

\$#KEditor

Description

Introduction

The Chart Editor is designed to help you quickly modify Charts.

To get help on any Topic in the Chart Editor, select the {bmc helpbutton.bmp} button at the top righthand side of the Editor window and drag it onto the Topic in question. TeeChart Pro help will show you the runtime property or method associated with the feature.

Editor design

There are 2 principal sections to the Chart editor, Chart parameters and the Series parameters, which are separated as 2 tabs of the Chart Editor.

{bmc Editortabs.bmp}

Chart pages

You may define overall Chart display parameters.

Series Pages

Series pages will contain parameters dependant on the series type concerned. The Combobox at the top of the Series tab page shows which series you are editing.

\$

\$Editor

#About_Chart_Editor

KEditor

\$AddSeries

`$#KAdd Series`

Description

Add a new Series to the Chart.

[S](#)

`#AddSeries_method`

`KAddSeries`

`$AdjustFrame`

`$##KKAdjustFrame`

Description

The AdjustFrame property controls if Chart Title and Foot frames will be resized to full Chart dimensions or to the title text width.

[S](#)

`#AdjustFrame_property`

`KAdjustFrame`

`$Alignment`

`$##KAlignment`

Description

The Alignment property defines the Legend position.

Legend can be currently placed at Top, Left, Right and Bottom side of Chart.

[S](#)

`#Alignment_property`

`KAlignment`

`$Alignment`

`$#KAlignment`

Description

The **Alignment** property decides the alignment of the text of a TChartShape.

[S](#)

`#Alignment_property__TChartShape`

`KAlignment`

`$Alignment`

`$##KKAlignment`

Description

The Alignment property controls how Chart Title and Foot text will be aligned within Chart rectangle.

A TChartTitle can optionally be surrounded by a Frame.

[S](#)

`#Alignment_property__TChartTitle`

`KAlignment`

`SEnable Zoom`

\$[#]_#^KEnable Zoom

Description

Enable/Disable Zoom on mouse drag.

[S](#)

[#]AllowZoom_property

^KEnable Zoom

^{\$}Angle

\$[#]_#^KAngle

Description

The Angle property determines the Axis Title rotation in degree units.

Valid values are 0, 90, 180, 270 and 360 degrees.

[S](#)

#Angle_property
^KAngle
\$Animated

`$#KAnimated`

Description

The **AnimatedZoom** property determines if Zoom will be performed directly or it will be displayed as an animated sequence of zooms. The **AnimatedZoomSteps** property controls the number of zoom steps. **AnimatedZoom** can be useful when there are a lot of points.

[S](#)

`#AnimatedZoom_property`

`KAnimated`

`$AnimatedSteps`

\$[#]_#^KAnimated Steps

Description

The **AnimatedZoomSteps** property determines the number of steps of the animated zooming sequence. Big number of steps can delay zooming. The **AnimatedZoom** property should be True.

[S](#)

[#]**AnimatedZoomSteps_property**

^K**AnimatedSteps**

^{\$}**AreaBrush**

\$[#]_#^KArea Brush

Description

Fill Brush characteristics of the Area of an Area Series

[S](#)

[#]AreaBrush_property
^KAreaBrush
^{\$}AreaPen

`$#KArea Pen`

Description

Pen used to draw Lines in Area Series.

[S](#)

`#AreaLinesPen_property`

`KAreaPen`

`$ArrowHeight`

\$[#]_#^KArrow Height

Description

The **ArrowHeight** property determines the vertical arrow head size in pixels for an **ArrowSeries**.

[S](#)

#ArrowHeight_property

^KArrowHeight

^SArrowWidth

\$[#]_#^KArrow Width

Description

The **ArrowWidth** property determines the horizontal arrow head size in pixels for an **ArrowSeries**.

[S](#)

[#]**ArrowWidth_property**
^K**ArrowWidth**
^{\$}**AutoMarkPosition**

\$[#]_#^KAuto Mark Position

Description

The **AutoMarkPosition** property controls if Marks on TBarSeries will be repositioned if there's the possibility of Mark overlapping. Marks are displaced to the top of the Bars to minimize the overlapping effect of Marks with long text or big font sizes.

[S](#)

[#]AutoMarkPosition_property

^KAutoMarkPosition

^SAutomatic

\$[#]_#^KAutomatic

Description

When disabled, Maximum and Minimum Axis values may be manually set.

[S](#)

[#]Automatic_property
^KAutomatic
^{\$}AutomaticMaximum

\$[#]_#^KAutomatic Maximum

Description

If **AxisAutomatic** is disabled setting **AutomaticMaximum** to True will automatically determine the **Axis Maximum** only. Minimum will require to be manually set.

[S](#)

[#]**AutomaticMaximum_property**

^K**AutomaticMaximum**

^S**AutomaticMinimum**

\$[#]_#^KAutomatic Minimum

Description

If **AxisAutomatic** is disabled setting **AutomaticMinimum** to True will automatically determine the **Axis Minimum** only. **Maximum** will require to be manually set.

[S](#)

[#]**AutomaticMinimum_property**

^K**AutomaticMinimum**

^{\$}**AxisPen**

`$##KKAxis Pen`

Description

The Axis property determines the kind of pen used to draw the Axis major lines. These are the lines which go from Axis Minimum to Axis Maximum screen positions.

[S](#)

`#Axis_property`

`KAxisPen`

`$AxisVisible`

\$[#]_#^KKVisible Axis

Description

Enables/disables Axis visible for all Axes at once.

[S](#)

[#]AxisVisible_property

^KAxisVisible

^{\$}BackImageLoad

`$#KBack Image Load`

Description

Back Chart Panel image. May be .bmp, .wmf, .emf or .ico.

[S](#)

`#BackImage_property`

`KBackImageLoad`

`$BackImageInside`

\$[#]_#^KBack Image Inside

Description

Restrict Backimage display to inside Chart Rectangle only.

[S](#)

#BackImageInside_Property

^KBackImageInside

^SBackImageMode

\$[#]_#^KKBack Image Mode

Description

The **BackImageMode** property determines how will the background bitmap be displayed.
Eg. Stretched, Tiled or Centered.

