

## Lotus Chart: Align property

{button ,AL('H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) The horizontal alignment of a text value.

### Data type

Integer (Enumerated)

### Syntax

*object.Font.Align* = *alignmentvalue*

*alignmentvalue* = *object.Font.Align*

### Legal values

<u>Value</u>	<u>Description</u>
\$DefaultAlign	Use current default alignment
\$LeftAlign	Align on the left margin
\$RightAlign	Align on the right margin
\$CenterAlign	Center the text
\$Justify	Use right and left justification

## Lotus Chart: ApplyChartStyle method

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

Applies the specified style to the chart.

### Syntax

*chart.ApplyChartStyle(stylename)*

### Parameters

*stylename*

The name of the file containing the chart style, specified as a string.

### Returns

Void

---

{button ,AL('H\_SAVECHARTSTYLE\_METHOD\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: AreValuesManual property

{button ,AL(`H\_CHARTMAJORGRIDLINES\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether gridlines are placed manually.

### Data type

Integer (Boolean)

### Syntax

*chart.axis.MajorGridlines.AreValuesManual* = *trueorfalse*

*trueorfalse* = *chart.axis.MajorGridlines.AreValuesManual*

### Legal values

True (1) or False (0)

---

{button ,AL(`H\_VALUE\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: AxisPlacement property

{button ,AL('H\_CHARTAXIS\_CLASS',0)} [See list of classes](#)

Controls the placement of the axis.

### Data type

Integer (Enumerated)

### Syntax

*chart.axis.AxisPlacement = placementvalue*

*placementvalue = chart.axis.AxisPlacement*

### Legal values

<u>Value</u>	<u>Description</u>
16	Standard position (bottom for x-axis; left for y-axis)
32	Alternate position (top for x-axis; right for y-axis)
48	Both (top and bottom for x-axis; left and right for y-axis)

## Lotus Chart: BackColorIndex property

{button ,AL(`H\_BACKGROUND\_CLASS ;H\_BORDER\_CLASS ;H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) The color for the background of a filled object in 1-2-3 and WordPro. In Freelance Graphics, use the color classes.

### Data type

Integer

### Syntax

*object*.BackColorIndex = *color*

*color* = *object*.BackColorIndex

### Legal values

0..255

### Usage

**Note** This property is available in Lotus 1-2-3 and Word Pro. For other Lotus products, consult the LotusScript Help file for that product.

The number you specify is an index into the current color palette.

Use BackColorIndex to set the color for the background of a filled object. To set the color of the pattern, use the ForeColorIndex property. If the pattern is set to a solid color, then only the ForeColorIndex shows up.

---

{button ,AL(`H\_FORECOLORINDEX\_PROPERTY\_MEMDEF ;H\_PATTERN\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Background class

The fill of a region.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartDataPoint	Fill
ChartLegend	Fill
ChartNumberGrid	Fill
ChartPieSlice	Fill
ChartPieSliceGroup	Fill
ChartPlot	Fill
ChartPlotWall	Fill
ChartSeries	Fill
ChartTable	Fill
ChartTableBase	Fill
ChartTitleBox	Fill

## Lotus Chart: Background class members

### Properties

[BackColorIndex](#)

[ForeColorIndex](#)

[Pattern](#)

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: BarClusterGapPercent property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_BARCLUSTERGAPPERCENT\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The space between clusters of bars, calculated as a percentage of the width of one bar.

### Data type

Integer

### Syntax

*barchart*.BarClusterGapPercent = *clusterpercent*

*clusterpercent* = *barchart*.BarClusterGapPercent

### Legal values

0..100, with 0 leaving no space between the bars, and 100 leaving space equivalent to the width of a bar between the bars.



## Lotus Chart: BarOverlapPercent property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The percentage of overlap of each bar in a bar chart. The percentage is calculated as the percent of the width of one bar.

### Data type

Integer

### Syntax

*barchart*.BarOverlapPercent = *overlappercent*

*overlappercent* = *barchart*.BarOverlapPercent

### Legal values

0..100, with 0 showing no overlap between the bars, and 100 having the bars overlap entirely, so that each bar is placed directly on top of the next bar.

## Lotus Chart: BarPlacement property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) How bar chart bars are organized.

### Data type

Integer (Enumerated)

### Syntax

*barchart*.BarPlacement = *placement*

*placement* = *barchart*.BarPlacement

### Legal values

<u>Value</u>	<u>Description</u>
\$InRows	Places each bar in a separate row, one behind the other. (Available for 3D charts only.)
\$Stacked	Places each bar stacked on the previous bar.
\$Clustered	Arranges bars next to each other, in groups.

## Lotus Chart: Bold property

{button ,AL(^H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to use a bold font for the specified text string.

### Data type

Boolean

### Syntax

*object.Font.Bold = trueorfalse*

*trueorfalse = object.Font.Bold*

### Legal values

True (1) or False (0)

## Lotus Chart: Border class

The border around an object or a line edge.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartAxis	Edge
ChartDataPoint	Edge
ChartGridline	Edge
ChartGridlines	Edge
ChartLegend	Edge
ChartMajorGridline	Edge
ChartMajorGridlines	Edge
ChartNumberGrid	Edge
ChartPlot	Edge
ChartPlotWall	Edge
ChartRegression	Edge
ChartSeries	Edge
ChartTable	Edge
ChartTableBase	Edge
ChartTableGridline	Edge
ChartTableGridlines	Edge
ChartTitleBox	Edge

## Lotus Chart: Border class members

### Properties

BackColorIndex

ForeColorIndex

Pattern

Style

Width

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: ChartAxisScale class

The numeric scale of an axis.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartAxis	Scale

## Lotus Chart: ChartAxisScale class members

### Properties

[CrossPoint](#)  
[CrossPointIsManual](#)  
[Exponent](#)  
[IsAscending](#)  
[IsNumeric](#)  
[MajorInterval](#)  
[MajorIntervalsManual](#)  
[Maximum](#)  
[MaximumIsManual](#)  
[Minimum](#)  
[MinimumIsManual](#)  
[MinorInterval](#)  
[MinorIntervalsManual](#)  
[ScaleType](#)

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: ChartAxisTitle class

The title and subtitle of an axis.

### Base classes

ChartTextEntry

### Contained by

<u>Class</u>	<u>Property</u>
ChartAxis	Title, Subtitle



## Lotus Chart: ChartAxisTitle class members

### Properties

Visible

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: ChartAxis class

An axis of a chart.

### Base class

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	XAxis, YAxis, SecondYAxis

---

{button ,AL(`H\_CHARTZAXIS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartAxis class members

### Properties

[AxisPlacement](#)  
[Edge](#) AS [Border](#)  
[Format](#) AS [NumberFormat](#)  
[MajorGridlines](#) AS [ChartMajorGridlines](#)  
[MajorTickmarkPlacement](#)  
[MajorTickmarks](#) AS [ChartGridlines](#)  
[MinorGridlines](#) AS [ChartGridlines](#)  
[MinorTickmarkPlacement](#)  
[MinorTickmarks](#) AS [ChartGridlines](#)  
[Scale](#) AS [ChartAxisScale](#)  
[Subtitle](#) AS [ChartAxisTitle](#)  
[SubtitlesManual](#)  
[SubtitlePlacement](#)  
[TickLabelCharWidth](#)  
[TickLabelPlacement](#)  
[TickLabels](#) AS [ChartTextLabels](#)  
[TickLabelsLink](#)  
[TickLabelSkip](#)  
[TickLabelWidthsManual](#)  
[Title](#) AS [ChartAxisTitle](#)  
[TitlePlacement](#)  
[Visible](#)

### Methods

[Select](#)

### Functions

None

### Events

None

## **Lotus Chart: ChartBase class**

A chart of any type. This class contains all other chart classes.

### **Base classes**

None

### **Usage**

The way in which you create a chart differs depending on the application you're using. For information on creating a chart, search for "Charts, creating with LotusScript" in the help Index.

When you create a chart, you automatically create all the components of a chart.

## Lotus Chart: ChartBase class members

### Properties

[BarClusterGapPercent](#)  
[BarOverlapPercent](#)  
[BarPlacement](#)  
[ConnectStackedBarLines](#)  
[DataLink](#) AS [ChartDataSource](#)  
[InsetLinesFromFrame](#)  
[Is3D](#)  
[IsDepthEffect](#)  
[IsHorizontal](#)  
[IsStacked](#)  
[Legend](#) AS [ChartLegend](#)  
[LinkUpdateMethod](#)  
[Note](#) AS [ChartTitleBox](#)  
[NumberGrid](#) AS [ChartNumberGrid](#)  
[Pies](#) AS [ChartPies](#)  
[PieSlices](#) AS [ChartPieSlices](#)  
[Plot](#) AS [ChartPlot](#)  
[RadarAxisTitles](#) AS [ChartTextLabels](#)  
[RadarLabelEachAxis](#)  
[RadarUseIndependentScales](#)  
[ScatterUseSeparateXValues](#)  
[SecondYAxis](#) AS [ChartAxis](#)  
[Series](#) AS [ChartSeriesCollection](#)  
[SeriesDataLink](#) AS [ChartDataSource](#)  
[SeriesParse](#)  
[StackedTotalsAngle](#)  
[StackedTotalsFormat](#) AS [NumberFormat](#)  
[StackedTotalsLabels](#) AS [ChartDataLabels](#)  
[Table](#) AS [ChartTable](#)  
[ThreeDElevation](#)  
[ThreeDLateralRotation](#)  
[ThreeDLightSource](#)  
[ThreeDPlatform](#)  
[ThreeDRowGapPercent](#)  
[Title](#) AS [ChartTitleBox](#)  
[Type](#)  
[UseCandlestickForHLCO](#)  
[UseSmartLookColors](#)  
[XAxis](#) AS [ChartAxis](#)  
[Y](#)  
[YAxis](#) AS [ChartAxis](#)  
[ZAxis](#) AS [ChartZAxis](#)

### Methods

[ApplyChartStyle](#)  
[SaveChartStyle](#)  
[Select](#)  
[UpdateLinks](#)

### Functions

None

### Events

None

## Lotus Chart: ChartDataLabels class

A set of value labels in a chart.

### Base class

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	StackedTotalsLabels
ChartNumberGrid	ColumnHeaders, ColumnTotals, RowHeaders, RowTotals
ChartPies	PercentLabels, TextLabels, ValueLabels
ChartSeries	DataLabels
ChartTable	ColumnHeaders, RowHeaders
ChartTableBase	ColumnHeaders, RowHeaders

## Lotus Chart: ChartDataLabels class members

### Properties

Font AS Font

Visible

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartDataLabel class

A data label that gets its font and link from its container.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartNumberGrid	RowTotalsHeader, ColumnTotalsHeader



## Lotus Chart: ChartDataLabel class members

### Properties

Font

Rotation

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartDataPoints class

The collection of all data markers for a specific chart series.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartSeries	DataPoints

### Usage

You cannot manipulate a collection of data points, you can only access attributes of individual points. To access an individual data point, use the ChartDataPoint class.

---

{button ,AL('H\_CHARTDATAPOINT\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartDataPoints class members

### Properties

None

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: ChartDataPoint class

A single data value in a chart series.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartSeries	DataPoints

### Usage

You access an individual data point using an index value, such as DataPoints(1), which represents the first data point.

## Lotus Chart: ChartDataPoint class members

### Properties

Edge AS Border

Fill AS Background

Text

Value

XValue

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartDataSource class

The link source of a chart subobject.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartLegend	DataLink
ChartLegendEntries	DataLink
ChartSeries	DataLink
ChartTextEntries	DataLink
ChartTextEntry	DataLink
ChartTextLabels	TickLabelsLink
ChartTitleBox	DataLink

## Lotus Chart: ChartDataSource class members

### Properties

Source

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: ChartGridlines class

A set of ruled lines on a chart.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartAxis	MajorTickmarks, MinorGridlines, MinorTickmarks

### Usage

To access a single grid line, use the ChartGridline class.



## Lotus Chart: ChartGridlines class members

### Properties

Edge AS Border

Visible

### Methods

Select

### Functions

None

### Events

None

**Lotus Chart: ChartGridline class**

A straight line on a chart.

**Base classes**

None

**Usage**

ChartGridline contains base attributes used both in the ChartMajorGridline and ChartTableGridline classes.

## Lotus Chart: ChartGridline class members

### Properties

Edge AS Border

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartLegendEntries class

The collection of text entries in a chart legend.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartLegend	Entries

---

{button ,AL(`H\_CHARTLEGEND\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartLegendEntries class members

### Properties

None

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: ChartLegend class

The chart legend.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	Legend

### Usage

To access the entries of a chart legend, use the ChartLegendEntries class.

---

{button ,AL('H\_CHARTLEGENDETRIES\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartLegend class members

### Properties

DataLink AS ChartDataSource

Edge AS Border

Entries AS ChartLegendEntries

Fill AS Background

Font AS Font

Height

InsidePlot

Placement

Visible

Width

X

Y

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartMajorGridlines class

The collection of all major gridlines.

### Base classes

None

### Contained by

List any classes that contain this class.

