printout.

Also see

PostDrawCurve, PreDrawCurve, PreDrawCurves

PostDrawCurves

PostDrawCurves(0LE_HANDLE hDC, Long Left, Long Top, Long Width, Long Height);

Description

Is sent in a drawing procedure after drawing the trend window.

Parameter

hDC	Handle of the device context in which the trend is drawn.
Left, Top, Width, Height	Coordinates of the range of the target window (in Pixel), that has to be redrawn.

Remark

This event can be used to draw the trend window to the foreground. This event is also sent during the printout.

Also see

PostDrawCurve, PreDrawCurve, PreDrawCurves

PreDrawCurve

PreDrawCurve(0LE_HANDLE hDC,**TVCurve** pCurve Long Left, Long Top, Long Width, Long Height);

Description

Is sent during the drawing procedure before the associated curve is plotted.

Parameter

hDC	handle of the device context in which the trend is drawn
pCurve	associated curve
Left, Top, Width, Height	coordinates of the range of the target window (in pixels) that must be redrawn

Remarks

This event can be used for drawing into the background of the trend window. This event is also sent during the printout.

Also see

PostDrawCurve, PostDrawCurve, PreDrawCurves

PreDrawCurves

PreDrawCurves(0LE_HANDLE hDC, Long Left, Long Top, Long Width, Long Height);

Description

Is sent during the drawing procedure after the trend background has been drawn and before the curves are plotted.

Parameter

hDC	handle of the device context in which the trend is drawn
Left, Top, Width, Height	coordinates of the range of the target window (in pixels) that must be redrawn

Remarks

This event can be used for drawing into the background of the trend window. This event is also sent during the printout.

Also see

PostDrawCurve, PostDrawCurve, PreDrawCurve

5.3.3 Standard-Events

Click	Standard mouse click event
DblClick	Standard mouse double-click event
KeyDown	Standard key down event
KeyPress	Standard key press event
KeyUp	Standard key up event
MouseDown	Standard mouse down event
MouseMove	Standard mouse press event
MouseUp	Standard mouse up event

Click

Click()

Description

Occurs when the user presses a mouse key over the control and releases it (Standard Event)

DblClick

DblClick()

Description

Occurs when the user executes a double-click over the control using a mouse key (Standard Event).

KeyDown

KeyDown(`Short*` KeyCode, `Short` Shift)

Occurs when the user clicks on a key while the Control has the focus (Standard Event).

Parameter

KeyCode	Key code of the pressed key
Shift	Status of the Shift keys

KeyPress

KeyPress(`Short*` KeyAscii)

Description

Occurs when the user presses and releases an ANSI-key (Standard Event).

Parameter

KeyAscii	Standard-ANSI-key

KeyUp

KeyUp(Short* KeyCode, Short Shift)

Description

Occurs when the user releases a key while the Control has the focus (Standard Event)

Parameter

KeyCode	Key code of the pressed key
Shift	Status of the Shift keys

MouseDown

MouseDown(Short Button, Short Shift, OLE_XPOS_PIXELS x, OLE_YPOS_PIXELS y)

Description

Occurs when the user presses a mouse key.
(Standard Event)

Parameter

Button	pressed mouse key
Shift	Status of the Shift keys
X,Y	Position of the mouse pointer in the coordinates of the Control

MouseMove

MouseMove(Short Button, Short Shift, OLE_XPOS_PIXELS x, OLE_YPOS_PIXELS y)

Description

Occurs when the user moves the mouse over the control.
(Standard Event)

Parameter

Button	pressed mouse keys
Shift	Status of the shift keys
X,Y	Position of the mouse pointer in the coordinates of the Control

MouseUp

MouseUp(Short Button, Short Shift, OLE_XPOS_PIXELS x,
OLE_YPOS_PIXELS y)

Description

Occurs when the user releases a mouse key that is positioned on the Control (Standard Event).