[S](#)

[#]**BackImageMode_property**

^K**BackImageMode**

^{\$}**BarBrush**

`$##KKBar Brush`

Description

Brush used to Fill Bars of a BarSeries.

[S](#)

`#BarBrush_property`

`KBarBrush`

`$BarPen`

`$##KKBar Pen`

Description

Pen used to Draw Lines of a Bar Series.

[S](#)

`#BarPen_property`

`KBarPen`

`$BarStyle`

\$[#]_#^KKBar Style

Description

Choose the Type of Bar for the Bar Series. Eg. Rectangle, Pyramid or Ellipse.

[S](#)

#BarStyle_property

^KBarStyle

\$BarWidthPercent

$\$^{\#}_{\#}^K$ KBar Width Percent

Description

The BarWidthPercent property determines the percent of total Bar width used. Setting BarWidthPercent := 100 makes joined Bars.

[S](#)

$\#$ BarWidthPercent_property
 $\#^K$ BarWidthPercent
 $\$$ Brush

`$#KBrush`

Description

Brush used to fill a Series Pointer

[S](#)

`#Brush_property`
`KBrush`
`$Brush`

`$##KKBrush`

Description

Brush used to fill a Shape

[S](#)

`#Brush_property__TChartShape`
`KBrush`
`SCaption`

`$#KKCaption`

Description

The Caption property defines the string of text used to draw near to each Chart Axis. When empty, no Title is displayed.

[S](#)

`#Caption_property`

`KCaption`

`$ChangeSeriesType`

\$[#]_#^KKChange Series Type

Description

Changes the Series Type to an available Type offered by the TeeChart Gallery.

[S](#)

[#]ChangeSeriesType_method

^KChangeSeriesType

^SScroll

`$#KScroll`

Description

TChart and TDBChart components allow, by default, runtime Scrolling. Users need to drag the mouse while holding the right mouse button pressed.

[S](#)

`#Chart_Scrolling_and_Panning`

`KScroll`

`$Chart3DPercent`

\$[#]_#^K KChart 3D Percent

Description

The **Chart3DPercent** property indicates the size ratio between Chart dimensions and Chart depth when **Chart.View3D** is True. You can specify a percent number from 1 to 100.

[S](#)

[#]Chart3DPercent_Property

^KChart3DPercent

^{\$}Aspect

`$#KAspect`

Description

The **Chart3DPercent** property indicates the size ratio between Chart dimensions and Chart depth when **Chart.View3D** is True. You can specify a percent number from 1 to 100.

[S](#)

`#Chart3DPercentAsp_Property`

`KAspect`

`$ShowPreview`

`$#KShow Preview`

Description

Shows a customisable PrintPreview of Chart.

[S](#)

`#ChartPreview_method`

`KShowPreview`

`$CircleBackColor`

`$##KKCircle Back Color`

Description

The CircleBackColor determines the color to fill the ellipse.

[S](#)

`#CircleBackColor_property`

`KKCircleBackColor`

`SCircle`

$\$^{\#}_{\#}K$ Circled

Description

The **Circled** property defines if a **CircledSeries** such as **TPieSeries** will be drawn elliptically or with same X and Y radius (circle).

[S](#)

$\#$ **Circled_property**

K **Circled**

$\$$ **ClipPoints**

`$#KClip Points`

Description

ClipPoints boolean property toggles the drawing of **Series** points within **Chart** boundaries, preventing other **Chart** regions from being overwritten.

[S](#)

`#ClipPoints_Property`

`KClipPoints`

`$PanelColor`

$\$^{\#}_{\#}^K$ KPanel Color

Description

Background colour of the entire TeeChart Panel.

[S](#)

$\#$ Color_property

K PanelColor

$\$$ Legend Color

`$#KLegend Color`

Description

Background colour of the Legend box.

[S](#)

`#Color_property__TChartLegend`

`KLegend Color`

`$Pen Color`

`$#KPen Color`

Description

Colour of the Chart Pen used for drawing lines.

`$`

`#Color_property__TChartPen`

`KPen Color`

`$Wall Color`

`$#K KWall Color`

Description

Colour of the Selected Chart Wall.

[S](#)

`#Color_property__TChartWall`
`KWall Color`
`$ColorEachPoint`

\$[#]_#^KColor Each Point

Description

When enabled, assigns a different, TeeChart selected colour, to each Series point.

[S](#)

[#]ColorEachPoint_property

^KColorEachPoint

^SColorWidth

\$[#]_#^KColor Width

Description

The width of the Colour box representing the Series (or Series point) in the Legend.

[S](#)

[#]ColorWidth_property

^KColorWidth

^SConnectingPen

`$##KKConnecting Pen`

Description

Pen porperties for the Line that connects Gantt Bars.

[S](#)

`#ConnectingPen_property`

`KConnectingPen`

`$Dark3D`

`$##KKDark3D`

Description

Dark shades the sides of 3D Objects.

[S](#)

`#Dark3D_property`

`KDark3D`

`$Direction`

`$##KKDirection`

Description

The Direction property specifies the direction the gradient fill will be applied.

[S](#)

`#Direction_property`

`KDirection`

`$DividingLines`

\$[#]_#^KKDividing Lines

Description

The **DividingLines** property specifies the Pen attributes used to draw lines separating Legend's items.

Lines are drawn horizontally for Left or Right aligned Legend and vertically for Top or Bottom Legend alignments.

[S](#)

[#]**DividingLines_property**

^K**DividingLines**

^{\$}**Draw3D**

`$##KDraw3D`

Description

The **Draw3D** property controls if **Series Pointers** will be drawn in a 3D way or not. Currently only rectangle points have 3D capability.

[S](#)

`#Draw3D_property`

`KDraw3D`

`$EndColor`

`$#KEnd Color`

Description

The **EndColor** property is one of the two colors used to create the gradient fill. The gradient fill is composed of two colors: **StartColor** and **EndColor**.