<u>Class</u>	<u>Property</u>
ChartAxis	MajorGridLines

---

{button ,AL('H\_CHARTMAJORGRIDLINE\_CLASS;H\_CHARTGRIDLINES\_CLASS;',0)} [See related topics](#)



## Lotus Chart: ChartMajorGridlines class members

### Properties

[AreValuesManual](#)

[Edge AS Border](#)

[GridlineCount](#)

[ShowLabels](#)

[Visible](#)

### Methods

[Select](#)

### Functions

None

### Events

None

## Lotus Chart: ChartMajorGridline class

A major grid line on an axis.

### Base classes

ChartGridline

### Contained by

<u>Class</u>	<u>Property</u>
ChartAxis	MajorGridlines

### Usage

You access an individual major grid line using an index value, such as MajorGridlines(1), which represents the first major grid line.

To access the collection of all major gridlines, use the ChartMajorGridlines class.

---

{button ,AL(`H\_CHARTGRIDLINES\_CLASS ;H\_CHARTMAJORGRIDLINES\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartMajorGridline class members

### Properties

Edge AS Border

Index

Value

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartNumberGrid class

A number grid chart.

### Base classes

ChartTableBase

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	NumberGrid

---

{button ,AL('H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartNumberGrid class members

### Properties

ColumnHeaders AS ChartDataLabels

ColumnTotals AS ChartDataLabels

DataArrangement

Edge AS Border

Fill AS Background

FirstHorizontal AS ChartTableGridline

FirstVertical AS ChartTableGridline

Height

InteriorHorizontals AS ChartTableGridlines

InteriorVerticals AS ChartTableGridlines

PlacementIsManual

RowHeaders AS ChartDataLabels

RowTotals AS ChartDataLabels

RowTotalsHeader AS ChartDataLabel

ColumnTotalsHeader

SeriesData AS ChartTableSeriesCollection

ShowUpperLeft

Width

X

Y

### Methods

Select

### Functions

None

### Events

None

**Lotus Chart: ChartPieSliceGroups class**

Represents the collection of slice groups in a pie or doughnut chart.

**Base classes**

None

**Lotus Chart: ChartPieSliceGroup class**

An individual slice across all pies or doughnuts in a multiple pie or doughnut chart.

**Base classes**

None

## Lotus Chart: ChartPieSlices class

The collection of slices in an individual pie chart.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	PieSlices
ChartPie	Slices

### Usage

You cannot manipulate a collection of pie slices, you can only access attributes of individual pie slices. To access an individual slice in a pie chart, use the ChartPieSlice class.

---

{button ,AL(`H\_CHARTPIESLICE\_CLASS ',0)} [See related topics](#)



**Lotus Chart: ChartPieSlices class members**

**Properties**

None

**Methods**

None

**Functions**

None

**Events**

None

## Lotus Chart: ChartPieSlice class

A single slice in an individual pie chart.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	PieSlices
ChartPie	Slices

## Lotus Chart: ChartPieSlice class members

### Properties

ExplodePercent

Fill AS Background

Visible

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartPies class

The collection of pies in a pie chart.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	Pies

## Lotus Chart: ChartPies class members

### Properties

[DepthPercent](#)  
[Elevation](#)  
[Height](#)  
[LayoutIsManual](#)  
[LayoutRowCount](#)  
[LightAngle](#)  
[PercentLabelFormat](#) AS [NumberFormat](#)  
[PercentLabelPlacement](#)  
[PercentLabels](#) AS [ChartDataLabels](#)  
[ScaleType](#)  
[ShadowPlacement](#)  
[ShadowSize](#)  
[SliceDirection](#)  
[SliceExplosion](#)  
[SliceSort](#)  
[StartAngle](#)  
[Subtitles](#) AS [ChartPieTitles](#)  
[SubtitlesAreManual](#)  
[TextLabels](#) AS [ChartDataLabels](#)  
[Titles](#) AS [ChartPieTitles](#)  
[UseSeparateLabelRanges](#)  
[ValueLabelFormat](#) AS [NumberFormat](#)  
[ValueLabelPlacement](#)  
[ValueLabels](#) AS [ChartDataLabels](#)  
[Width](#)  
[X](#)  
[Y](#)

### Methods

[Select](#)

### Functions

None

### Events

None

## Lotus Chart: ChartPieTitles class

A collection of pie titles or subtitles.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartPies	Subtitles, Titles

### Usage

To access a single pie title or subtitle, use the ChartPieTitle class.

---

{button ,AL('H\_CHARTPIETITLE\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartPieTitles class members

### Properties

DataLink AS ChartDataSource

Font AS Font

Placement

Text

Visible

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartPie class

An individual pie.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	Pies

### Usage

To access the collection of pies in a multiple pie chart, use the ChartPies class.



## Lotus Chart: ChartPie class members

### Properties

Visible

SlicesBeforeOther

OtherSliceText

Slices

DataLink

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartPlotWall class

The wall or floor of a 3D chart.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartPlot	LeftWall, RightWall, PlatformFrontFace, PlatformRightFace, PlatformTopFace

## Lotus Chart: ChartPlotWall class members

### Properties

BackColorIndex

Edge AS Border

Fill AS Background

ForeColorIndex

Pattern

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartPlot class

The plot area of a chart.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	Plot

## Lotus Chart: ChartPlot class members

### Properties

Edge AS Border

Fill AS Background

HasPicture

Height

LeftWall AS ChartPlotWall

PlatformFrontFace AS ChartPlotWall

PlatformRightFace AS ChartPlotWall

PlatformTopFace AS ChartPlotWall

RadarHasPolygonalBoundary

RightWall AS ChartPlotWall

Visible

Width

X

Y

### Methods

DeletePicture

PastePicture

RevertLayout

Select

### Functions

None

### Events

None

## Lotus Chart: ChartRegression class

The regression (trend) line for a series.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartSeries	Regression

## Lotus Chart: ChartRegression class members

### Properties

Edge AS Border

EndPoint

PlaceInNote

RSquared

Slope

StartPoint

Type

Visible

XAxisMaximum

XAxisMinimum

YIntercept

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartSeriesCollection class

The collection of data series in a chart.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	Series

### Usage

You cannot manipulate a collection of chart series, you can only access attributes of individual series. To access a single series, use the ChartSeries class.

---

{button ,AL('H\_CHARTSERIES\_CLASS ',0)} [See related topics](#)



**Lotus Chart: ChartSeriesCollection class members**

**Properties**

None

**Methods**

None

**Functions**

None

**Events**

None

## Lotus Chart: ChartSeries class

A single series in a chart.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	Series

### Usage

You access an individual series using an index value, such as Series(1), which represents the first series.

## Lotus Chart: ChartSeries class members

### Properties

[ConnectPoints](#)  
[DataLabels](#) AS [ChartDataLabels](#)  
[DataLink](#) AS [ChartDataSource](#)  
[DataPoints](#) AS [ChartDataPoints](#)  
[Edge](#) AS [Border](#)  
[Fill](#) AS [Background](#)  
[HasData](#)  
[HasPicture](#)  
[Index](#)  
[LabelAngle](#)  
[LabelPlacement](#)  
[MarkerSymbol](#)  
[MarkerVisible](#)  
[Name](#)  
[PictureLayout](#)  
[PictureUnits](#)  
[Regression](#) AS [ChartRegression](#)  
[ShowPercentLabels](#)  
[ShowTextLabel](#)s  
[ShowValueLabels](#)  
[TextLabel](#)sLink  
[Type](#)  
[Use2YAxis](#)  
[Visible](#)  
[XValueLink](#)

### Methods

[DeletePicture](#)  
[PastePicture](#)  
[Select](#)

### Functions

None

### Events

None

## Lotus Chart: ChartTableBase class

The base class for number grids and tables under a chart.

### Base classes

None

### Usage

ChartTableBase contains base attributes used both in the ChartTable and ChartNumberGrid classes.

---

{button ,AL('H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLE\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartTableBase class members

### Properties

ColumnHeaders AS ChartDataLabels

Edge AS Border

Fill AS Background

FirstHorizontal AS ChartTableGridline

FirstVertical AS ChartTableGridline

InteriorHorizontals AS ChartTableGridlines

InteriorVerticals AS ChartTableGridlines

RowHeaders AS ChartDataLabels

SeriesData AS ChartTableSeriesCollection

ShowUpperLeft

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartTableGridlines class

A set of grid lines in a chart table or number grid.

### Base classes

ChartGridlines

### Contained by

<u>Class</u>	<u>Property</u>
ChartNumberGrid	InteriorHorizontals, InteriorVerticals
ChartTable	InteriorHorizontals, InteriorVerticals
ChartTableBase	InteriorHorizontals, InteriorVerticals

### Usage

To access the first horizontal or first vertical grid line, use the ChartTableGridline class.

---

{button ,AL('H\_CHARTGRIDLINES\_CLASS ;H\_CHARTTABLEGRIDLINE\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartTableGridlines class members

### Properties

[Edge AS Border](#)  
[ShowInDataArea](#)  
[ShowInHeaderArea](#)  
[Visible](#)

### Methods

[Select](#)

### Functions

None

### Events

None

## Lotus Chart: ChartTableGridline class

A single grid line in a table under a chart or number grid.

### Base classes

ChartGridline

### Contained by

<u>Class</u>	<u>Property</u>
ChartNumberGrid	FirstHorizontal, FirstVertical
ChartTable	FirstHorizontal, FirstVertical
ChartTableBase	FirstHorizontal, FirstVertical

### Usage

To access the all grid lines other than the first horizontal or vertical, use the ChartTableGridlines class.

---

{button ,AL('H\_CHARTGRIDLINES\_CLASS ;H\_CHARTTABLEGRIDLINES\_CLASS ',0)} [See related topics](#)



## Lotus Chart: ChartTableGridline class members

### Properties

[Edge AS Border](#)  
[ShowInDataArea](#)  
[ShowInHeaderArea](#)  
[Visible](#)

### Methods

[Select](#)

### Functions

None

### Events

None

## Lotus Chart: ChartTableSeriesCollection class

Represents all series data in a number grid chart or table.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartTableBase	SeriesData

## Lotus Chart: ChartTableSeriesCollection class members

### Properties

[Font](#)

[Format](#)

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: ChartTable class

A table under a chart.

### Base classes

ChartTableBase

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	Table

---

{button ,AL('H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartTable class members

### Properties

ColumnHeaders AS ChartDataLabels

Edge AS Border

Fill AS Background

FirstHorizontal AS ChartTableGridline

FirstVertical AS ChartTableGridline

InteriorHorizontals AS ChartTableGridlines

InteriorVerticals AS ChartTableGridlines

RowHeaders AS ChartDataLabels

SeriesData AS ChartTableSeriesCollection

ShowUpperLeft

Visible

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartTextEntries class

The entries in a note or title.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartTitleBox	Lines

### Usage

To access a single line of text in a note or title, use the ChartTextEntry class.

---

{button ,AL('H\_CHARTTEXTENTRY\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartTextEntries class members

### Properties

[DataLink](#) AS [ChartDataSource](#)

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: ChartTextEntry class

A single entry in a chart note or title, or the title or subtitle of a chart axis.

### Base classes

ChartTextLabel

### Contained by

<u>Class</u>	<u>Property</u>
ChartAxis	Subtitle, Title
ChartTitleBox	Lines

### Usage

You access an individual entry in a line or note using an index value, such as Lines(1), which represents the first line in a title or note.

To access the collection of text entries in a title or note, use the ChartTextEntries class.

---

{button ,AL('H\_CHARTTEXTENTRIES\_CLASS ;H\_CHARTTEXTLABEL\_CLASS ',0)} [See related topics](#)



## Lotus Chart: ChartTextEntry class members

### Properties

DataLink AS ChartDataSource

Font AS Font

Rotation

Text

Visible

### Methods

Select

### Functions

None

### Events

None

**Lotus Chart: ChartTextLabelLinked class**

Represents an individual, linkable text label.

**Base classes**

None

## Lotus Chart: ChartTextLabels class

A collection of text labels.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartAxis	TickLabels
ChartZAxis	TickLabels

## Lotus Chart: ChartTextLabels class members

### Properties

DataLink AS ChartDataSource

Font AS Font

Visible

### Methods

Select

### Functions

None

### Events

None

**Lotus Chart: ChartTextLabel class**

A text label in a pie slice.

**Base classes**

None

**Usage**

ChartTextLabel contains base attributes used in the ChartTextEntry class.