Parameter

Button	pressed mouse keys
Shift	Status of the Shift keys
X,Y	Position of the mouse pointer in the coordinates of the Control

5.4 Declarations

All constants used by *TrendViewer* are defined by "enum"-types which are detailed in the following section.
"enum"-types:

enumCurveButtons	Enum type for the position of the curve buttons
enumCurveTypes	Enum type for the curve types
enumCurves	Enum type for the curve numbers
enumFillStyle	Enum type for the fill styles
enumLineStyle	Enum type for the line styles
enumMargins	Enum type for the margins
enumPrintLegendStyles	Enum type for the legend positions in the printout
enumRulerStyle	Enum type fur the auxiliairy rulers
enumStripChartsStyle	Enum type for the stripchart styles
enumSymbols	Enum type for the smbol types
enumScaleGroup	Enum type for the group number of the scale positioning
enumScaleTypes	Enum type for the scale types
enumTextAlign	Enum type for the text alignment in the scale legend
enumTrendCursor	Enum type for the different mouse cursors of a trend
enumTrendZoom	Enum type for the zoom variants
enumXScalePosRows	Enum type for the scale rows of the x-scale positions
enumXScalePosColumn	Enum type for the column numbers of the x-scale positons
enumXScales	Enum type for the identifiers of the x-scales
enumYScalePosRows	Enum type for the scale numbers of the y-scale positions
enumYScalePosColumn	v for the column numbers of the y-scale positions
enumYScales	Enum type for the identifiers of the y-scales

enumCurveButtons

defines all possible positions of the curve keys

```
typedef enum
{
 cbNone = 0,       // no curve keys
 cbTop = 1,        // curve keys are on top
 cbScale = 2,      // curve keys are on the y-scales
} enumCurveButtons;
```

enumCurveTypes

defines the different curve types

```
typedef enum
{
 isLinear = 0,        // linear
 isStepsRight = 1,    // steps right
 isCenteredSteps = 2, // centered steps
 isStepsLeft = 3,     // steps left
 isBinary = 3,        // binary
} enumCurveTypes;
```

enumCurves

defines the identification number for the curves

```
typedef enum
{
 cnCurve1 = 0,        // 1stcurve
 cnCurve2 = 1,        // 2nd curve
 cnCurve3 = 2,        // 3rd curve
 cnCurve4 = 3,        // 4th curve
 cnCurve5 = 4,        // 5th curve
 cnCurve6 = 5,        // 6th curve
 cnCurve7 = 6,        // 7th curve
 cnCurve8 = 7,        // 8th curve
 cnCurve9 = 8,        // 9th curve
 cnCurve10 = 9,       // 10th curve
 cnCurveMax = 10,     // 511th curve (last possible curve)
} enumCurves;
```

enumFillStyle

defines the different fill styles of an envelope or flood curve

```
typedef enum
{
 fsSolid = 0,      // solid
 fsHorizontal = 1, // horizontal /* ----- */
 fsVertical = 2,   // vertical   /* ||||| */
 fsFDiagonal = 3,  // diagonal   /* \\\\\ */
 fsBDiagonal = 4,  // diagonal   /* ///// */
 fsCross = 5,      // crossed   /* +++++ */
 fsDiagCross = 6,  // X          /* xxxxx */
 fsTransparent = 7,// Transparent
} enumFillStyle;
```

enumLineStyle

defines the different line styles

```
typedef enum
{
 lsSolid = 0,      // solid          /* _____ */
 lsDash = 1,       // dashed          /* _ _ _ */
 lsDot = 2,        // dotted            /* ..... */
 lsDashDot = 3,    // dash-dotted         /* ._._. */
 lsDashDotDot = 4, // dashed-dotted-dotted   /* ///// */
 lsNull = 5,       // not visible
} enumLineStyle;
```

enumMargins

defines the margins of *TrendViewer*

```
typedef enum
{
 rmLeft = 0,       // left
 rmTop = 1,        // top
 rmRight = 2,      // right
 rmBottom = 3,     // bottom
} enumMargins;
```

enumPrintLegendStyles

defines the identification number to position the legend when printing the trend control

```
typedef enum
{
 plNoLegend = 0,   // no legend
 plLeft = 1,       // legend left from trend
 plTop = 2,        // legend at trend top
 plRight = 3,      // legend right from trend
 plBottom = 4,     // legend at trend bottom
} enumPrintLegendStyles;
```

enumRulerStyle

defines the styles for the auxiliary rulers

```
typedef enum
{
 rsNoRuler         = 0,// no ruler
 rsVerticalRuler   = 1,// vertical ruler
 rsHorizontalRuler = 2,// horizontal ruler
} enumRulerStyle;
```

enumStripChartsStyle

defines all styles for the strip charts

```
typedef enum
{
 scNoStripCharts   = 0,// no strip charts
 scLeftSide        = 1,// strip charts on the left side
 scRigthSide       = 2,// strip charts on the right side
 scBothSides       = 3,// strip charts on the both sides
} enumStripChartsStyle;
```