[S](#)

`#EndColor_Property__TChartGradient`

`KEndColor`

`$ExchangeSeries`

`$#KExchange Series`

Description

Interchanges the Series index between 2 Series. Their index position will determine there position from front to back on the Chart.

[S](#)

`#ExchangeSeries_Method`

`KExchangeSeries`

`$Legend Font`

`$##KLegend Font`

Description

Text font for Legend Text

[S](#)

`#Font_property__TChartLegend`
`KLegend Font`
`$Shape Font`

`$##K` KShape Font

Description

Text font for text placed in a ChartShape Series' Shape.

[S](#)

`#Font_property__TChartShape`
`K`Shape Font
`$`Axis Title Font

`$##KKAxis Title Font`

Description

Text font for Axis titles.

[S](#)

`#Font_property_TChartAxisTitle`
`KAxis Title Font`
`STitle Font`

`$#KTitle Font`

Description

Text font for the Chart title text.

[S](#)

`#Font_property_TChartTitle`
`KTitle Font`
`$Frame`

`$#KFrame`

Description

Enables/disables the Chart rectangle surrounding Frame.

[S](#)

`#Frame_property`
`KFrame`
`$Legend Frame`

\$[#]_#^K KLegend Frame

Description

Enables/disables the rectangle surrounding the Legend.

[S](#)

#Frame_property__TChartLegend
^KLegend Frame
\$GridPen

`$##KKGrid Pen`

Description

Pen used to draw Chart Grid Lines.

[S](#)

`#GridPen_property`

`KGridPen`

`$RotationAngle`

`$#KRotation Angle`

Description

The **RotationAngle** can be a valid integer number between 0 and 359. This will the angle each **Pie** sector will be rotated counter-clockwise.

[S](#)

`#hlp_PIEROTATION`
`KRotationAngle`
`$Axis Maximum`

\$[#]_#^KAxis Maximum

Description

Axis Maximum is the highest value an **Axis** will use to scale their dependent **Series** point values.

It can be any number or **DateTime** value.

It must be greater than the **Axis.Minimum** value.

VERY IMPORTANT:

Axis.Automatic property must be **FALSE**.

[S](#)

[#]**hlp_TCHARTAXISMAX**

^K**Axis Maximum**

^S**Horizontal Margin**

\$[#]_#^KHorizontal Margin

Description

The HorizMargin property specifies the number of screen pixels between Legend and Chart rectangles. By default its 0, meaning Legend will calculate a predefined margin based on total Legend width.

[S](#)

#HorizMargin_property

^KHorizontal Margin

^SHorizontal Size

\$[#]_#^KHorizontal Size

Description

The HorizSize property specifies the Series Pointer width in logical pixels.

Series that derive from TPointSeries usually override the HorizSize and VertSize properties.

[S](#)

#HorizSize_property__TSeriesPointer_
^KHorizontal Size
^{\$}Increment

`$##KIncrement`

Description

Axis Increment is the minimum step between axis labels. It must be a positive number or **DateTime** value. TChart will use this value as the starting axis labels step. If there is not enough space for all labels, TChart will calculate a bigger one.

[S](#)

`#Increment_property`

`KIncrement`

`$InflateMargins`

\$#^{#K}Inflate Margins

Description

The **InflateMargins** property controls if Chart dimensions will be rescaled to accomodate the **Series.Pointer HorizSize** and **VertSize** pixels. When **False**, Chart Axis scales will be preserved and points close to the Axis limits will be partially displayed.

[S](#)

#InflateMargins_property
KInflateMargins
\$Inverted Axis

`$#KInverted Axis`

Description

When Inverted is True, Axis Minimum and Maximum scales will be swapped. Axis labels and Series points will be displayed accordingly. This applies both to vertical and horizontal axis.

[S](#)

`#Inverted_property`
`KInverted Axis`
`$InvertedStairs`

\$[#]_#^KInverted Stairs

Description

When Stairs is set to True you may set InvertedStairs to True to alter the direction of the step.

[S](#)

#InvertedStairs_property

^KInvertedStairs

^{\$}Labels Visible

\$[#]_#^K KLabels Visible

Description

The **Labels** property shows or hide **Axis Labels**. Set it to **False** to draw the **Axis Ticks** and / or **Grid lines** only.

[S](#)

[#]**Labels_property**

^K**Labels Visible**

^{\$}**Labels enable**

\$[#]_#^K KLabels enable

Description

The **Labels** property shows or hide **Axis Labels**. Set it to **False** to draw the **Axis Ticks** and / or **Grid lines** only.

[S](#)

[#]**LabelsAx_property**

^K**Labels enable**

^{\$}**Labels Font**

\$[#]_#^KKLabels Font

Description

Text font for Axis Labels.

[S](#)

[#]LabelsFont_property
^KLabels Font
^{\$}Labels On Axis

`$#KLabels On Axis`

Description

The `LabelsOnAxis` property controls if Labels just at Axis Minimum and Maximum positions will be shown or NOT.

[S](#)

`#LabelsOnAxis_property`
`KLabels On Axis`
`$Labels Separation`

\$[#]_#^KLabels Separation

Description

The **LabelsSeparation** property specifies the percent amount of minimum distance between Axis Labels.

Setting it to "0" zero makes TChartAxis skip calculating overlapping labels. (No clipping is performed).

[S](#)

[#]LabelsSeparation_property

^KLabels Separation

^SLabels Size

\$[#]_#^K KLabels Size

Description

The LabelsSize property is 0 by default.

Therefore the space between the Axis and the Chart will be automatically calculated based on the Axis Labels Width and Height. Any other value will set the space in pixels.

[S](#)

[#]LabelsSize_Property

^KLabels Size

^{\$}Label Style

\$[#]_#^K KLabel Style

Description

Possible values:

Auto - Chooses the Style automatically.

None - No label.

Value - Axis labeling is based on axis Minimum and Maximum properties. **Mark** - Label using SeriesMarks style.

Text - Label using Series.XLabels strings.