## Lotus Chart: ChartTextLabel class members

### Properties

Font AS Font

Rotation

Text

Visible

### Methods

Select

### Functions

None

### Events

None

## Lotus Chart: ChartTitleBox class

A chart title or note.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	Note, Title

## Lotus Chart: ChartTitleBox class members

### Properties

DataLink AS ChartDataSource

Edge AS Border

Fill AS Background

Height

Lines AS ChartTextEntries

Lines AS ChartTextEntry

Placement

Visible

Width

X

Y

### Methods

Select

### Functions

None

### Events

None



## Lotus Chart: ChartZAxis class

The z-axis of a 3D chart.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartBase	ZAxis

---

{button ,AL(`H\_CHARTAXIS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ChartZAxis class members

### Properties

Font AS Font

TickLabels AS ChartTextLabels

TickLabelSkip

Visible

### Methods

None

### Functions

None

### Events

None

## Lotus Chart LotusScript A-Z

A  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M

N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z

### A

[Align property](#)  
[ApplyChartStyle method](#)  
[AreValuesManual property](#)  
[AxisPlacement property](#)

### B

[BackColorIndex property](#)  
[Background class](#)  
[BarClusterGapPercent property](#)  
[BarOverlapPercent property](#)  
[BarPlacement property](#)  
[Bold property](#)  
[Border class](#)

### C

[ChartAxis class](#)  
[ChartAxisScale class](#)  
[ChartAxisTitle class](#)  
[ChartBase class](#)  
[ChartDataLabel class](#)  
[ChartDataLabels class](#)  
[ChartDataPoint class](#)  
[ChartDataPoints class](#)  
[ChartDataSource class](#)  
[ChartGridline class](#)

[ChartGridlines class](#)  
[ChartLegend class](#)  
[ChartLegendEntries class](#)  
[ChartMajorGridline class](#)  
[ChartMajorGridlines class](#)  
[ChartNumberGrid class](#)  
[ChartPie class](#)  
[ChartPies class](#)  
[ChartPieSlice class](#)  
[ChartPieSliceGroup class](#)  
[ChartPieSliceGroups class](#)  
[ChartPieSlices class](#)  
[ChartPieTitles class](#)  
[ChartPlot class](#)  
[ChartPlotWall class](#)  
[ChartRegression class](#)  
[ChartSeries class](#)  
[ChartSeriesCollection class](#)  
[ChartTable class](#)  
[ChartTableBase class](#)  
[ChartTableGridline class](#)  
[ChartTableGridlines class](#)  
[ChartTableSeriesCollection class](#)  
[ChartTextEntries class](#)  
[ChartTextEntry class](#)  
[ChartTextLabel class](#)  
[ChartTextLabelLinked class](#)  
[ChartTextLabels class](#)  
[ChartTitleBox class](#)  
[ChartZAxis class](#)  
[ColumnHeaders property](#)  
[ColumnTotals property](#)  
[ColumnTotalsHeader property](#)  
[ConnectPoints property](#)  
[ConnectStackedBarLines property](#)  
[CrossPoint property](#)  
[CrossPointsManual property](#)

## D

[DataArrangement property](#)  
[DataLabels property](#)  
[DataLink property](#)  
[DataPoints property](#)  
[DeletePicture method](#)  
[DepthPercent property](#)

## E

[Edge property](#)  
[Elevation property](#)  
[EndPoint property](#)  
[Entries property](#)  
[ExplodePercent property](#)  
[Exponent property](#)

## F

[FaceName property](#)  
[Fill property](#)  
[FirstHorizontal property](#)  
[FirstVertical property](#)  
[Font property](#)  
[Font class](#)  
[ForeColorIndex property](#)  
[Format property](#)

## G

[GridlineCount property](#)

## H

[HasData property](#)  
[HasPicture property](#)  
[Height property](#)

## I

[Index property](#)  
[InsetLinesFromFrame property](#)  
[InsidePlot property](#)  
[InteriorHorizontals property](#)  
[InteriorVerticals property](#)  
[Is3D property](#)  
[IsAscending property](#)  
[IsDepthEffect property](#)  
[IsHorizontal property](#)  
[IsNumeric property](#)  
[IsStacked property](#)  
[Italic property](#)

## L

[LabelAngle property](#)  
[LabelPlacement property](#)  
[LayoutIsManual property](#)  
[LayoutRowCount property](#)  
[LeftWall property](#)  
[Legend property](#)  
[LightAngle property](#)  
[Lines property](#)  
[LinkUpdateMethod property](#)  
[ListSelection property](#)

## M

[MajorGridlines property](#)  
[MajorInterval property](#)  
[MajorIntervalsManual property](#)  
[MajorTickmarkPlacement property](#)  
[MajorTickmarks property](#)  
[MarkerSymbol property](#)  
[MarkerVisible property](#)  
[MaximumIsManual property](#)  
[Maximum property](#)

[Minimum property](#)  
[MinimumIsManual property](#)  
[MinorGridlines property](#)  
[MinorInterval property](#)  
[MinorIntervalsManual property](#)  
[MinorTickmarkPlacement property](#)  
[MinorTickmarks property](#)

## N

[Name property](#)  
[Note property](#)  
[NumberFormat class](#)  
[NumberGrid property](#)

## O

[OtherSliceText property](#)

## P

[Parentheses property](#)  
[PastePicture method](#)  
[Pattern property](#)  
[PercentLabelFormat property](#)  
[PercentLabelPlacement property](#)  
[PercentLabels property](#)  
[PictureLayout property](#)  
[PictureUnits property](#)  
[Pies property](#)  
[PieSlices property](#)  
[PlaceInNote property](#)  
[Placement property](#)  
[PlacementIsManual property](#)  
[Plain property](#)  
[PlatformFrontFace property](#)  
[PlatformRightFace property](#)  
[PlatformTopFace property](#)  
[Plot property](#)  
[Precision property](#)

## R

[RadarAxisTitles property](#)  
[RadarHasPolygonalBoundary property](#)  
[RadarLabelEachAxis property](#)  
[RadarUseIndependentScales property](#)  
[Regression property](#)  
[RevertLayout method](#)  
[RightWall property](#)  
[Rotation property](#)  
[RowHeaders property](#)  
[RowTotals property](#)  
[RowTotalsHeader property](#)  
[RSquared property](#)

## S

[SaveChartStyle method](#)  
[Scale property](#)  
[ScaleType property](#)  
[ScatterUseSeparateXValues property](#)  
[SecondYAxis property](#)  
[Select method](#)  
[Series property](#)  
[SeriesData property](#)  
[SeriesDataLink property](#)  
[SeriesParse property](#)  
[SetNumericFormat method](#)  
[ShadowPlacement property](#)  
[ShadowSize property](#)  
[ShowInDataArea property](#)  
[ShowInHeaderArea property](#)  
[ShowLabels property](#)  
[ShowPercentLabels property](#)  
[ShowTextLabels property](#)  
[ShowUpperLeft property](#)  
[ShowValueLabels property](#)  
[Size property](#)  
[SliceDirection property](#)  
[SliceExplosion property](#)  
[Slices property](#)  
[SlicesBeforeOther property](#)  
[SliceSort property](#)  
[Slope property](#)  
[Source property](#)  
[StackedTotalsAngle property](#)  
[StackedTotalsFormat property](#)  
[StackedTotalsLabels property](#)  
[StartAngle property](#)  
[StartPoint property](#)  
[StrikeThrough property](#)  
[String property](#)  
[Style property](#)  
[Subtitle property](#)  
[SubtitlesManual property](#)  
[SubtitlePlacement property](#)  
[Subtitles property](#)  
[SubtitlesAreManual property](#)

## T

[Table property](#)  
[Text property](#)  
[TextLabelPlacement property](#)  
[TextLabels property](#)  
[TextLabelsLink property](#)  
[ThreeDElevation property](#)  
[ThreeDLateralRotation property](#)  
[ThreeDLightSource property](#)  
[ThreeDPlatform property](#)  
[ThreeDRowGapPercent property](#)  
[TickLabelCharWidth property](#)  
[TickLabelPlacement property](#)  
[TickLabels property](#)  
[TickLabelSkip property](#)

[TickLabelsLink property](#)  
[TickLabelWidthIsManual property](#)  
[Title property \(ChartBase class\)](#)  
[Title property \(ChartAxis class\)](#)  
[TitlePlacement property](#)  
[Titles property](#)  
[Type property \(NumberFormat class\)](#)  
[Type property \(ChartRegression class\)](#)  
[Type property \(ChartBase and ChartSeries class\)](#)

## U

[Underline property](#)  
[UpdateLinks method](#)  
[Use2YAxis property](#)  
[UseCandlestickForHLCO property](#)  
[UseSeparateLabelRanges property](#)  
[UseSmartLookColors property](#)

## V

[Value property](#)  
[ValueLabelFormat property](#)  
[ValueLabelPlacement property](#)  
[ValueLabels property](#)  
[VertAlign property](#)  
[Visible property](#)

## W

[Width property \(Border class\)](#)  
[Width property \(ChartLegend, ChartNumberGrid, ChartPies, ChartPlot, and ChartTitleBox classes\)](#)

## X

[X property](#)  
[XAxis property](#)  
[XAxisMaximum property](#)  
[XAxisMinimum property](#)  
[XValue property](#)  
[XValueLink property](#)

## Y

[Y property](#)  
[YAxis property](#)  
[YIntercept property](#)

## Z

[ZAxis property](#)



## Lotus Chart Classes

[Background](#)  
[Border](#)  
[ChartAxis](#)  
[ChartAxisScale](#)  
[ChartAxisTitle](#)  
[ChartBase](#)  
[ChartDataLabel](#)  
[ChartDataLabels](#)  
[ChartDataPoint](#)  
[ChartDataPoints](#)  
[ChartDataSource](#)  
[ChartGridline](#)  
[ChartGridlines](#)  
[ChartLegend](#)  
[ChartLegendEntries](#)  
[ChartMajorGridline](#)  
[ChartMajorGridlines](#)  
[ChartNumberGrid](#)  
[ChartPie](#)  
[ChartPies](#)  
[ChartPieSlice](#)  
[ChartPieSliceGroup](#)  
[ChartPieSliceGroups](#)  
[ChartPieSlices](#)  
[ChartPieTitles](#)  
[ChartPlot](#)  
[ChartPlotWall](#)  
[ChartRegression](#)  
[ChartSeries](#)  
[ChartSeriesCollection](#)  
[ChartTable](#)  
[ChartTableBase](#)  
[ChartTableGridline](#)  
[ChartTableGridlines](#)  
[ChartTableSeriesCollection](#)  
[ChartTextEntries](#)  
[ChartTextEntry](#)  
[ChartTextLabel](#)  
[ChartTextLabelLinked](#)  
[ChartTextLabels](#)  
[ChartTitleBox](#)  
[ChartZAxis](#)  
[Font](#)  
[NumberFormat](#)

## **Lotus Chart Methods**

ApplyChartStyle

DeletePicture

PastePicture

RevertLayout

SaveChartStyle

Select

SetNumericFormat

UpdateLinks

## Lotus Chart Properties

J  
J  
J  
J  
J  
J  
J  
J  
J  
J  
J  
J

J  
J  
J  
J  
J  
J  
J  
J  
J  
J  
J  
J  
J

### A

Align  
AreValuesManual  
AxisPlacement

### B

BackColorIndex  
BarClusterGapPercent  
BarOverlapPercent  
BarPlacement  
Bold

### C

ColumnHeaders  
ColumnTotals  
ColumnTotalsHeader  
ConnectPoints  
ConnectStackedBarLines  
CrossPoint  
CrossPointsManual

### D

DataArrangement  
DataLabels  
DataLink

DataPoints  
DepthPercent

## **E**

Edge  
Elevation  
EndPoint  
Entries  
ExplodePercent  
Exponent

## **F**

FaceName  
Fill  
FirstHorizontal  
FirstVertical  
Font  
ForeColorIndex  
Format

## **G**

GridlineCount

## **H**

HasData  
HasPicture  
Height

## **I**

Index  
InsetLinesFromFrame  
InsidePlot  
InteriorHorizontals  
InteriorVerticals  
Is3D  
IsAscending  
IsDepthEffect  
IsHorizontal  
IsNumeric  
IsStacked  
Italic

## **L**

LabelAngle  
LabelPlacement  
LayoutIsManual  
LayoutRowCount  
LeftWall  
Legend  
LightAngle  
Lines  
LinkUpdateMethod

[ListSelection](#)

## **M**

[MajorGridlines](#)

[MajorInterval](#)

[MajorIntervalsManual](#)

[MajorTickmarkPlacement](#)

[MajorTickmarks](#)

[MarkerSymbol](#)

[MarkerVisible](#)

[Maximum](#)

[MaximumIsManual](#)

[Minimum](#)

[MinimumIsManual](#)

[MinorGridlines](#)

[MinorInterval](#)

[MinorIntervalsManual](#)

[MinorTickmarkPlacement](#)

[MinorTickmarks](#)

## **N**

[Name](#)

[Note](#)

[NumberGrid](#)

## **O**

[OtherSliceText](#)

## **P**

[Parentheses](#)

[Pattern](#)

[PercentLabelFormat](#)

[PercentLabelPlacement](#)

[PercentLabels](#)

[PictureLayout](#)

[PictureUnits](#)

[Pies](#)

[PieSlices](#)

[PlaceInNote](#)

[Placement](#)

[PlacementIsManual](#)

[Plain](#)

[PlatformFrontFace](#)

[PlatformRightFace](#)

[PlatformTopFace](#)

[Plot](#)

[Precision](#)

## **R**

[RadarAxisTitles](#)

[RadarHasPolygonalBoundary](#)

[RadarLabelEachAxis](#)

[RadarUseIndependentScales](#)  
[Regression](#)  
[RightWall](#)  
[Rotation](#)  
[RowHeaders](#)  
[RowTotals](#)  
[RSquared](#)