enumSymbols

defines the identification number for the curve symbols

```
typedef enum
{
 tcTriangle = 0,   // triangle
 tcQuad = 1,       // quadrat
 tcRhomb = 2,      // rhomb
 tcOct = 3,        // octahedron
 tcCross = 4,      // cross
 tcX = 5,          // X
 tcDCross = 6,     // double cross
 tcQuadCross = 7,  // quad cross
 tcQuadX = 8,      // quad with x
 tcRhombCross = 9, // rhom with cross
 tcOctCross = 10,  // octahedron with cross
 tcOctX = 11,      // octahedron with X
} enumSymbols;
```

enumScaleGroup

defines the possible scale groups to which the data scales in the trend control can be assigned

```
typedef enum
{
 sgTop    = 0,      // scale is on the trend top
 sgBottom = 1,      // scale is on the trend bottom
 sgLeft   = 2,      // scale is on the left
 sgRight  = 3,      // scale is on the right
} enumScaleGroup;
```

enumScaleTypes

defines the possible scale groups to which the data scales in the trend control can be assigned

```
typedef enum
{
 stDecimal = 0,     // decimal scale
 stTime   = 1,      // time scale (only x-scale)
 stDegree = 2,      // degree scale
 stLogarithmic = 3, // logarithmic scale
} enumScaleTypes;
```

enumTextAlign

defines the text alignment in the scale legend

```
typedef enum
{
 taCenter = 0,     // centered
 taLeft = 1,       // left
 taRight = 2,      // right
} enumTextAlign;
```

enumTrendCursor

Identification for the mouse cursor of the trend:

```
typedef enum
{
 tcStandard = 0,   // Standard-Cursor(scales)
 tcTVRuler = 1,    // Cursor for the vertical ruler
 tcTHRuler = 2,    // Cursor for the horizontal ruler
 tcScLButton = 3,  // left mouse button cursor(scales)
 tcScRButton = 4,  // right mouse button cursor(scales)
 tcScMButton = 5,  // center mouse button cursor(scales)
 tcTLButton = 6,   // left mouse button cursor(Trend)
 tcTRButton = 7,   // right mouse button cursor(Trend)
 tcTMButton = 8,   // center mouse button cursor(Trend)
} enumTrendCursor;
```

enumTrendZoom

defines the identification number for zooming in the trend window

```
typedef enum
{
 tzNoZoom = 0,     // no zoom
 tzZoom = 1,       // zoom(moving,compressing,
                      spreading)
 tzLassoZoom = 2,  // lasso zoom
} enumTrendZoom;
```

enumXScalePosRows

defines the scale rows in a x-scale column

```
typedef enum
{
 srXRow1  = 0,       // 1st row
 srXRow2  = 1,       // 2nd row
} enumXScalePosRows;
```

enumXScalePosColumn

defines the columns in the x-scale-groups

```
typedef enum
{
 scXColumn1= 0,     // 1st column
 scXColumn2= 1,     // for later implementation
} enumXScalePosColumn;
```

enumXScales

defines the identifiers for the x-scales in the trend

```
typedef enum
{
 dsXScale1= 0,      // 1st X-scale
 dsXScale2= 1,      // 2nd X-scale
} enumXScales;
```

enumYScalePosRows

defines the scale rows in a y-scale column

```
typedef enum
{
 srYRow1  = 0,       // 1st row
 srYRow2  = 1,       // 2nd row
 srYRow3  = 1,       // 3rd row
 srYRow4  = 1,       // 4th row
 srYRow5  = 1,       // 5th row
 srYRow6  = 1,       // 6th row
 srYRow7  = 1,       // 7th row
 srYRow8  = 1,       // 8th row
 srYRow9  = 1,       // 9th row
 srYRow10 = 1,       // 10th row
} enumYScalePosRows;
```

enumYScalePosColumn

defines the column within the y-scale groups

```
typedef enum
{
 scYColumn1= 0,      // 1st column
 scYColumn2= 1,      // 2nd column
 scYColumn3= 1,      // 3rd column
 scYColumn4= 1,      // 4th column
 scYColumn5= 1,      // 5th column
 scYColumn6= 1,      // 6th column
 scYColumn7= 1,      // 7th column
 scYColumn8= 1,      // 8th column
 scYColumn9= 1,      // 9th column
 scYColumn10= 1,     // 10th column
} enumYScalePosColumn;
```