[S](#)

#LabelStyle_property

^KLabel Style

^SLine Brush

\$[#]_#^KLine Brush

Description

The LineBrush property defines the brush style used to fill LineSeries contents.

It has effect only when Chart1.View3D is True.

The Legend reflects automatically the selected brush style.

[S](#)

[#]LineBrush_property

^KLine Brush

^{\$}Line Pen

`$##KKLine Pen`

Description

The LinePen property determines what kind of pen will be used for drawing the line connecting all points.

[S](#)

`#LinePen_property`

`KLine Pen`

`$Logarithmic`

`$#KLogarithmic`

Description

This property scales the Axis Logarithmically when enabled. Axis Minimum and Maximum values should be greater than 0, and Axis cannot be of DateTime type.

[S](#)

`#Logarithmic_property`
`KLogarithmic`
`$Margins`

`$#KMargins`

Description

Each Chart component has four margin properties: **LeftMargin**, **RightMargin**, **TopMargin**, **BottomMargin**.

These properties are expressed in screen pixels. Default values are 8 for top and bottom margins and 12 for left and right margins.

[S](#)

`#Margins_properties`

`KMargins`

`$Maximum Points Per Page`

\$[#]_#^KMaximum Points Per Page

Description

The **MaxPointsPerPage** property controls "TeeChart AutoPaging".

Setting it to a number greater than zero makes TeeChart to internally divide Series points in Pages.

[S](#)

[#]**MaxPointsPerPage_property**

^K**Maximum Points Per Page**

^{\$}**Axis Minimum**

\$[#]_#^KAxis Minimum

Description

Axis Minimum is the lowest value an Axis will use to scale their dependent Series point values. Can be any number or DateTime value. Axis 'Automatic' must be OFF.

Must be lower than the Axis.Maximum value.

[S](#)

[#]Minimum_property
^KAxis Minimum
^{\$}MinorTick Count

`$#KMinorTick Count`

Description

The **MinorTickCount** property determines the number of **Axis** minor ticks. **Axis** minor ticks are the **Axis** sub-ticks between major ticks.

[S](#)

`#MinorTickCount_property`
`KMinorTick Count`
`$MinorTick Length`

\$[#]_#^KKMinorTick Length

Description

The **MinorTickLength** property indicates the length in pixels of **Axis Minor** ticks.

[S](#)

[#]**MinorTickLength_property**
^K**MinorTick Length**
^{\$}**Minor Ticks**

\$[#]_#^KMinor Ticks

Description

The MinorTicks property is the Pen used to draw the Axis Minor ticks. Minor ticks will only be displayed if MinorTicks.Visible is True.

[S](#)

#MinorTicks_property

^KMinor Ticks

^{\$}Minimum Value

\$[#]_#^KMinimum Value

Description

The MinValue property returns the lowest of all values inside the TChartValuelist.

[S](#)

#MinValue_property
^KMinimum Value
\$MultiArea

`$##KKMultiArea`

Description

The **MultiArea** property determines the kind of displayed Area when there's more than one **AreaSeries** with the same **ParentChart**. The default value is **maNone**, meaning all Areas will be drawn one behind the other. **maStacked** and **maStacked100** modes will draw each Area on top of previous one. **msStacked100** adjusts each individual point to a common 0..100 axis scale. The order which Series are accumulated depends on the **Chart.SeriesList** property.

[S](#)

`#MultiArea_property`

`KMultiArea`

`$MultiBar`

`$#KMultiBar`

Description

If you have more than one TBarSeries in the same Chart, then you can choose if they will be drawn side-by-side, back-to-front or Stacked. Side-by-side means the Bar width will be divided by the number of Bar Series.

maNone: all Areas will be drawn one behind the other.

maStacked and **maStacked100:** draws each Area on top of previous one. **msStacked100** adjusts to a common 0..100 axis scale.

[S](#)

`#MultiBar_Property`

`KMultiBar`

`$NextPage`

`$#KNext Page`

Description

Advances a page divided Chart by one Page.

[S](#)

`#NextPage_Method`
`KNextPage`
`$Offset Percent`

\$[#]_#^KOffset Percent

Description

The **OffsetPercent** property indicates the Bar displacement in percent of Bar size. Displacement is horizontal for **TBarSeries** and vertical for **THorizBarSeries**. This property can be used to create "overlaid" Bar charts. Values may be +ve or -ve.

[S](#)

#OffsetPercent_property
^KOffset Percent
^{\$}First/Last

\$[#]_#^KFirst/Last

Description

Use the navigation buttons to scroll through the pages of a multipage Chart. First will take you to the first page and Last to the last page.

S

#Page_PropertyFL
^KFirst/Last
\$Page

`$#KKPage`

Description

The Page property determines the current visible points of Series in a Chart with MaxPointsPerPage property greater than zero.

[S](#)

`#Page_Property`

`KPage`

`$Shape Pen`

`$#KShape Pen`

Description

Determines Pen used to draw Lines of a Shape.

[S](#)

`#Pen_property__TChartShape`
`KShape Pen`
`$Pointer Pen`

`$##KKPointer Pen`

Description

Determines Pen used to draw the Lines of a Series Point.

[S](#)

`#Pen_property__TSeriesPointer`

`KKPointer Pen`

`SCircle Pen`

$\$^{\#}{}^{\mu}{}^K$ Circle Pen

Description

Determines Pen used to draw the Lines of a Circled (eg. Pie) Series.

[**\$\\$\$**](#)

$\#$ Piepen_property

K Circle Pen

$\$$ Previous Page

[\\$#KPrevious Page](#)

Description

Moves back one page on a pageable Chart.

[S](#)

[#PreviousPage_Method](#)
[KPrevious Page](#)
[\\$Remove Series](#)

`$##KKRemove Series`

Description

Removes the currently selected Series.

[S](#)

`#RemoveSeries_method`

`KKRemove Series`

`$$Resize Chart`

`$#KResize Chart`

Description

The **ResizeChart** property indicates if Legend will automatically reduce the Chart rectangle to prevent overlap between Legend and Chart rectangles. This enables placing of the Legend inside the Chart Rectangle.