## S

[Scale](#)  
[ScaleType](#)  
[ScatterUseSeparateXValues](#)  
[SecondYAxis](#)  
[Series](#)  
[SeriesData](#)  
[SeriesDataLink](#)  
[SeriesParse](#)  
[ShadowPlacement](#)  
[ShadowSize](#)  
[ShowInDataArea](#)  
[ShowInHeaderArea](#)  
[ShowLabels](#)  
[ShowPercentLabels](#)  
[ShowTextLabel](#)  
[ShowUpperLeft](#)  
[ShowValueLabels](#)  
[Size](#)  
[SliceDirection](#)  
[SliceExplosion](#)  
[Slices](#)  
[SlicesBeforeOther](#)  
[SliceSort](#)  
[Slope](#)  
[Source](#)  
[StackedTotalsAngle](#)  
[StackedTotalsFormat](#)  
[StackedTotalsLabels](#)  
[StartAngle](#)  
[StartPoint](#)  
[StrikeThrough](#)  
[Style](#)  
[Subtitle](#)  
[SubtitlesManual](#)  
[SubtitlePlacement](#)  
[Subtitles](#)  
[SubtitlesAreManual](#)

## T

[Table](#)  
[Text](#)  
[TextLabelPlacement](#)  
[TextLabels](#)  
[TextLabelsLink](#)  
[ThreeDElevation](#)  
[ThreeDLateralRotation](#)  
[ThreeDLightSource](#)

[ThreeDPlatform](#)  
[ThreeDRowGapPercent](#)  
[TickLabelCharWidth](#)  
[TickLabelPlacement](#)  
[TickLabels](#)  
[TickLabelSkip](#)  
[TickLabelsLink](#)  
[TickLabelWidthsManual](#)  
[Title \(ChartBase class\)](#)  
[Title \(ChartAxis class\)](#)  
[TitlePlacement](#)  
[Titles](#)  
[Type \(ChartBase and ChartSeries class\)](#)  
[Type \(NumberFormat class\)](#)  
[Type \(ChartRegression class\)](#)

## U

[Underline](#)  
[Use2YAxis](#)  
[UseCandlestickForHLCO](#)  
[UseSeparateLabelRanges](#)  
[UseSmartLookColors](#)

## V

[Value](#)  
[ValueLabelFormat](#)  
[ValueLabelPlacement](#)  
[ValueLabels](#)  
[VertAlign](#)  
[Visible](#)

## W

[Width \(Border class\)](#)  
[Width \(ChartLegend, ChartNumberGrid, ChartPies, ChartPlot, and ChartTitleBox classes\)](#)

## X

[X](#)  
[XAxis](#)  
[XAxisMaximum](#)  
[XAxisMinimum](#)  
[XValue](#)  
[XValueLink](#)

## Y

[Y](#)  
[YAxis](#)  
[YIntercept](#)

## Z

[ZAxis](#)

## Lotus Chart: ColumnHeaders property

{button ,AL(`H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)}

[See list of classes](#)

(Read-write) The headers of columns in a number grid or table.

### Data type

ChartDataLabels

### Syntax

*chart.Table.ColumnHeaders(num).attribute = value*

or

*chart.NumberGrid.ColumnHeaders(num).attribute = value*

*value = chart.Table.ColumnHeaders(num).attribute*

or

*value = chart.NumberGrid.ColumnHeaders(num).attribute*

### Legal values

The values for *num* in ColumnHeaders can be between 1 and 300.

---

{button ,AL(`H\_CHARTDATALABELS\_CLASS ',0)} [See related topics](#)



## Lotus Chart: ColumnTotalsHeader property

{button ,AL(^H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)}

[See list of classes](#)

(Read-write) The headers for column totals in a number grid.

### Data type

ChartDataLabel

### Syntax

*chart.NumberGrid.ColumnTotalsHeader.attribute = value*

*value = chart.NumberGrid.ColumnTotalsHeader.attribute*

---

{button ,AL(^H\_CHARTDATALABELS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ColumnTotals property

{button ,AL('H\_CHARTNUMBERGRID\_CLASS ',0)} [See list of classes](#)

(Read-write) A column showing the totals in a number grid.

### Data type

ChartDataLabels

### Syntax

*chart.NumberGrid.ColumnTotals.attribute = value*

*value = chart.NumberGrid.ColumnTotals.attribute*

---

{button ,AL('H\_CHARTDATALABELS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ConnectPoints property

{button ,AL(^H\_CHARTSERIES\_CLASS',0)} [See list of classes](#)

(Read-write) Whether to connect the points in a series of a line chart.

### Data type

Integer (Boolean)

### Syntax

*chart*.Series(*num*).ConnectPoints = *trueorfalse*

*trueorfalse* = *chart*.Series(*num*).ConnectPoints

### Legal values

True (1) or False (0)

The values for *num* in the Series can be between 1 and 30.

## Lotus Chart: ConnectStackedBarLines property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_ISSTACKED\_CONNECTSTACKEDBARLINES\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) Whether to show lines connecting bars in a stacked bar chart.

### Data type

Boolean

### Syntax

*chart.ConnectStackedBarLines* = *trueorfalse*

*trueorfalse* = *chart.ConnectStackedBarLines*

### Legal values

True (1) or False (0)

## Lotus Chart: CrossPointIsManual property

{button ,AL(`H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether the point at which the other axis crosses the current axis (the axis intercept) is set manually.

### Data type

Boolean

### Syntax

*chart.axis.Scale.CrossPointIsManual = trueorfalse*

*trueorfalse = chart.axis.Scale.CrossPointIsManual*

### Legal values

True (1) or False (0)

---

{button ,AL(`H\_CROSSPOINT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: CrossPoint property

{button ,AL(^H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) The value at which the opposite axis is drawn. Also known as the axis intercept.

### Data type

Double

### Syntax

*chart.axis.Scale.CrossPoint* = *crosspoint*

*crosspoint* = *chart.axis.Scale.CrossPoint*

## Lotus Chart: DataArrangement property

{button ,AL(^H\_CHARTNUMBERGRID\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to transpose the data in the number grid.

### Data type

Integer (Enumerated)

### Syntax

*chart.NumberGrid.DataArrangement = arrangement*

*arrangement = chart.NumberGrid.DataArrangement*

### Legal values

<u>Value</u>	<u>Definition</u>
\$ByColumns	Parse data columnwise
\$ByRows	Parse data rowwise

## Lotus Chart: DataLabels property

{button ,AL('H\_CHARTSERIES\_CLASS',0)} [See list of classes](#)

(Read-write) Value and percent labels that appear for each data point in a series.

### Data type

ChartDataLabels

### Syntax

*chart*.Series(*num*).DataLabels.*attribute* = *value*

*value* = *chart*.Series(*num*).DataLabels.*attribute*

---

{button ,AL('H\_SHOWVALUELABELS\_PROPERTY\_MEMDEF ;H\_SHOWPERCENTLABELS\_PROPERTY\_MEMDEF ;H\_CHARTDATALABELS\_CLASS',0)} [See related topics](#)



## Lotus Chart: DataLink property

{button ,AL(^H\_CHARTPIETITLE\_CLASS ;H\_CHARTTEXTENTRIES\_CLASS ;H\_CHARTTEXTENTRY\_CLASS ;H\_CHARTSERIES\_CLASS ;H\_CHARTTITLEBOX\_CLASS ;H\_CHARTLEGEND\_CLASS ;H\_CHARTTEXTLABELLINKED\_CLASS::WIN95;' ,0)} [See list of classes](#)

(Read-write) The link source of a chart subobject.

### Data type

ChartDataSource

### Syntax

*object.DataLink.attribute = value*

*value = object.DataLink.attribute*

### Usage

You use the DataLink property to specify the source of the information for the current object. For example, you can specify the source for the text of a chart title using the following syntax:

*chart.Title.DataLink.Source = sourcestring*

You specify the source for the text of the first line of the chart title using this syntax:

*chart.Title.Lines(1).DataLink.Source = sourcestring*

---

{button ,AL(^H\_CHARTDATASOURCE\_CLASS ;H\_SOURCE\_PROPERTY\_MEMDEF ' ,0)} [See related topics](#)

## Lotus Chart: DataPoints property

{button ,AL(`H\_CHARTSERIES\_CLASS`,0)} [See list of classes](#)

{button ,AL(`H\_EXAMPLE\_SERIES\_DATAPOINTS\_PROPERTY\_EXSCRIPT`,1)} [See example](#)

(Read-write) The data points in a chart series.

### Data type

ChartDataPoint

### Syntax

*chart.Series(num).DataPoints(num).attribute = value*

*value = chart.Series(num).DataPoints(num).attribute*

---

{button ,AL(`H\_CHARTDATAPOINT\_CLASS`,0)} [See related topics](#)

## Lotus Chart: DeletePicture method

{button ,AL(^H\_CHARTSERIES\_CLASS ;H\_CHARTPLOT\_CLASS',0)} [See list of classes](#)

Deletes the picture from the plot or specified bar chart series.

### Syntax

*chart*.Plot.DeletePicture

or

*chart*.Series(*num*).DeletePicture

### Returns

Void

---

{button ,AL(^H\_PASTEPICTURE\_METHOD\_MEMDEF ;H\_HASPICTURE\_PROPERTY\_MEMDEF ;H\_PICTUREUNIT  
S\_PROPERTY\_MEMDEF ;H\_PICTURELAYOUT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: DepthPercent property

{button ,AL(^H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) The depth of each pie as a percentage of its radius. Available only for 3D pies.

### Data type

Integer

### Syntax

*chart.Pies.DepthPercent* = *value*

*value* = *chart.Pies.DepthPercent*

### Legal values

1..100

## Lotus Chart: Edge property

{button ,AL(`H\_CHARTAXIS\_CLASS ;H\_CHARTDATAPOINT\_CLASS ;H\_CHARTGRIDLINES\_CLASS ;H\_CHARTGRIDLINE\_CLASS ;H\_CHARTLEGEND\_CLASS ;H\_CHARTMAJORGRIDLINES\_CLASS ;H\_CHARTMAJORGRIDLINE\_CLASS ;H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTPLOTWALL\_CLASS ;H\_CHARTPLOT\_CLASS ;H\_CHARTREGRESSION\_CLASS ;H\_CHARTSERIES\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ;H\_CHARTTITLEBOX\_CLASS ;H\_CHARTTABLEGRIDLINES\_CLASS ;H\_CHARTTABLEGRIDLINE\_CLASSES ',0)} [See list of classes](#)

(Read-write) The edge of an object.

### Data type

Border

### Syntax

*object*.**Edge**.*attribute* = *value*

*value* = *object*.**Edge**.*attribute*

---

{button ,AL(`H\_BORDER\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Elevation property

{button ,AL(^H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) The angle between the viewer's eye and the plane of the pie (at 90 degrees, the pie would appear to be two-dimensional). Available only for 3D pies.

### Data type

Integer

### Syntax

*chart.Pies.Elevation = value*

*value = chart.Pies.Elevation*

### Legal values

5..85 degrees

## Lotus Chart: EndPoint property

{button ,AL('H\_CHARTREGRESSION\_CLASS ',0)} [See list of classes](#)

(Read-Write) The last point for which to calculate the trend (regression) line.

### Data type

Integer

### Syntax

*chart*.Series(*num*).Regression.EndPoint = *datapoint*

*datapoint* = *chart*.Series(*num*).Regression.EndPoint

### Legal values

*datapoint* can be any valid data point in the series.

### Usage

If you do not set the endpoint, the default is the last point in the series.

---

{button ,AL('H\_XAXISMINIMUM\_PROPERTY\_MEMDEF ;H\_XAXISMAXIMUM\_PROPERTY\_MEMDEF ;H\_STARTPOINT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Entries property

{button ,AL(`H\_CHARTLEGEND\_CLASS ',0)} [See list of classes](#)

{button ,AL(`H\_EXAMPLE\_LEGEND\_ENTRIES\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-Write) The entries in the chart legend.