enumYScales

defines the identifiers for the y-scales in the trend

```
typedef enum
{
 dsYScale1= 0,       // 1st Y-Scale
 dsYScale1= 1,       // 2nd Y-Scale
 dsYScale1= 1,       // 3rd Y-Scale
 dsYScale1= 1,       // 4th Y-Scale
 dsYScale1= 1,       // 5th Y-Scale
 dsYScale1= 1,       // 6th Y-Scale
 dsYScale1= 1,       // 7th Y-Scale
 dsYScale1= 1,       // 8th Y-Scale
 dsYScale1= 1,       // 9th Y-Scale
 dsYScale1= 1,       // 10th Y-Scale
} enumYScales;
```

5.5 RulerWindow

5.5.1 Properties

General properties of the ruler window:

Dispatch	Links the window to TrendViewer

Dispatch

`IDISPATCH*` Dispatch

Description

This property links the ruler window to TrendViewer.

Range of values

`IDISPATCH`	TrendviewerCtrl

Sample program

```
Tv3_LegendWnd1.Dispatch = m_Trend
```

5.5.2 Methods

Methods of the ruler window :

Update	updates the legend window

Update

`void` Update

Description

This method allows to manually update the ruler window.

Remark

If the ruler window is linked to TrendViewer via property Dispatch, updates are normally automatically carried out via corresponding events.

5.5.3 Events

Events of the ruler window :

XScFormatText	Formatting text to display the x-scale values.
YScFormatText	Formatting text to display the y-scale values.
CurveFormatText	Formatting text to display the curve values.

XScFormatText

XScFormatText(`Short` ScaleId, `Double` Value, `BSTR*` ValueText)

Beschreibung

Is fired when the Control generates text to display the Values from the x-scale.

Parameter

ScaleId	Number of the X-Scale
Value	Value, that was formatted
ValueText	Pointer to a text that contains the formatted numeric value.

Also see

YScFormatText, **CurveFormatText**

YScFormatText

YScFormatText(`Short` ScaleId, `Double` Value, `BSTR*` ValueText)

Beschreibung

Is fired when the Control generates text to display the Values from the y-scale.

Parameter

ScaleId	Number of theY-Scale
Value	Value, that was formatted
ValueText	Pointer to a text that contains the formatted numeric value.

Also see

XScFormatText, **CurveFormatText**

CurveFormatText

CurveFormatText(`Short` ScaleId, `Double` Value, `BSTR*` ValueText)

Beschreibung

Is fired when the Control generates text to display the Values from the curves.

Parameter

CurveId	Number of the Curve
Value	Value, that was formatted
ValueText	Pointer to a text that contains the formatted numeric value.