[S](#)

`#ResizeChart_property`

`KResize Chart`

`$Round First Label`

\$[#]_#^KRound First Label

Description

The RoundFirstLabel property controls if Axis labels will be automatically "rounded" to the nearest magnitude. This applies both to DateTime and non-DateTime axis values. When False, Axis labels will start at Maximum Axis value.

[S](#)

[#]RoundFirstLabel_property

^KRound First Label

^{\$}Export Chart

\$[#]_#^KExport Chart

Description

This button will save the current chart as to the specified File Name. Options for export are bitmap (bmp), Windows metafile (wmf or emf), Clipboard or TeeChart tee format. Tee templates are an efficient way to save runtime Chart appearance and may be loaded at runtime using the LoadChartFromFile.

[S](#)

[#]SaveChartToFile_Method

^KExport Chart

^{\$}Scale Last Page

`$#KScale Last Page`

Description

In a pageable Chart, when enabled, the last Chart page will have the same horizontal scaling than the other pages. When disabled, the last Chart page scaling will be adjusted based on the number of visible points on that last page.

`S`

`#ScaleLastPage_property`
`KScale Last Page`
`$Series Color`

$\$^{\#}_{\#}^K$ Series Color

Description

Overall Colour of Series. This may be overridden by defining colours of individual Series points.

[S](#)

$\#$ SeriesColor_property

K Series Color

$\$$ Shadow Color

\$[#]_#^KKShadow Color

Description

The ShadowColor property specifies the color to fill the Legend shadow effect.

[S](#)

[#]ShadowColor_property

^KShadow Color

^{\$}Shadow Size

`$#KShadow Size`

Description

The ShadowSize property controls the amount of pixels used to draw the Legend shadow effect.

[S](#)

`#ShadowSize_property`

`KShadow Size`

`$Side Margins`

`$#KSide Margins`

Description

The **SideMargins** property controls if first and last displayed Bar will be separated from the Chart rectangle. By default, margins are set to half the sum of all Bar Series bar widths.

[S](#)

`#SideMargins_property`

`KSide Margins`

`$Stairs`

`$#KStairs`

Description

In most situations, a Series draws a line between each Line point. This makes the Line appear as a "mountain" shape. However, enabling 'Stairs' will make the Series to draw 2 Lines between each pair of points, thus giving a "stairs" appearance.

[S](#)

`#Stairs_property`

`KStairs`

`$Start Color`

\$[#]_#^KKStart Color

Description

The **StartColor** property is one of the two colors used to create the gradient fill. The gradient fill is composed of two colors: **StartColor** and **EndColor**.

[S](#)

#StartColor_Property__TChartGradient

^KStart Color

^{\$}Pen Style

`$##KKPen Style`

Description

The Style property determines the style in which the pen draw lines on the canvas.

[S](#)

`#Style_property__TChartPen`

`KKPen Style`

`$Shape Style`

`$#KShape Style`

Description

The **Style** property defines how a **TChartShape** component appears on a **Chart**.

These are the possible values and their meanings:

chasRectangle The shape is a rectangle

chasCircle The shape is a circle

chasVertLine The shape is a vertical line

chasHorizLine The shape is an horizontal line

chasTriangle The shape is a triangle

chasInvertTriangle The shape is an inverted triangle

chasLine The shape is a line

chasDiamond The shape is a diamond

[S](#)

`#Style_property__TChartShape`

`KShape Style`

`$Pointer Style`

`$##KPointer Style`

Description

The Style property defines the shape used to display the Series Points.

The Pointer can be:

psRectangle

psCircle

psTriangle

psDownTriangle

psCross

psDiagCross

psStar

[s](#)

`#Style_property__TSeriesPointer`

`KPointer Style`

`$Axes`

`$##KKAxes`

Description

Chart components have 5 TChartAxis:

LeftAxis, RightAxis, TopAxis, BottomAxis and DepthAxis.

Select an Axis with the radio button to activate that Axis for editing.

[S](#)

`#TChartAxis_Component`

`KKAxes`

`$Gradient`

`$#KKGradient`

Description

The TChartGradient component contains all properties used to draw a nice Chart background.

The gradient effect is made of two colors and many middle transition colors between them.

[S](#)

`#TChartGradient_Component`

`KGradient`

`$Gradient`

\$[#]_#^KKGradient

Description

Enables/disables a background Gradient on the Chart panel set according the Start and End colours.

[S](#)

[#]TChartGradient_Visible_property

^KGradient

^{\$}Titles' Brush

`$#KTitles' Brush`

Description

The **Brush** property determines the kind of brush used to fill the rectangle behind **Chart.Title** and **Chart.Foot** text.

[S](#)

`#TChartTitle_Brush_property_`

`KTitles' Brush`

`$Titles`

`$#KTitles`

Description

The TChartTitle component is used to display Chart Title and Foot text strings at top and bottom Chart sides respectively. The Text property contains the text to be displayed.

[S](#)

`#TChartTitle_Component`

`KTitles`

`$Wall Brush`

`$##KKWall Brush`

Description

The **Brush** property determines the kind of brush used to fill the **Chart Walls** background.

The **Chart.View3DWalls** property should be **True** to make walls visible.

[S](#)

`#TChartWall_Brush_property`

`KWall Brush`

`SWalls`

`$#K Walls`

Description

The TChartWall component controls how to display the Chart left, back and bottom walls.

[S](#)

`#TChartWall_Component`
`K Walls`
`$Size`

`$#KSize`

Description

The **Size** property determines the thickness of the selected Chart wall.

[S](#)

`#TChartWall_Size_property`
`KSize`
`$Text`

`$##KText`

Description

The **Text** property is used by TChartShape component to display customized text content inside Shapes.

[S](#)

`#Text_property__TChartShape`

`KText`

`$Text`

`$#KText`

Description

The text applying to a Chart Header or Footer depending on which is currently selected.

[S](#)

`#Text_property__TChartTitle`
`KText`
`$Tick InnerLength`

`$#KTick InnerLength`

Description

The `TickInnerLength` property defines the length in pixels of Axis ticks drawn inside Chart boundaries.