### Data type

ChartLegendEntries

### Syntax

*chart.Legend.Entries(num).attribute = value*

*value = chart.Legend.Entries(num).attribute*

---

{button ,AL(`H\_CHARTLEGENDENTRIES\_CLASS ',0)} [See related topics](#)



' Example: BarClusterGapPercent property

' Set the type of the chart to bar chart

```
myChart.Type = $Bar
```

' Set the bar cluster gap value to 50%

```
myChart.BarClusterGapPercent = 50
```

' Example: Edge property

' Change the properties of the border around the legend  
myChart.Legend.Visible = True

myChart.Legend.Edge.Style = \$Dash

myChart.Legend.Edge.Width = \$ThickBorder

```
' Example: EndPoint and StartPoint properties
' Specify to add a linear regression to the first
' series (in a scatter chart, only)
myChart.Series(1).Regression.Type = $LinearRegression
```

```
' Set the first and last points for which to
' calculate the trend line
```

```
myChart.Series(1).Regression.StartPoint = 3
```

```
myChart.Series(1).Regression.EndPoint = 150
```

```
' Put the regression information in the note, and
' make the note visible
```

```
myChart.Series(1).Regression.PlaceInNote = True
```

```
myChart.Note.Visible = True
```

```
' Example: Font and FaceName properties
' Set the typeface for the first line of the title
mychart.Title.Lines(1).Font.FaceName = "Arial"
mychart.Title.Visible = True
```

```
' Example: Is3D property
' Adds 3D to a line chart
myChart.Type = $Line
myChart.Is3D = TRUE
```

' Example: IsAscending property

' Set the IsAscending property of the Y-axis

```
myChart.YAxis.Scale.IsAscending = TRUE
```

```
' Example: IsStacked and ConnectStackedBarLines properties  
' Set the type to stacked bar chart
```

```
myChart.Type = $Bar  
myChart.IsStacked = TRUE
```

```
' Show connecting lines
```

```
myChart.ConnectStackedBarLines = TRUE
```

```
' Example: Legend and Entries properties

' Set the individual legend entry values

myChart.Legend.Entries(1).Text = "Chocolate"

myChart.Legend.Entries(2).Text = "Vanilla"

myChart.Legend.Entries(3).Text = "Strawberry"
myChart.Legend.Visible = True
```



```
' Example: Lines and Note properties

' Specify the text for the first two lines of the note

myChart.Note.Lines(1).Text = "Collected in June"

myChart.Note.Lines(1).Font.Size = 12
myChart.Note.Lines(2).Text = "By John Smith"
myChart.Note.Lines(2).Font.Size = 10

myChart.Note.Visible = True
```

```
' Example: NumberGrid and ShowUpperLeft properties
' Turns off display of upper-left corner in a number grid
myChart.NumberGrid.ColumnHeaders.Visible = True
myChart.NumberGrid.RowHeaders.Visible = True
myChart.NumberGrid.ShowUpperLeft = False
```

' Example: PastePicture method, PictureUnits and PictureLayout properties

' Paste a picture in the bars of the second series

```
myBarChart.Series(2).PastePicture
```

' Stack pictures in the bars, with each picture equal to 5 data units

```
myBarChart.Series(2).PictureUnits = 5
```

```
myBarChart.Series(2).PictureLayout = $StackedPicture
```

' Example: PercentLabels and PercentLabelPlacement property

' Display percent labels inside pie slices

```
myChart.Pies.PercentLabels.Visible = True
```

```
myChart.Pies.PercentLabelPlacement = $LabelsInside
```

```
' Example: Pies, SliceExplosion, and Subtitles properties
```

```
' Set explosion of pie slices to 25%, and show  
' pie totals (using the Subtitles property)
```

```
myChart.Type = $MultiplePie  
myChart.Pies.SliceExplosion = 25  
myChart.Pies.Subtitles.Visible = True
```

```
' Now increase the size of the pie titles, and make  
' them bold
```

```
myChart.Pies.Titles.Font.Size = 12  
myChart.Pies.Titles.Font.Bold = True
```

```
' Example: Series and DataPoints properties

' Specify that the chart is a bar chart

myChart.Type = $Bar

' Enter data in the chart
for i = 1 to 10
    for j = 1 to 10
        myChart.Series(i).DataPoints(j).Value = i * (j + i)
    next j
next i
```

```
' Example: Size property
' Sets the size of the first line of the title
' to 12 points, and makes it bold
myChart.Title.Lines(1).Font.Size = 12
myChart.Title.Lines(1).Font.Bold = True
```

```
' Example: SubtitlePlacement property
' Specify the text for the XAxis title and subtitle

myChart.XAxis.Title.Text = "Sales"

myChart.XAxis.Subtitle.Text = "Per Quarter"

' Place the subtitle beneath the title

myChart.XAxis.SubtitlePlacement = $SubtitleBeneath
```



```
' Example: Title property

' Specify the text for the first line of the title

' and make the title visible

myChart.Title.Lines(1).Text = "Sales Projections"

myChart.Title.Lines(1).Font.Size = 12

myChart.Title.Visible = TRUE
```

```
' Example: Title and Subtitle properties
' Specify the text for the XAxis title and make it visible

myChart.XAxis.Title.Text = "Sales"

myChart.XAxis.Title.Visible = TRUE

' Specify the text for the XAxis subtitle and make it visible

myChart.XAxis.Subtitle.Text = "Per Quarter"

myChart.XAxis.Subtitle.Visible = TRUE
```

' Example: Type property (ChartBase and ChartSeries classes)

' Specify that the chart is a mixed chart

```
myChart.Type = $Mixed
```

' Set the four series to different types

```
myChart.Series(1).Type = $Bar
```

```
myChart.Series(2).Type = $Area
```

```
myChart.Series(3).Type = $Line
```

```
myChart.Series(4).Type = $Area
```

```
' Example: Value property
' Set the value of the data points for the first
' two series of the chart
mychart.Type = $Bar

myChart.Series(1).DataPoints(1).Value = 4.5
myChart.Series(1).DataPoints(2).Value = 3.7
myChart.Series(1).DataPoints(3).Value = 5.9
myChart.Series(2).DataPoints(1).Value = 3.3
myChart.Series(2).DataPoints(2).Value = 5.2
myChart.Series(2).DataPoints(3).Value = 6.1

' Specify that the y-axis major gridlines have manual values
' and turn off standard gridlines
myChart.YAxis.MajorGridlines.AreValuesManual = True
myChart.YAxis.MajorGridlines.Visible = False

' Specify a single major gridline
myChart.YAxis.MajorGridlines.Count = 1

' Set the value of the gridline to be 4.0
myChart.YAxis.MajorGridlines(1).Value = 4.0
```

```
' Example: Visible property  
  
' Specify the text for the first line of the title  
  
' and make the title visible  
  
myChart.Title.Lines(1).Text = "Sales Projections"  
  
myChart.Title.Visible = TRUE
```

```
' Example: Width property

' Add a thick border to the title

myChart.Title.Lines(1).Text = "Sales Projections"

myChart.Title.Edge.Width = $ThickBorder
```

## Lotus Chart: ExplodePercent property

{button ,AL(`H\_CHARTPIESLICE\_CLASS ',0)} [See list of classes](#)

(Read-write) The percentage to explode a single pie slice.

### Data type

Integer

### Syntax

*chart.PieSlices(num).ExplodePercent = percent*

*percent = chart.PieSlices(num).ExplodePercent*

### Legal values

0..100

---

{button ,AL(`H\_SLICEEXPLOSION\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Exponent property

{button ,AL(`H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to display the subtitle for the axis scale.

### Data type

Boolean

### Syntax

*chart.axis.Scale.Exponent* = *exponent*

*exponent* = *chart.axis.Scale.Exponent*

### Legal values

True (1) or False (0)

---

{button ,AL(`H\_SUBTITLEISMANUAL\_PROPERTY\_MEMDEF ',0)} [See related topics](#)



### **Lotus Chart: FaceName property**

{button ,AL('H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) The font to use for a text string.

#### **Data type**

String

#### **Syntax**

*object*.Font.FaceName = *font*

*font* = *object*.Font.FaceName

## Lotus Chart: Fill property

{button ,AL(^H\_CHARTDATAPOINT\_CLASS ;H\_CHARTLEGEND\_CLASS ;H\_CHARTPIESLICE\_CLASS ;H\_CHART  
PLOTWALL\_CLASS ;H\_CHARTPLOT\_CLASS ;H\_CHARTSERIES\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_  
CHARTTITLEBOX\_CLASS ;H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLE\_CLASS ',0)} [See list of classes](#)

(Read-write) The pattern and color used to fill a solid area (such as a plot, legend, or bar chart bars).

### Data type

Background

### Syntax

*object.Fill.attribute = value*

*value = object.Fill.attribute*

---

{button ,AL(^H\_BACKGROUND\_CLASS ;H\_FORECOLORINDEX\_PROPERTY\_MEMDEF ;H\_BACKCOLORINDEX\_  
PROPERTY\_MEMDEF ;H\_PATTERN\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: FirstHorizontal property

{button ,AL(`H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)}

[See list of classes](#)

(Read-write) Attributes for the first horizontal line in a number grid or table under a chart.

### Data type

ChartTableGridline

### Syntax

*chart.Table.FirstHorizontal.attribute = value*

or

*chart.NumberGrid.FirstHorizontal.attribute = value*

*value = chart.Table.FirstHorizontal.attribute*

or

*value = chart.NumberGrid.FirstHorizontal.attribute*

---

{button ,AL(`H\_CHARTTABLEGRIDLINE\_CLASS ;H\_INTERIORHORIZONTALS\_PROPERTY\_MEMDEF ;H\_INTERIORVERTICALS\_PROPERTY\_MEMDEF ;H\_FIRSTVERTICAL\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: FirstVertical property

{button ,AL(^H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)}

[See list of classes](#)

(Read-write) Attributes for the first vertical line in a number grid or table under a chart.

### Data type

ChartTableGridline

### Syntax

*chart.Table.FirstVertical.attribute = value*

or

*chart.NumberGrid.FirstVertical.attribute = value*

*value = chart.Table.FirstVertical.attribute*

or

*value = chart.NumberGrid.FirstVertical.attribute*

---

{button ,AL(^H\_CHARTTABLEGRIDLINE\_CLASS ;H\_INTERIORHORIZONTALS\_PROPERTY\_MEMDEF ;H\_INTERIORVERTICALS\_PROPERTY\_MEMDEF ;H\_FIRSTHORIZONTAL\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Font class

The text style of an object.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartDataLabels	Font
ChartLegend	Font
ChartLegendEntries	Font
ChartPieTitles	Font
ChartTextEntry	Font
ChartTextLabel	Font
ChartTextLabels	Font
ChartZAxis	Font

## Lotus Chart: Font class members

### Properties

Align  
BackColorIndex  
Bold  
FaceName  
ForeColorIndex  
Italic  
Plain  
Rotation  
Size  
StrikeThrough  
Underline  
VertAlign

### Methods

None

### Functions

None

### Events

None

## Lotus Chart: Font property

{button ,AL(^H\_CHARTDATALABELS\_CLASS ;H\_CHARTLEGENDENTRIES\_CLASS ;H\_CHARTLEGEND\_CLASS ;  
H\_CHARTPIETITLES\_CLASS ;H\_CHARTTEXTENTRY\_CLASS ;H\_CHARTTEXTLABELS\_CLASS ;H\_CHARTTE  
XTLABEL\_CLASS ;H\_CHARTZAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The font of a text value.

### Data type

Font

### Syntax

*object*.**Font**.*attribute* = *value*

*value* = *object*.**Font**.*attribute*

## Lotus Chart: ForeColorIndex property

{button ,AL(^H\_BACKGROUND\_CLASS ;H\_BORDER\_CLASS ;H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) The color for the pattern in a filled object in 1-2-3 and WordPro. In Freelance Graphics, use the color classes.

### Data type

Integer

### Syntax

*object*.ForeColorIndex = *color*

*color* = *object*.ForeColorIndex

### Legal values

0..255

### Usage

The number you specify is an index into the current color palette.

Use ForeColorIndex to set the color of the pattern of a filled object. To set the background of the fill pattern, use the BackColorIndex property. If the pattern is set to a solid color, then only the ForeColorIndex shows up.

---

{button ,AL(^H\_BACKCOLORINDEX\_PROPERTY\_MEMDEF ;H\_PATTERN\_PROPERTY\_MEMDEF ',0)} [See related topics](#)



## Lotus Chart: Format property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The format of numbers.

### Data type

NumberFormat

### Syntax

*object.Format.attribute = value*

*value = object.Format.attribute*

---

{button ,AL('H\_NUMBERFORMAT\_CLASS ',0)} [See related topics](#)

## Lotus Chart: GridlineCount property

{button ,AL(^H\_CHARTMAJORGRIDLINES\_CLASS ',0)} [See list of classes](#)  
(Read-write) The number of manually placed grid lines on the specified axis.

### Data type

Integer

### Syntax

*number* = *chart.axis*.MajorGridLines.GridlineCount

*chart.axis*.MajorGridLines.GridlineCount = *number*

### Legal values

You can specify from 0 to 20 extra grid lines.

### **Lotus Chart: HasData property**

{button ,AL(^H\_CHARTSERIES\_CLASS',0)} [See list of classes](#)

(Read-only) Whether the series has data associated with it, or just style and options settings.

#### **Data type**

Integer (Boolean)

#### **Syntax**

*trueorfalse* = *chart.Series(num).HasData*

#### **Legal values**

True (1) or False (0)

## Lotus Chart: HasPicture property

{button ,AL(^H\_CHARTPLOT\_CLASS ;H\_CHARTSERIES\_CLASS ',0)} [See list of classes](#)

(Read-only) Whether a picture is applied to the series or plot.

### Data type

Integer (Boolean)

### Syntax

*trueorfalse* = *chart.Series(num).HasPicture*

or

*trueorfalse* = *chart.Plot.HasPicture*

### Legal values

True (1) or False (0)

---

{button ,AL(^H\_PASTEPICTURE\_METHOD\_MEMDEF ;H\_DELETEPICTURE\_METHOD\_MEMDEF ;H\_PICTUREUNITS\_PROPERTY\_MEMDEF ;H\_PICTURELAYOUT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Height property

{button ,AL(^H\_CHARTBASE\_CLASS ;H\_CHARTLEGEND\_CLASS ;H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTPIES\_CLASS ;H\_CHARTPLOT\_CLASS ;H\_CHARTTITLEBOX\_CLASS ',0)} [See list of classes](#)

(Read-write) The height of the object.

### Data type

Long

### Syntax

*object.Height = heightvalue*

*heightvalue = object.Height*

### Usage

Height is read-only for the ChartBase class.

---

{button ,AL(^H\_WIDTH\_PROPERTY\_MEMDEF ;H\_X\_PROPERTY\_MEMDEF ;H\_Y\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Index property

{button ,AL(^H\_CHARTMAJORGRIDLINE\_CLASS ;H\_CHARTSERIES\_CLASS ',0)} [See list of classes](#)

(Read-only) The index of the current series or major gridline.