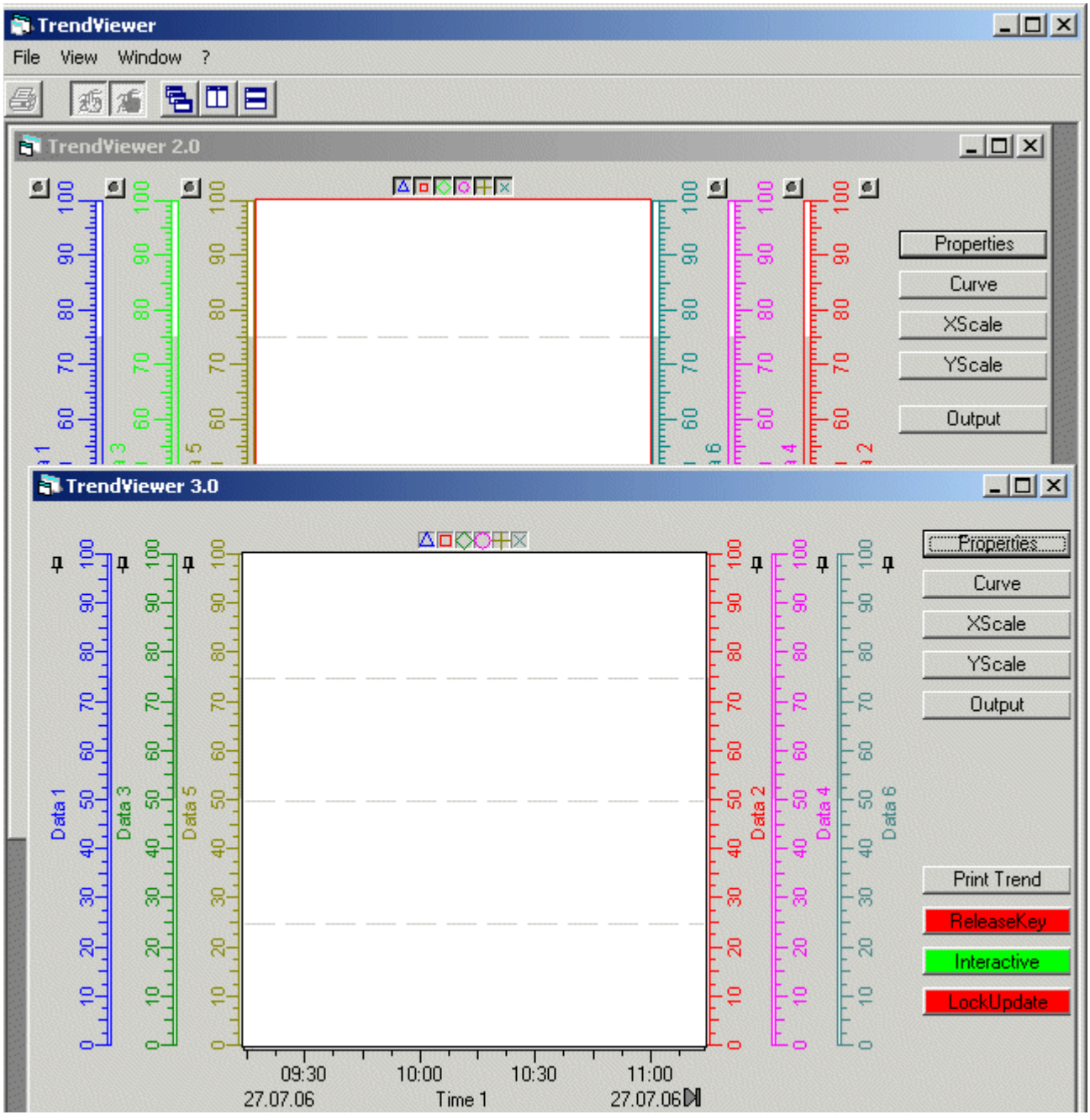
Also see

XScFormatText, **YScFormatText**

6. Test routine

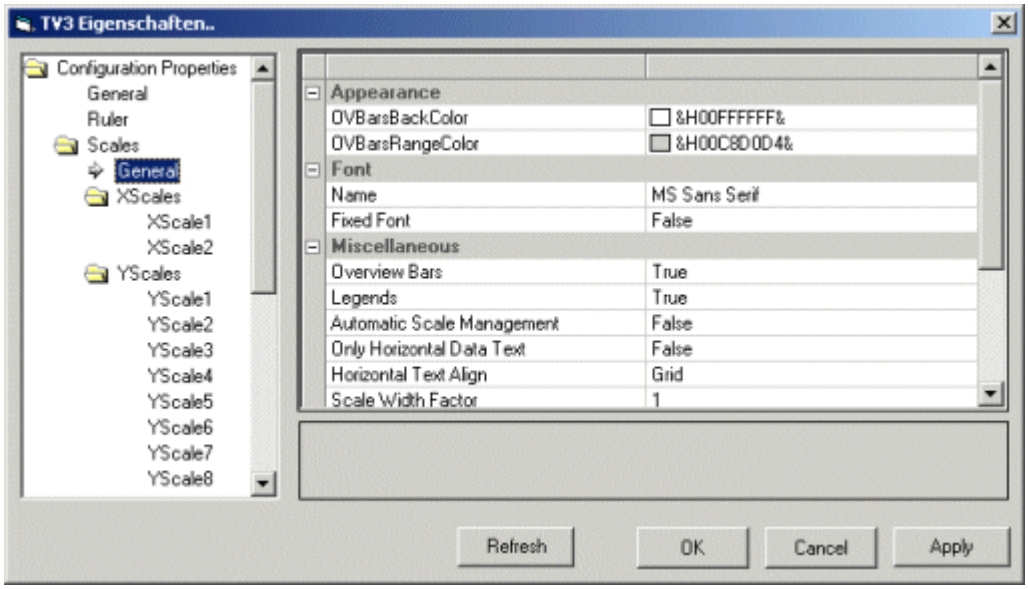
The test routine contained in the *TrendViewer* software package helps test the features and use of the interactive objects as well as enhance a comparison between the current and the preceding release.