[S](#)

`#TickInnerLength_property`
`KTick InnerLength`
`STick Length`

\$[#]_#^KTick Length

Description

The TickLength property defines the length of Axis Ticks in logical pixels.

Use the Ticks Pen property to change pen attributes.

[S](#)

#TickLength_property

^KTick Length

^{\$}Tick On Labels Only

`$#KTick On Labels Only`

Description

This property sets the Axis Ticks and Axis Grid to be drawn only to coincide at Labels. Otherwise they will be drawn at all axis increment positions.

[S](#)

`#TickOnLabelsOnly_property`
`KTick On Labels Only`
`STicks`

`$#KTicks`

Description

The **Ticks** property determines the kind of Pen used to draw Axis marks along the Axis line.

The 3 kinds of ticks are: **Ticks**, **MinorTicks** and **TicksInner**.

[S](#)

`#Ticks_property`

`KTicks`

`$TicksInner`

`$#KInner Ticks`

Description

The TicksInner property determines the kind of Pen used to draw Axis marks along the Axis line. This is the same Ticks property does but lines are drawn inside Chart boundaries instead.

[S](#)

`#TicksInner_property`

`KTicksInner`

`$Title`

`$#KTitle`

Description

Every TChartSeries has a Title property of type String. The Title property is used in TChart.Legend to draw the series descriptions. If Title is empty, then the Series component Name will be used to draw the legend.

[S](#)

`#Title_property`
`KTitle`
`$Title`

`$##KTitle`

Description

The Title property defines the Text and formatting properties to be drawn at Top Chart side.

[S](#)

`#Title_property__TChart`
`KTitle`
`$Title`

`$#KTitle`

Description

Axis Title for the currently selected Axis.

[S](#)

`#Title_property__TChartAxis`
`KTitle`
`$Title Size`

`$#KTitle Size`

Description

The `TitleSize` property is 0 by default.

Therefore the space between the Axis Title and the Chart will be automatically calculated based on the Axis Title Width and Height.

[S](#)

`#TitleSize_property`
`KTitle Size`
`$Top Position`

`$#KTop Position`

Description

Default Value: 10

The TopPos property specifies the Legend's top position in percent of total chart height.

It's used when TChartLegend.Alignment is Left or Right only.

[S](#)

`#TopPos_property`

`KTop Position`

`$Transparent`

`$#KTransparent`

Description

Sets the Chart Shape's background to transparent.

[S](#)

`#Transparent_property__TChartShape`
`KTransparent`
`$UsePatterns`

`$#KUse Patterns`

Description

The **UsePatterns** property indicates, when True, that Pie Sectors will be filled using different Brush pattern styles. There are 6 different pattern styles.

[S](#)

`#UsePatterns_property`

`KUsePatterns`

`SUseOrigin`

\$[#]_#^KUse the Origin

Description

The UseYOrigin property defines if Bars will be bottom aligned to the YOrigin property value.

When False, the minimum of all Bar values is used as the Bar origins value.

[S](#)

#UseYOrigin_property

^KUseOrigin

^SValueFormat

`$#K` KValue Format

Description

ValueFormat sets the way to display values. Eg. `###0.0###`

[S](#)

`#ValueFormat_property`

`KValueFormat`

`$Point Vertical Size`

\$[#]_#^KKPoint Vertical Size

Description

Vertical size of the Series' Point

[S](#)

#VertSize_property__TSeriesPointer_
^KPoint Vertical Size
^{\$}View3D

`$##KView3D`

Description

Enables viewing of the Chart as 2D or 3D

[S](#)

`#View3D_property`
`KView3D`
`$Visible Wall`

`$##KKVisible Wall`

Description

View3DWalls will draw **Left, Back and Bottom "walls"** to simulate 3D effect.

[S](#)

`#View3DWalls_property`

`KVisible Wall`

`$Visible Axis`

`$#KVisible Axis`

Description

Enables/disables display of currently selected Axis.

[S](#)

`#Visible_property__TChartAxis`
`KVisible Axis`
`$Visible Legend`

`$#KVisible Legend`

Description

Enables/disables display of Legend.

[S](#)

`#Visible_property__TChartLegend`
`KVisible Legend`
`$Visible`

\$[#]_#^KVisible

Description

Enables/disables display of Lines.

[S](#)

#Visible_property__TChartPen
^KVisible
^{\$}Visible Title

`$#KVisible Title`

Description

Enables/disables display of the Chart Title.

[S](#)

`#Visible_property__TChartTitle`
`KVisible Title`
`$Visible Pointer`

`$#KVisible Pointer`

Description

Enables/disables display of Series' Points.

[S](#)

`#Visible_property__TSeriesPointer`
`KVisible Pointer`
`$Width`

\$[#]_#^KWidth

Description

Width of Pen. For Width greater than 1 style is disabled, the Line will be drawn as a solid Line.

[S](#)

[#]Width_property__TChartPen
^KWidth
^{\$}X

$\$^{\# \mu K} X$

Description

These properties define the Top - Left and Bottom - Right coordinates of the englobing TChartShape rectangle.

[S](#)

$\#X_{\text{property}}$

$^K X$

$\$X$ Radius

`$#K` **XX Radius**

Description

The `XXRadius` property returns the exact ellipse's radius horizontal size in pixels.

[S](#)

`#XXRadius_property`

`KX Radius`

`$Origin`

\$[#]_#^KKOrigin

Description

The YOrigin property determines the axis value used as a common bottom for all Bar points.

The UseYOrigin property must be True (the default) to use the YOrigin property.

[S](#)

#YOrigin_property

^KKOrigin

^{\$}Y Radius

`$##KY Radius`

Description

The YRadius property returns the exact ellipse's radius vertical size in pixels.

[S](#)

`#YRadius_property`

`KY Radius`

`$Elevation`

\$[#]_#^KElevation

Description

Elevation describes front plane rotation by rotation degrees (0 - 360°). Increasing the value positively will bring the top of the Chart towards the viewer and the bottom of the Chart away.