### Data type

Integer

### Syntax

*index* = *chart*.**Series**(*num*).**Index**

or

*index* = *chart.axis*.**MajorGridlines**(*num*).**Index**

## Lotus Chart: InsetLinesFromFrame property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) In a line chart, whether to display a space between the lines and the plot frame, or to extend the lines all the way to the frame.

### Data type

Boolean

### Syntax

*chart*.Legend.InsetLinesFromFrame = *trueorfalse*

*trueorfalse* = *chart*.Legend.InsetLinesFromFrame

### Legal values

True (1) or False (0)

## Lotus Chart: InsidePlot property

{button ,AL(^H\_CHARTLEGEND\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to display the legend inside the plot boundary.

### Data type

Integer (Boolean)

### Syntax

*chart.Legend.InsidePlot = trueorfalse*

*trueorfalse = chart.Legend.InsidePlot*

### Legal values

True (1) or False (0)



## Lotus Chart: InteriorHorizontal property

{button ,AL(`H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)}

[See list of classes](#)

(Read-write) The interior horizontal rules in a number grid or table under a chart.

### Data type

ChartTableGridlines

### Syntax

*chart.Table.InteriorHorizontal.attribute = value*

or

*chart.NumberGrid.InteriorHorizontal.attribute = value*

*value = chart.Table.InteriorHorizontal.attribute*

or

*value = chart.NumberGrid.InteriorHorizontal.attribute*

---

{button ,AL(`H\_CHARTTABLEGRIDLINES\_CLASS ;H\_INTERIORVERTICALS\_PROPERTY\_MEMDEF ;H\_FIRSTVERTICAL\_PROPERTY\_MEMDEF ;H\_FIRSTHORIZONTAL\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: InteriorVerticals property

{button ,AL(`H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)}

[See list of classes](#)

(Read-write) The interior vertical rules in a number grid or table under a chart.

### Data type

ChartTableGridlines

### Syntax

*chart.Table.InteriorVerticals.attribute = value*

or

*chart.NumberGrid.InteriorVerticals.attribute = value*

*value = chart.Table.InteriorVerticals.attribute*

or

*value = chart.NumberGrid.InteriorVerticals.attribute*

---

{button ,AL(`H\_CHARTTABLEGRIDLINES\_CLASS ;H\_INTERIORHORIZONTALS\_PROPERTY\_MEMDEF ;H\_FIRST  
VERTICAL\_PROPERTY\_MEMDEF ;H\_FIRSTHORIZONTAL\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Is3D property

{button ,AL(`H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

{button ,AL(`h\_example\_is3d\_property\_exscript',1)} [See example](#)

(Read-Write) Whether the chart is a three-dimensional chart.

### Data type

Boolean

### Syntax

*chart.Is3D = trueorfalse*

*trueorfalse = chart.Is3D*

### Legal values

True (1) or False (0)

## Lotus Chart: IsAscending property

{button ,AL(^H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

{button ,AL(^H\_EXAMPLE\_ISASCENDING\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) Whether axis values display in ascending or descending order.

### Data type

Boolean

### Syntax

*chart.axis.Scale.IsAscending* = *ascendingValue*

*ascendingValue* = *chart.axis.Scale.IsAscending*

### Legal values

True (1) or False (0)

## Lotus Chart: IsDepthEffect property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether the chart displays with a depth effect.

### Data type

Integer (Boolean)

### Syntax

*chart.IsDepthEffect = trueorfalse*

*trueorfalse = chart.IsDepthEffect*

### Legal values

True (1) or False (0)

## Lotus Chart: IsHorizontal property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether the chart uses a horizontal or vertical orientation.

### Data type

Integer (Boolean)

### Syntax

*chart.IsHorizontal = trueorfalse*

*trueorfalse = chart.IsHorizontal*

### Legal values

True (1) or False (0)

### Usage

Use the IsHorizontal class with bar, line, area, scatter, HLCO, or mixed charts.

## Lotus Chart: IsNumeric property

{button ,AL('H\_CHARTAXISSCALE\_CLASS',0)} [See list of classes](#)

(Readonly) Whether the axis supports scale properties.

### Data type

Integer (Boolean)

### Syntax

*true* or *false* = *chart.axis.Scale.IsNumeric*

### Legal values

True (1) or False (0).

### Usage

Always returns True for the y-axis.

## Lotus Chart: IsStacked property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_ISSTACKED\_CONNECTSTACKEDBARLINES\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) Whether the chart is a stacked chart.

### Data type

Integer (Boolean)

### Syntax

*chart.IsStacked* = *trueorfalse*

*trueorfalse* = *chart.IsStacked*

### Legal values

True (1) or False (0)

### Usage

The IsStacked property applies to bar charts, line charts, area charts, mixed charts, and radar charts.



## Lotus Chart: Italic property

{button ,AL(^H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to display the text in italics.

### Data type

Integer (Boolean)

### Syntax

*object.Font.TextItalic* = *trueorfalse*

*trueorfalse* = *object.Font.TextItalic*

### Legal values

True (1) or False (0)

## Lotus Chart: LabelAngle property

{button ,AL(^H\_CHARTSERIES\_CLASS',0)} [See list of classes](#)

(Read-write) The angle at which to display series labels.

### Data type

Integer

### Syntax

*chart*.Series(*num*).LabelAngle = *angle*

*angle* = *chart*.Series(*num*).LabelAngle

### Legal values

You can enter any integer from 0 to 360 for the angle.

## Lotus Chart: LabelPlacement property

{button ,AL(^H\_CHARTSERIES\_CLASS',0)} [See list of classes](#)

(Read-write) The placement of series labels.

### Data type

Integer (Enumerated)

### Syntax

*chart.Series(num).LabelPlacement = placementValue*

*placementValue = chart.Series(num).LabelPlacement*

### Legal values

<u>Value</u>	<u>Description</u>
\$LabelsCenter	Places labels centered on the data point. (For bar, line, area, and scatter series.)
\$LabelsRight	Places labels to the right of the data point. (For line and scatter series.)
\$LabelsBelow	Places labels below the data point. (For bar, line, area, and scatter series.)
\$LabelsLeft	Places labels to the left of the data point. (For line and scatter series.)
\$LabelsAbove	Places labels above the data point. (For bar, line, area, and scatter series.)

## Lotus Chart: LayoutIsManual property

{button ,AL('H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) Whether the layout of multiple pies is controlled manually or automatically. If LayoutIsManual is False, then the setting of LayoutRowCount is ignored.

### Data type

Integer (Boolean)

### Syntax

*chart.Pies.LayoutIsManual = trueorfalse*

*trueorfalse = chart.Pies.LayoutIsManual*

### Legal values

True (1) or False (0)

---

{button ,AL('H\_LAYOUTROWCOUNT\_PROPERTY\_MEMDEF',0)} [See related topics](#)

## Lotus Chart: LayoutRowCount property

{button ,AL('H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) The number of pies to be plotted in each row of pies in a multiple pie chart. If LayoutIsManual is False, then the setting of LayoutRowCount is ignored.

### Data type

Integer

### Syntax

*chart.Pies.LayoutRowCount = numberofpies*

*numberofpies = chart.Pies.LayoutRowCount*

### Legal values

1..30

---

{button ,AL('H\_LAYOUTISMANUAL\_PROPERTY\_MEMDEF',0)} [See related topics](#)

## Lotus Chart: LeftWall property

{button ,AL('H\_CHARTPLOT\_CLASS ',0)} [See list of classes](#)

(Read-write) The left wall of the platform of a 3D chart or a chart with depth.

### Data type

ChartPlotWall

### Syntax

*chart.Plot.LeftWall.attribute = value*

*value = chart.Plot.LeftWall.attribute*

---

{button ,AL('H\_CHARTPLOTWALL\_CLASS ;H\_RIGHTWALL\_PROPERTY\_MEMDEF ;H\_PLATFORMFRONTFACE\_PROPERTY\_MEMDEF ;H\_PLATFORMTOPFACE\_PROPERTY\_MEMDEF ;H\_PLATFORMRIGHTFACE\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Legend property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_LEGEND\_ENTRIES\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The legend in a chart.

### Data type

ChartLegend

### Syntax

*chart.Legend.attribute = value*

*value = chart.Legend.attribute*

### **Lotus Chart: LightAngle property**

{button ,AL(^H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) The angle at which to show the shadow behind a 3D pie chart or doughnut chart.

#### **Data type**

Integer

#### **Syntax**

*chart.Pies.LightAngle = angle*

*angle = chart.Pies.LightAngle*



## Lotus Chart: Lines property

{button ,AL(^H\_CHARTTITLEBOX\_CLASS ',0)} [See list of classes](#)

{button ,AL(^H\_EXAMPLE\_LINES\_NOTE\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The lines of text in a chart title or note.

### Data type

ChartTextEntries (the collection of lines)

ChartTextEntry (a single line)

### Syntax

*chart.Title.Lines.attribute = value*

or

*chart.Note.Lines.attribute = value*

or

*chart.Title.Lines(num).attribute = value*

or

*chart.Note.Lines(num).attribute = value*

*value = chart.Title.Lines.attribute*

or

*value = chart.Note.Lines.attribute*

or

*value = chart.Title.Lines(num).attribute*

or

*value = chart.Note.Lines(num).attribute*

### Legal values

To specify a single line of text, enter an integer between 1 and 3, inclusive, for *num*.

---

{button ,AL(^H\_CHARTTEXTENTRIES\_CLASS ;H\_CHARTTEXTENTRY\_CLASS ',0)} [See related topics](#)

## Lotus Chart: LinkUpdateMethod property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-Write) Whether to update chart links manually or automatically.

### Data type

Integer (Enumerated)

### Syntax

*chart.LinkUpdateMethod* = *method*

*method* = *chart.LinkUpdateMethod*

### Legal values

<u>Value</u>	<u>Description</u>
\$LinkAutomatic	Update links automatically, whenever the source changes.
\$LinkManual	Update links only when the UpdateLinks method is called.

---

{button ,AL('H\_UPDATELINKS\_METHOD\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: ListSelection property

{button ,AL(`H\_NUMBERFORMAT\_CLASS ',0)} [See list of classes](#)

(Read-write) The format choice for a numeric format type (such as the type of currency or the style of the date).

### Data type

Long

### Syntax

*object*.ListSelection = *selection*

*selection* = *object*.ListSelection

### Legal values

The actual values depend on the format type.

---

{button ,AL(`H\_TYPE\_NUMBERFORMAT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: MajorGridlines property

{button ,AL(`H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The major grid lines for an axis.

### Data type

ChartMajorGridlines (the collection of grid lines)

ChartMajorGridline (a single grid line)

### Syntax

*chart.axis.MajorGridlines.attribute = value*

or

*chart.axis.MajorGridlines(num).attribute = value*

*value = chart.axis.MajorGridlines.attribute*

or

*value = chart.axis.MajorGridlines(num).attribute*

### Legal values

To specify a single gridline, enter an integer between 1 and 3000, inclusive, for *num*.

---

{button ,AL(`H\_CHARTMAJORGRIDLINES\_CLASS ;H\_CHARTMAJORGRIDLINE\_CLASS ',0)} [See related topics](#)

## Lotus Chart: MajorIntervallsManual property

{button ,AL(^H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether the major tick interval is set manually.

### Data type

Boolean

### Syntax

*chart.axis.Scale.MajorIntervallsManual = trueorfalse*

*trueorfalse = chart.axis.Scale.MajorIntervallsManual*

### Legal values

True (1) or False (0)

## Lotus Chart: MajorInterval property

{button ,AL(^H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) The interval between major tick marks on an axis.

### Data type

Double

### Syntax

*chart.axis.Scale.MajorInterval* = *intervalnumber*

*intervalnumber* = *chart.axis.Scale.MajorInterval*

## Lotus Chart: MajorTickmarkPlacement property

{button ,AL(^H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) Where to place the major tick marks.

### Data type

Integer (Enumerated)

### Syntax

*chart.axis*.MajorTickmarkPlacement = *placement*

*placement* = *chart.axis*.MajorTickmarkPlacement

### Legal values

<u>Value</u>	<u>Description</u>
\$TicksInside	Inside the plot lines
\$TicksOutside	Outside the plot lines
\$TicksAcross	Across the plot lines
\$NoTicks	Do not display tick marks

## Lotus Chart: MajorTickmarks property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The major tick marks on an axis.

### Data type

ChartGridlines

### Syntax

*chart.axis.MajorTickmarks.attribute = value*

*value = chart.axis.MajorTickmarks.attribute*

---

{button ,AL('H\_CHARTGRIDLINES\_CLASS ',0)} [See related topics](#)



## Lotus Chart: MarkerSymbol property

{button ,AL(^H\_CHARTSERIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The symbols used for series markers in scatter charts, line charts, and radar charts.