All properties and functions provided in both releases can be compared directly in their effect.



6.1 Properties

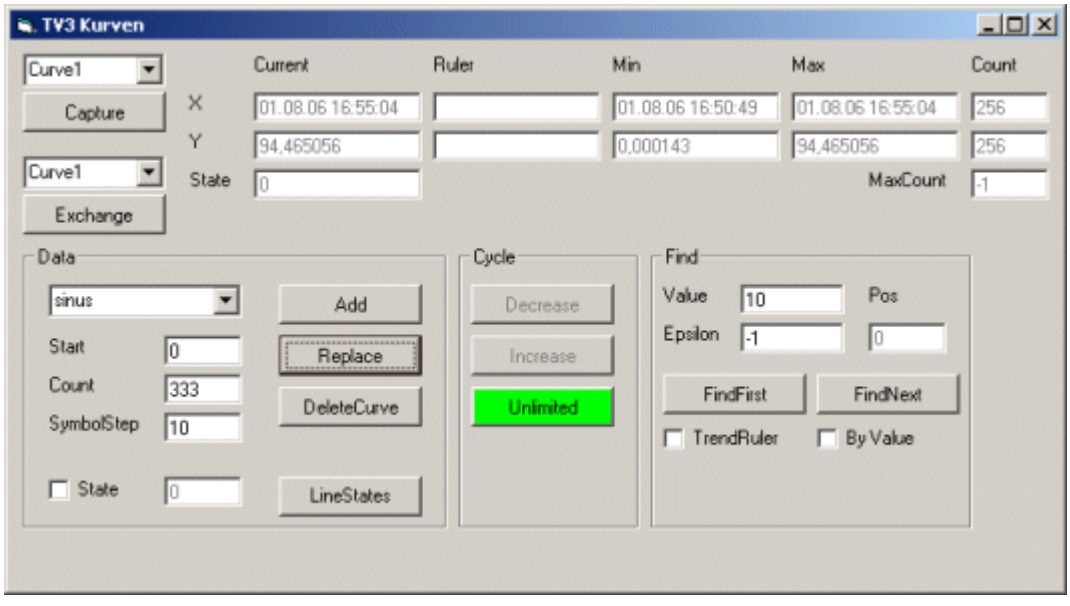
A form is called via function **Properties**. In this form, general properties of *TrendViewer* as well as properties of its objects can be set. Especially, the combination of properties which lead to certain effects can be checked with the test routine.



10 y-scales and 10 curves can be set via the properties. All additional scales and curves can be via reached via the program interface.

6.1.1 Curves

Before the properties of curves are visible, the curves must first be generated. Key **curves** displays a form for the generation of curves.



Curves with different functions can be generated in field **Data**. The number of the curve points can be predefined, their marking with symbols and their status value. Button **Add** is used with an empty curve buffer for the generation of a number of values predefined in **Count**. If the curve buffer is not empty, additional values are added using **Add**.

The graphic display of a status value is set in a special form which is opened via LineStates. Within a curve it is such possible to switch between several statuses, which are activated via the Checkbox and are defined by entering the status value.

Field **Cycle** specifies whether the curve points are to lie in a rotating buffer which is limited in length. If the buffer is not limited (unlimited), a high consumption of memory may be required in online-curves. The length of the curve buffer can be increased or reduced. The currently set variables can be viewed in the fields **Min**, **Max** and **Count**.

The built-in search function is demonstrated in field **Find.** *TrendViewer* can find curve points with predefined value and adjustable accuracy and displays them in a trend window by means of a ruler.

6.1.2 Scales

Keys **XScale** and **YScale** display a form in which the initial values and the final values of the scales are set and the capture function can be tested.

6.1.3 Output

Events which are activated with interactive objects can be drawn in a protocol window, which is opened with key **Output**.

6.1.4 Print Trend

Key Print Trend outputs the currently selected print-layout of *TrendViewer* on a standard printer. The readability of a diagram - especially with black and white printers - can such be tested.

6.1.5 Set interactions

The two keys **ReleaseKey** and **Interactive** switch the functions of the mouse. When Interactive is deactivated, it is impossible to change *TrendViewer* interactively. If **ReleaseKey** is activated the positions of the scales are changed via the mouse, while the zoom function is executed in deactivated status.