S

#Elevation_property

^KElevation

^SChart Horizontal Offset

\$[#]_#^KChart Horizontal Offset

Description

HorizOffset will move the **Chart Rectangle** horizontally across the **Chart Panel**.

[S](#)

[#]**HorizOffset_property**

^K**Chart Horizontal Offset**

^S**Orthogonal**

`$#KOrthogonal`

Description

Disabling Orthogonal (setting to False) disables the 2D Canvas and enables the 'Full' 3D Canvas allowing Elevation and Rotation displacement of the Chart.

[S](#)

`#Orthogonal_property`

`KOrthogonal`

`$Rotation`

\$[#]_#^KRotation

Description

Rotation describes front plane rotation by rotation degrees (0 - 360°). Increasing the value positively will bring the right of the Chart towards the viewer and the left of the Chart away.

[S](#)

#Rotation_property
^KRotation
^{\$}Tilt

\$[#]_#^KTilt

Description

ilt will rotate the Chart Rectangle within the Chart Panel. Positive values (from 0 to 360°) rotate the Chart anti-clockwise, negative values, clockwise.

***Important. Orthogonal should be set to False for Rotation to act on the Chart.**

S

#Tilt_property

^KTilt

\$Chart Vertical Offset

\$[#]_#^KChart Vertical Offset

Description

VertOffset will move the Chart Rectangle vertically across the Chart Panel. Positive values move the Chart downwards, negative values upwards.

[S](#)

#VertOffset_property
^KChart Vertical Offset
\$Zoom

`$#KZoom`

Description

View3DOptions' Zoom will zoom the whole Chart. Expressed as a percentage, Increasing the value positively will bring the Chart towards the viewer, increasing the overall Chart size as the Zoom value increases.

[S](#)

`#Zoom_property_View3d`

`KZoom`

`$Explode Biggest Slice`

`$##KExplode Biggest Slice`

Description

Use this property to explode out from the chart the largest slice.

[S](#)

`#ExplodeBiggest_property_TPieSeries`

`KExplode Biggest Slice`

`SOtherSlice`

`$##KKOther Slice`

Description

Use this property to define the grouping size for the 'Other' slice of the Pie. Grouping may be expressed as a percentage or value.

[S](#)

`#OtherSlice_property_TPieSeries`
`KOtherSlice`
`$Style`

`$#KStyle`

Description

Defines whether to use value or percentage to group the 'other' Pie slice.(eg. None, BelowPercent, BelowValue);

[S](#)

`#Style_property_TPieOtherSlice`

`KStyle`

`$Text`

$\$^{\#}_{\#}K\text{Text}$

Description

Label for 'Otherslice'

[S](#)

$\# \text{Text_property_TPicOtherSlice}$
 $K\text{Text}$
 $\$ \text{Value}$

$\$^{\#}_{\#}KValue$

Description

Threshold value below which to include data point (or slice) in grouped 'Otherslice'.

[S](#)

$\#Value_property_TPieOtherSlice$
 $KValue$
 $\$Grid\ Centered$

\$[#]_#^KKGrid Centered

Description

Places Axis Grid lines between (not at) label positions.

[S](#)

#GridCentered_property_TChartAxis
^KGrid Centered
^{\$}Zoom Text enable

`$##KKZoom Text enable`

Description

ZoomText enables/disables the zooming of text when using the **TView3DOptions.Zoom** property.

[S](#)

`#ZoomText_Property_View3D`

`KZoom Text enable`

`$Marks Style`

\$[#]_#^K KMarks Style

Description

MarksStyle defines the possible values of the **TSeriesMarks.Style** property. **Series Marks** will draw a textual representation of each point values.

smsValue	1234
smsPercent	12 %
smsLabel	Cars
smsLabelPercent	Cars 12 %
smsLabelValue	Cars 1234
smsLegend	(Depends on LegendTextStyle)
smsPercentTotal	12 % of 1234
smsLabelPercentTotal	Cars 12 % of 1234
smsXValue	{ 21/6/1996 }

[S](#)

#Style_property__TSeriesMarks
^K Marks Style
^S Marks Back Colour

`$##KKMarks Back Colour`

Description

The BackColor property defines the color used to fill the Marks background rectangle.

[S](#)

`#BackColor_property_TSeriesMarks`

`KKMarks Back Colour`

`SMarks Font`

`$##K` KMarks Font

Description

The **Font** property determines the font used to draw the **Series Marks**.

[S](#)

`#Font_Property__T`SeriesMarks
`K`Marks Font
`$`Marks Frame

`$##KKMarks Frame`

Description

The **Frame** property determines the kind of pen used to draw a rectangle around a **Series Marks**.

[S](#)

`#Frame_property__TSeriesMarks`

`KMarks Frame`

`SMarks Arrow`

`$##KKMarks Arrow`

Description

The **Arrow** property determines the kind of pen used to draw a line connecting the **Point Mark** to the corresponding **Series** point.

[S](#)

`#Arrow_property`

`KKMarks Arrow`

`$ArrowLength`

$\$^{\#}_{\#}K$ Arrow Length

Description

The **ArrowLength** property determines the number of pixels used to display a line connecting the **Series Marks** to their corresponding points.

[S](#)

$\#$ ArrowLength_property
 K ArrowLength
 $\$$ Clipped

`$##KKClipped`

Description

The Clip boolean property toggles if Marks outside Chart limits will overwrite any other Chart region.

[S](#)

`#Clip_Property
KKClipped
$Transparent`

`$#KTransparent`

Description

The Transparent property controls if Series Marks background will be filled or not.

[S](#)

`#Transparent_property`
`KTransparent`
`$Visible`

\$[#]_#^KKVisible

Description

The Visible property controls if Series Marks will be displayed or not.

[S](#)

#Visible_property__TSeriesMarks
^KVisible
^{\$}Show in Legend

`$#KShow in Legend`

Description

Controls whether or not the series title should display in Chart.Legend. It is only meaningful when when LegendStyle is 'Series' or 'LastValues'.