### Data type

Integer (Enumerated)

### Syntax

*chart.Series(num).MarkerSymbol = symbol*

*symbol = chart.Series(num).MarkerSymbol*

### Legal values

<u>Value</u>	<u>Value</u>
\$SquareMarker	\$HDiamondMarker
\$DiamondMarker	\$HTriangleMarker
\$TriangleMarker	\$HCircleMarker
\$CircleMarker	\$HStarMarker
\$StarMarker	\$HDTriangleMarker
\$DTriangleMarker	\$XMarker
\$XSquareMarker	\$CrossMarker
\$CSquareMarker	\$AsteriskMarker
\$ASquareMarker	\$UYMarker
\$UYSquareMarker	\$HorizLineMarker
\$HLSquareMarker	\$VertLineMarker
\$VSquareMarker	\$HSCircleMarker
\$HSquareMarker	\$RBulletMarker
\$CCircleMarker	\$SCircleMarker

## Lotus Chart: MarkerVisible property

{button ,AL(^H\_CHARTSERIES\_CLASS',0)} [See list of classes](#)

(Read-write) Whether to show markers for the specified series.

### Data type

Integer (Boolean)

### Syntax

*chart.Series(num).MarkerVisible = trueorfalse*

*trueorfalse = chart.Series(num).MarkerVisible*

### Legal values

True (1) or False (0)

## Lotus Chart: MaximumIsManual property

{button ,AL(`H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to set the maximum value of the axis manually.

### Data type

Boolean

### Syntax

*chart.axis.Scale.MaximumIsManual = trueorfalse*

*trueorfalse = chart.axis.Scale.MaximumIsManual*

### Legal values

True (1) or False (0)

---

{button ,AL(`H\_MAXIMUM\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Maximum property

{button ,AL('H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) The maximum value of the axis.

### Data type

Double

### Syntax

*chart.axis*.Scale.Maximum = *maximumvalue*

*maximumvalue* = *chart.axis*.Scale.Maximum

---

{button ,AL('H\_MAXIMUMISMANUAL\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: MinimumIsManual property

{button ,AL(`H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to set the minimum value of the axis manually.

### Data type

Boolean

### Syntax

*chart.axis.Scale.MinimumIsManual = trueorfalse*

*trueorfalse = chart.axis.Scale.MinimumIsManual*

### Legal values

True (1) or False (0)

---

{button ,AL(`H\_MINIMUM\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Minimum property

{button ,AL(`H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) The minimum value of the axis.

### Data type

Double

### Syntax

*chart.axis.Scale.Minimum* = *minimumvalue*

*minimumvalue* = *chart.axis.Scale.Minimum*

---

{button ,AL(`H\_MINIMUMISMANUAL\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: MinorGridlines property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The minor grid lines for an axis.

### Data type

ChartGridLines

### Syntax

*chart.axis.MinorGridlines.attribute = value*

*value = chart.axis.MinorGridlines.attribute*

---

{button ,AL('H\_CHARTGRIDLINES\_CLASS ',0)} [See related topics](#)

## Lotus Chart: MinorIntervallsManual property

{button ,AL(^H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether the minor tick interval is set manually.

### Data type

Boolean

### Syntax

*chart.axis.Scale.MinorIntervallsManual = trueorfalse*

*trueorfalse = chart.axis.Scale.MinorIntervallsManual*

### Legal values

True (1) or False (0)



## Lotus Chart: MinorInterval property

{button ,AL(^H\_CHARTAXISSCALE\_CLASS ',0)} [See list of classes](#)

(Read-write) The interval between minor tick marks on an axis.

### Data type

Double

### Syntax

*chart.axis.Scale.MinorInterval* = *intervalnumber*

*intervalnumber* = *chart.axis.Scale.MinorInterval*

## Lotus Chart: MinorTickmarkPlacement property

{button ,AL(^H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-Write) Where to place the minor tickmarks.

### Data type

Integer (Enumerated)

### Syntax

*chart.axis.MinorTickmarkPlacement = placement*

*placement = chart.axis.MinorTickmarkPlacement*

### Legal values

<u>Value</u>	<u>Description</u>
\$TicksInside	Inside the plot lines
\$TicksOutside	Outside the plot lines
\$TicksAcross	Across the plot lines
\$NoTicks	Do not display tick marks

## Lotus Chart: MinorTickmarks property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The minor ticks on an axis.

### Data type

ChartGridlines

### Syntax

*chart.axis.MinorTickmarks.attribute = value*

*value = chart.axis.MinorTickmarks.attribute*

---

{button ,AL('H\_CHARTGRIDLINES\_CLASS ',0)} [See related topics](#)

**Lotus Chart: Name property**

{button ,AL(^H\_CHARTSERIES\_CLASS ',0)} [See list of classes](#)

(Read-only) The name of the current series.

**Data type**

String

**Syntax**

*name* = *chart*.Series(*num*).Name

## Lotus Chart: Note property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_LINES\_NOTE\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The chart note.

### Data type

ChartTitleBox

### Syntax

*chart.Note.attribute = value*

*value = chart.Note.attribute*

---

{button ,AL('H\_CHARTTITLEBOX\_CLASS ',0)} [See related topics](#)

## Lotus Chart: NumberFormat class

The numeric format of an object.

### Base classes

None

### Contained by

<u>Class</u>	<u>Property</u>
ChartAxis	Format
ChartBase	StackedTotalsFormat
ChartPies	PercentLabelFormat, ValueLabelFormat
ChartTableSeriesCollection	Format

## Lotus Chart: NumberFormat class members

### Properties

ListSelection

Parentheses

Precision

String

Type

### Methods

SetNumericFormat

### Functions

None

### Events

None

## Lotus Chart: NumberGrid property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) A number grid chart.

### Data type

[ChartNumberGrid](#)

### Syntax

*chart*.**NumberGrid**.*attribute* = *value*

*value* = *chart*.**NumberGrid**.*attribute*

### Usage

Where *attribute* is a property or contained property of the ChartNumberGrid instance.



### **Lotus Chart: OtherSliceText property**

{button ,AL(^H\_CHARTPIE\_CLASS',0)} [See list of classes](#)

(Read-write) The name for the single combined slice in a pie chart.

#### **Data type**

String

#### **Syntax**

*chart.Pies(num).OtherSliceText = value*

*value = chart.Pies(num).OtherSliceText*

## Lotus Chart: Parentheses property

{button ,AL(^H\_NUMBERFORMAT\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to show parentheses around the value.

### Data type

Integer (Boolean)

### Syntax

*object.Parentheses* = *trueorfalse*

*trueorfalse* = *object.Parentheses*

### Legal values

True (1) or False (0)

## Lotus Chart: PastePicture method

{button ,AL(`H\_CHARTSERIES\_CLASS ;H\_CHARTPLOT\_CLASS',0)} [See list of classes](#)

Pastes the picture on the Clipboard into the plot or specified bar chart series.

### Syntax

*chart*.Plot.PastePicture

or

*chart*.Series(*num*).PastePicture

### Returns

Void

---

{button ,AL(`H\_DELETEPICTURE\_METHOD\_MEMDEF ;H\_HASPICTURE\_PROPERTY\_MEMDEF ;H\_PICTUREUNITS\_PROPERTY\_MEMDEF ;H\_PICTURELAYOUT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Pattern property

{button ,AL(^H\_BACKGROUND\_CLASS ;H\_BORDER\_CLASS;H\_CHARTPLOTWALL\_CLASS;',0)} [See list of classes](#)

(Read-write) The pattern used to fill the wall of a 3D plot, or a border or background.

### Data type

Integer (Enumerated)

### Syntax

*object*.**Pattern** = *value*

*value* = *object*.**Pattern**

### Legal values

<u>Value</u>	<u>Value</u>
\$NoFill	\$DottedSquarehatch
\$SolidForeground	\$WideSquareHatch
\$SolidBackground	\$DarkestGray
\$DoubleRightHatch	\$SecondDarkestGray
\$TripleRightHatch	\$ThirdDarkestGray
\$DoubleCoarseRightHatch	\$FourthDarkestGray
\$SingleCoarseRightHatch	\$FifthLightestGray
\$SingleRightHatch	\$FourthLightestGray
\$BroadRightHatch	\$ThirdLightestGray
\$VertHatch	\$SecondLightestGray
\$VertCoarseHatch	\$LightestGray
\$DoubleLeftHatch	\$Checkers
\$TripleLeftHatch	\$FineCheckers
\$DoubleCoarseLeftHatch	\$BroadLeftHatch
\$SingleCoarseLeftHatch	\$SingleLeftHatch
\$HorizHatch	\$GradVertFG2BG
\$HorizCoarseHatch	\$GradVertBG2FG
\$CrossHatch	\$GradHorizFG2BG
\$CoarseCrossHatch	\$GradHorizBG2FG
\$WideCrossHatch	\$GradFSlashFG2BG
\$FineCrossHatch	\$GradFSlashBG2FG
\$SquareHatch	\$GradBSlashFG2BG
\$CoarseSquareHatch	\$GradBSlashBG2FG

## Lotus Chart: PercentLabelFormat property

{button ,AL('H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The format used for percent labels in a pie chart.

### Data type

NumberFormat

### Syntax

*chart.Pies.PercentLabelFormat.attribute = value*

*value = chart.Pies.PercentLabelFormat.attribute*

---

{button ,AL('H\_NUMBERFORMAT\_CLASS ',0)} [See related topics](#)

## Lotus Chart: PercentLabelPlacement property

{button ,AL('H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_PERCENTLABELS\_PERCENTLABELPLACEMENT\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The placement of the percent labels in pie slices.

### Data type

Integer (Enumerated)

### Syntax

*chart.Pies.PercentLabelPlacement* = *value*

*value* = *chart.Pies.PercentLabelPlacement*

### Legal values

<u>Value</u>	<u>Description</u>
\$LabelsDefault	Places labels at default position.
\$LabelsOutside	Places labels outside pie slices.
\$LabelsInside	Places labels inside pie slices.

## Lotus Chart: PercentLabels property

{button ,AL('H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_PERCENTLABELS\_PERCENTLABELPLACEMENT\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The percent labels in a pie chart.

### Data type

ChartDataLabels

### Syntax

*chart.Pies.PercentLabels.attribute = value*

*value = chart.Pies.PercentLabels.attribute*

---

{button ,AL('H\_CHARTDATALABELS\_CLASS',0)} [See related topics](#)

## Lotus Chart: PictureLayout property

{button ,AL('H\_CHARTSERIES\_CLASS',0)} [See list of classes](#)

(Read-write) How to display a bitmap pasted into a bar series.

### Data type

Integer (Enumerated)

### Syntax

*chart*.Series(*num*).PictureLayout = *value*

*value* = *chart*.Series(*num*).PictureLayout

### Legal values

<u>Value</u>	<u>Description</u>
\$IsoStackedPicture	Maintain the correct aspect ratio for the picture.
\$StretchedPicture	Stretch the picture to fill the bar.
\$StackedPicture	Fill the bars with pictures, with one picture equal to the number of units specified in PictureUnits.

---

{button ,AL('H\_PICTUREUNITS\_PROPERTY\_MEMDEF ;H\_PASTEPICTURE\_METHOD\_MEMDEF ;H\_HASPICTURE\_PROPERTY\_MEMDEF ;H\_DELETEPICTURE\_METHOD\_MEMDEF',0)} [See related topics](#)



## Lotus Chart: PictureUnits property

{button ,AL(`H\_CHARTSERIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The number of units to use when stacking pictures in a bar series.

### Data type

Double

### Syntax

*chart*.Series(*num*).PictureUnits = *number*

*number* = *chart*.Series(*num*).PictureUnits

### Usage

When the PictureLayout is \$StackedPicture, then each picture is equal to the units specified in this property.

---

{button ,AL(`H\_PICTURELAYOUT\_PROPERTY\_MEMDEF ;H\_PASTEPICTURE\_METHOD\_MEMDEF ;H\_HASPICTURE\_PROPERTY\_MEMDEF ;H\_DELETEPICTURE\_METHOD\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: PieSlices property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The slices in a pie chart or doughnut chart.

### Data type

ChartPieSlices

### Syntax

*chart.PieSlices(num).attribute = value*

*value = chart.PieSlices(num).attribute*

---

{button ,AL('H\_CHARTPIESLICE\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Pies property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The pies in a pie chart or doughnuts in a doughnut chart.

### Data type

ChartPies class

### Syntax

*chart.Pies.attribute = value*

*value = chart.Pies.attribute*

---

{button ,AL('H\_CHARTPIES\_CLASS ',0)} [See related topics](#)

### **Lotus Chart: PlacelnNote property**

{button ,AL('H\_CHARTREGRESSION\_CLASS ',0)} [See list of classes](#)  
(Read-write) Whether to display the regression results in the chart note.

#### **Data type**

Integer (Boolean)

#### **Syntax**

*chart.Series(num).Regression.PlacelnNote = trueorfalse*

*trueorfalse = chart.Series(num).Regression.PlacelnNote*

#### **Legal values**

True (1) or False (0)

## Lotus Chart: PlacementIsManual property

{button ,AL(`H\_CHARTNUMBERGRID\_CLASS ',0)} [See list of classes](#)

(Read-write) To restore the number grid size and position.

### Data type

Integer (Boolean)

### Syntax

*chart.NumberGrid.PlacementIsManual* = *trueorfalse*

*trueorfalse* = *chart.NumberGrid.PlacementIsManual*

### Usage

When you want to return to the number grid to its original size and position, set the value to *false*.