[S](#)

`#ShowInLegend_property`

`KShow in Legend`

`SSeries Cursor`

`$##KKSeries Cursor`

Description

The Cursor property is the image used when the mouse passes into the region covered by a Series point. Each Series determines the intersection of points with mouse coordinates each time the mouse moves.

[S](#)

`#Cursor_property`

`KKSeries Cursor`

`SHorizontal Axis`

$\$^{\#}_{\#}^K$ Horizontal Axis

Description

Defines the Horizontal Axis (Top or Bottom axis) to which the Series will be horizontally scaled.

[S](#)

$\#$ HorizAxis_property

K Horizontal Axis

$\$$ Vertical Axis

\$[#]_#^KVertical Axis

Description

Defines the Vertical Axis (Left or Right axis) to which the Series will be vertically scaled.

[S](#)

#VertAxis_property
^KVertical Axis
^{\$}DateTime

`$#KKDateTime`

Description

The **DateTime** property tells TeeChart what type the numbers are. The horizontal (x axis) and vertical (y axis) value defaults are number format (DateTime False).

[S](#)

`#DateTime_property`

`KDateTime`

`$PercentFormat`

`$#KPercentFormat`

Description

Chart Series components have a PercentFormat property. It is used to draw the Series Marks Percent Style figures.

[S](#)

`#PercentFormat_property`
`KPercentFormat`
`$Legend Style`

\$[#]_#^KLegend Style

Description

The LegendStyle property defines which items will be displayed in Chart Legend.

IsAuto TChartLegend draws Series Titles if there's more than one TChartSeries in the TChart

IsSeries TChartLegend draws the Series Titles (also if there's only one TChartSeries in the chart).

IsValues TChartLegend draws the first Active TChartSeries values.

IsLastValues TChartLegend draws the Last Value of each Active TChartSeries (similar to IsSeries).

[S](#)

[#]LegendStyle_property

^KLegend Style

^SLegend Text Style

\$[#]_#^KLegend Text Style

Description

LegendTextStyle defines the possible values of the **TChartLegend.TextStyle** property.

ItsPlain	Summer
ItsLeftValue	1234 Summer
ItsRightValue	Summer 1234
ItsLeftPercent	5.1 % Summer
ItsRightPercent	Summer 5.1 %
ItsXValue	4321 (Applies only to Series with X values. See TChartSeries.AddXY method.)

[S](#)

#TextStyle_property
KLegend Text Style
\$Wall Border

\$[#]_#^KWall Border

Description

Determines the kind of pen used to draw the Chart Walls frame. The Chart.View3DWalls property should be True to make walls visible.

[S](#)

[#]TChartWall_Pen_property

^KWall Border

^SPosition

`$#KPosition`

Description

Position of Axis as percentage of Chart Height (horizontal Axis) or Chart Width (vertical Axis). Top Left is 0,0.

[S](#)

`#PositionPercent_property`
`KPosition`
`$Start`

$\$^{\#}_{\#}KStart$

Description

Start of Axis as percentage of width (horizontal Axis) and height (vertical Axis) of the Chart rectangle. Original Axis scale will be fitted to new Axis height/width. Left Top is 0,0

$\$$

$\#StartPosition_property$
 $KStart$
 $\$End$

\$[#]_#^KKEnd

Description

End of Axis as percentage of width (horizontal Axis) and height (vertical Axis) of the Chart rectangle. Original Axis scale will be fitted to new Axis height/width. Left Top is 0,0

S

#EndPosition_property

^KKEnd

^SLabels Multiline

\$[#]_#^KLabels Multiline

Description

When True, spaces in point Labels or in Axis DateTimeFormat / ValueFormat are used to break the label in more than one line of text).

When False, labels with #13 characters are break in lines.

Example

Series1.Add(1234, 'Hello'+#13+'World', clGreen);

[S](#)

#LabelsMultiline_property

^KLabels Multiline

^SChart Title Color

`$#K`Chart Title Color

Description

The Colour used to fill the Chart background

[S](#)

`#Color_property_T`ChartTitle
`K`Chart Title Color
`$`Transparent Wall

`$#KTransparent Wall`

Description

Defines whether the Wall will be filled with the Wall colour or be transparent.

[S](#)

`#Transparent_property_TChartWall`
`KTransparent Wall`
`$Clone Series`

$\$^{\#}_{\#}K$ Clone Series

Description

Makes a new, duplicate Series of selected Series.

[S](#)

$\#$ Cloneseries
 K Clone Series
 $\$$ Period

`$#KPeriod`

Description

Period refers to the point range over which the function is applied. Eg. A **Period** of 3 for the **Average** function will calculate and plot the average for every 3 points. A **Period** of 0 sets TeeChart to add it's default **Period**.

[S](#)

`#Period_property`

`KPeriod`

`$Set Function source`

\$[#]_#^KKSet Function source

Description

Use the 'Available' and 'Selected' windows to select Series to be used as source data for the function. For multiple Series input (eg. Subtract Function) the Top Series in the List will subtract the 2nd and so on..

S

#SetFunction_method

^KSet Function source

^{\$}Datasource

\$[#]_#^KKDatasource

Description

The "point provider" (or DataSource) can be:

1) Another Chart Series.

2) Any TTable, TQuery, TClientDataset or Delphi database dataset.

Eg. Select a TTable in the Dataset Window and its fields will become available for Labels, X and Y values.

[S](#)

#DataSource_property

^KDatasource

^{\$}BevelOuter property (TChart)

`$#KOuter bevel`

Description

Sets the size of the bevel that will form the outer side of the Chart panel.

[S](#)

`#BevelOuter_Property`

`KBevelOuter`

`$BevelInner property (TChart)`

\$[#]_#^KInner Bevel

Description

Sets the size of the bevel that will form the inner side of the Chart panel.

[S](#)

#BevelInner_Property

^KBevelInner

^{\$}Perspective property

`$##KKPerspective`

Description

Use this property with Orthogonal unchecked to modify the 3D perspective of the Chart. Larger values add more depth perspective.

`#Perspective_Property
KPerspective`