---

{button ,AL(`H\_CHARTDATALABELS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Placement property

{button ,AL(^H\_CHARTLEGEND\_CLASS ;H\_CHARTPIETITLES\_CLASS ;H\_CHARTTITLEBOX\_CLASS ',0)} [See list of classes](#)

(Read-write) The placement of the legend, title, pie titles, or note.

### Data type

Integer (Enumerated)

### Syntax

*chart.Legend.Placement = placement*

or

*chart.Title.Placement = placement*

or

*chart.Note.Placement = placement*

or

*chart.Pies.Titles.Placement = placement*

or

*chart.Pies.Subtitles.Placement = placement*

*placement = chart.Legend.Placement*

or

*placement = chart.Title.Placement*

or

*placement = chart.Note.Placement*

or

*placement = chart.Pies.Titles.Placement*

or

*placement = chart.Pies.Subtitles.Placement*

### Legal values

For titles, notes, and legends:

<u>Value</u>	<u>Description</u>
\$Northwest	Places the object in the top-left corner of the chart frame.
\$North	Places the object at the top of the chart frame.
\$Northeast	Places the object in the top-right corner of the chart frame.
\$West	Places the object on the left side of the chart frame.
\$East	Places the object on the right side of the chart frame.
\$SouthWest	Places the object in the bottom-left corner of the chart frame.
\$South	Places the object at the bottom of the chart frame.
\$SouthEast	Places the object in the bottom-right corner of the chart frame.
\$CustomPlacement	Allows the object to be placed manually.

For pie titles and subtitles:

<u>Value</u>	<u>Description</u>
\$TitleAbove	Places object above the pies.
\$TitleBelow	Places object below the pies.



## Lotus Chart: Plain property

{button ,AL(^H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether the text is shown without any special formatting (such as bold, italics, underlining).

### Data type

Boolean

### Syntax

*object.Font.Plain = trueorfalse*

*trueorfalse = object.Font.Plain*

### Legal values

True (1) or False (0)



## Lotus Chart: PlatformFrontFace property

{button ,AL('H\_CHARTPLOT\_CLASS ',0)} [See list of classes](#)

(Read-write) The front face of a 3D chart.

### Data type

ChartPlotWall

### Syntax

*chart.Plot.PlatformFrontFace.attribute = value*

*value = chart.Plot.PlatformFrontFace.attribute*

---

{button ,AL('H\_CHARTPLOTWALL\_CLASS ;H\_RIGHTWALL\_PROPERTY\_MEMDEF ;H\_LEFTWALL\_PROPERTY\_MEMDEF ;H\_PLATFORMTOPFACE\_PROPERTY\_MEMDEF ;H\_PLATFORMRIGHTFACE\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: PlatformRightFace property

{button ,AL('H\_CHARTPLOT\_CLASS ',0)} [See list of classes](#)

(Read-write) The right face of a 3D chart.

### Data type

ChartPlotWall

### Syntax

*chart.Plot.PlatformRightFace.attribute = value*

*value = chart.Plot.PlatformRightFace.attribute*

---

{button ,AL('H\_CHARTPLOTWALL\_CLASS ;H\_RIGHTWALL\_PROPERTY\_MEMDEF ;H\_LEFTWALL\_PROPERTY\_MEMDEF ;H\_PLATFORMFRONTFACE\_PROPERTY\_MEMDEF ;H\_PLATFORMTOPFACE\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: PlatformTopFace property

{button ,AL('H\_CHARTPLOT\_CLASS ',0)} [See list of classes](#)

(Read-write) The top face of a 3D or with-depth chart.

### Data type

ChartPlotWall

### Syntax

*chart.Plot.PlatformTopFace.attribute = value*

*value = chart.Plot.PlatformTopFace.attribute*

---

{button ,AL('H\_CHARTPLOTWALL\_CLASS ;H\_RIGHTWALL\_PROPERTY\_MEMDEF ;H\_LEFTWALL\_PROPERTY\_MEMDEF ;H\_PLATFORMFRONTFACE\_PROPERTY\_MEMDEF ;H\_PLATFORMRIGHTFACE\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Plot property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The plot of the chart.

### Data type

ChartPlot

### Syntax

*chart.Plot.attribute = value*

*value = chart.Plot.attribute*

---

{button ,AL('H\_CHARTPLOT\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Precision property

{button ,AL(^H\_NUMBERFORMAT\_CLASS ',0)} [See list of classes](#)

(Read-write) The precision with which to display the number.

### Data type

Integer

### Syntax

*object.Precision* = *value*

*value* = *object.Precision*

## Lotus Chart: RadarAxisTitles property

{button ,AL(^H\_CHARTBASE\_CLASS;'0)} [See list of classes](#)

(Read-write) Controls the visibility and font of the axis titles on a radar chart.

### Data type

[ChartTextLabels](#)

### Syntax

*value* = *radarchart.RadarAxisTitles.attribute*

*radarchart.RadarAxisTitles.attribute* = *value*

## Lotus Chart: RadarHasPolygonalBoundary property

{button ,AL('H\_CHARTPLOT\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether the radar chart uses a polygonal or circular plot.

### Data type

Integer (Boolean)

### Syntax

*radarchart.Plot.RadarHasPolygonalBoundary = trueorfalse*

*trueorfalse = radarchart.Plot.RadarHasPolygonalBoundary*

### Legal values

True (1) or False (0)

## Lotus Chart: RadarLabelEachAxis property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to show data labels on each axis in a radar chart.

### Data type

Boolean

### Syntax

*radarchart.RadarLabelEachAxis = trueorfalse*

*trueorfalse = radarchart.RadarLabelEachAxis*

### Legal values

True (1) or False (0)



## Lotus Chart: RadarUseIndependentScales property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to use separate scales for each axis in a radar chart.

### Data type

Boolean

### Syntax

*radarchart.RadarUseIndependentScales = trueorfalse*

*trueorfalse = radarchart.RadarUseIndependentScales*

### Legal values

True (1) or False (0)

## Lotus Chart: Regression property

{button ,AL('H\_CHARTSERIES\_CLASS',0)} [See list of classes](#)

(Read-write) The regression (trend) line for a series.

### Data type

ChartRegression

### Syntax

*chart*.**Series**(*num*).**Regression**.*attribute* = *value*

*value* = *chart*.**Series**(*num*).**Regression**.*attribute*

---

{button ,AL('H\_CHARTREGRESSION\_CLASS ;H\_SLOPE\_PROPERTY\_MEMDEF ;H\_RSQUARED\_PROPERTY\_M  
EMDEF ;H\_YINTERCEPT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## **Lotus Chart: RevertLayout method**

{button ,AL(^H\_NUMBERFORMAT\_CLASS',0)} [See list of classes](#)

Restore the original layout of the plot.

### **Syntax**

*chart*.Plot.RevertLayout

### **Return values**

Void

## Lotus Chart: RightWall property

{button ,AL('H\_CHARTPLOT\_CLASS ',0)} [See list of classes](#)

(Read-write) The right wall of the plot in a 3D or with-depth chart.

### Data type

ChartPlotWall

### Syntax

*chart.Plot.RightWall.attribute* = *value*

*value* = *chart.Plot.RightWall.attribute*

---

{button ,AL('H\_CHARTPLOTWALL\_CLASS ;H\_LEFTWALL\_PROPERTY\_MEMDEF ;H\_PLATFORMFRONTFACE\_P  
ROPERTY\_MEMDEF ;H\_PLATFORMTOPFACE\_PROPERTY\_MEMDEF ;H\_PLATFORMRIGHTFACE\_PROPER  
TY\_MEMDEF ',0)} [See related topics](#)

### **Lotus Chart: Rotation property**

{button ,AL('H\_CHARTTEXTENTRY\_CLASS ;H\_CHARTTEXTLABEL\_CLASS ;H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) The angle at which to display text.

**Note** Provided for compatibility; to be implemented in a future release.

## Lotus Chart: RowHeaders property

{button ,AL(`H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)}

[See list of classes](#)

(Read-write) The headers for rows in a number grid or data table.

### Data type

ChartDataLabels (collection of row headers)

### Syntax

*chart.Table.RowHeaders.attribute = value*

or

*chart.NumberGrid.RowHeaders.attribute = value*

*value = chart.Table.RowHeaders.attribute*

or

*value = chart.NumberGrid.RowHeaders.attribute*

---

{button ,AL(`H\_CHARTDATALABELS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: RowTotalsHeader property

{button ,AL(^H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)}

[See list of classes](#)

(Read-write) The headers for row totals in a number grid.

### Data type

[ChartDataLabels](#)

### Syntax

*chart.NumberGrid.RowTotalsHeader.attribute = value*

*value = chart.NumberGrid.RowTotalsHeader.attribute*

### Usage

Where attribute is a property or contained property of the ChartDataLabels instance.

## Lotus Chart: RowTotals property

{button ,AL('H\_CHARTNUMBERGRID\_CLASS ',0)} [See list of classes](#)

(Read-write) The row totals in a number grid.

### Data type

ChartDataLabels (collection of row totals)

### Syntax

*chart.NumberGrid.RowTotals.attribute = value*

*value = chart.NumberGrid.RowTotals.attribute*

---

{button ,AL('H\_CHARTDATALABELS\_CLASS ',0)} [See related topics](#)



## **Lotus Chart: RSquared property**

{button ,AL(^H\_CHARTREGRESSION\_CLASS ',0)} [See list of classes](#)

(Read-only) The RSquared value for a regression line.

### **Data type**

Double

### **Syntax**

*rsquaredValue* = *chart.Series(num).Regression.RSquared*

## Lotus Chart: SaveChartStyle method

{button ,AL(`H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

Saves the current look of the chart as a new style.

### Syntax

*chart*.SaveChartStyle(*stylename*)

### Parameters

*stylename*

The file name to assign to the chart style, specified as a string.

### Returns

Void

---

{button ,AL(`H\_APPLYCHARTSTYLE\_METHOD\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: ScaleType property

{button ,AL(^H\_CHARTAXISSCALE\_CLASS ;H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The type of scaling to use for an axis or pie charts.

### Data type

Integer (Enumerated)

### Syntax

*chart.axis.Scale.ScaleType* = *scalingtype*

or

*chart.Pies.ScaleType* = *scalingtype*

*scalingtype* = *chart.axis.Scale.ScaleType*

or

*scalingtype* = *chart.Pies.ScaleType*

### Legal values

For axis scales:

<u>Value</u>	<u>Description</u>
\$LinearScale	Uses standard, linear scale
\$PercentScale	Uses a 100% scale
\$LogScale	Uses a logarithmic scale

For pie charts:

<u>Value</u>	<u>Description</u>
\$Unscaled	Shows all pies the same size
\$ScaleByArea	Scales the pies by using greater area for larger pies
\$ScaleByHeight	Scales the pies by using greater height for larger pies, for 3D pie charts only

## Lotus Chart: Scale property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The scale of the axis.

### Data type

ChartAxisScale

### Syntax

*chart.axis*.**Scale**.*attribute* = *value*

*value* = *chart.axis*.**Scale**.*attribute*

---

{button ,AL('H\_CHARTAXISSCALE\_CLASS ',0)} [See related topics](#)

## Lotus Chart: ScatterUseSeparateXValues property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether scatter chart series use separate x values. If true, each plotted series is associated with its own range of x-values.

### Data type

Boolean

### Syntax

*chart.ScatterUseSeparateXValues = trueorfalse*

*trueorfalse = chart.ScatterUseSeparateXValues*

### Legal values

True (1) or False (0)

## Lotus Chart: SecondYAxis property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Specifies the attributes for the 2nd y-axis in a chart.

### Data type

ChartAxis

### Syntax

*chart.SecondYAxis.attribute = value*

*value = chart.SecondYAxis.attribute*

---

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See related topics](#)

## **Lotus Chart: Select method**

Selects a chart part so the user can manipulate it.

### **Syntax**

*object*.Select

### **Returns**

Void

### **Usage**

When writing a script to manipulate charts, you do not need to select a chart part using this method. Use the Select method to select a portion of the chart for user manipulation.

## Lotus Chart: SeriesDataLink property

{button ,AL(^H\_CHARTBASE\_CLASS',0)} [See list of classes](#)

(Read-write) The link source of a chart series.

### Data type

ChartDataSource

### Syntax

*value* = *chart*.SeriesDataLink.Source

*chart*.SeriesDataLink.Source = *value*



## Lotus Chart: SeriesData property

{button ,AL(^H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTTABLEBASE\_CLASS ;H\_CHARTTABLE\_CLASS ',0)}

[See list of classes](#)

(Read-write) The display of a numeric series in a table or number grid.

### Data type

ChartTableSeriesCollection

### Syntax

*chart.Table.SeriesData(num).attribute = value*

or

*chart.NumberGrid.SeriesData(num).attribute= value*

*value = chart.Table.SeriesData(num).attribute= value*

or

*value = chart.NumberGrid.SeriesData(num).attribute*

### Legal values

1..30 for specifying a single series.

---

{button ,AL(^H\_CHARTDATALABELS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: SeriesParse property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether individual series correspond to rows, columns, or arbitrary ranges of the data source.

### Data type

Integer (Enumerated)

### Syntax

*chart*.SeriesParse = *parsevalue*

*parsevalue* = *chart*.SeriesParse

### Legal values

<u>Value</u>	<u>Description</u>
\$ByColumns	Series data is stored in columns
\$ByRows	Series data is stored in rows
\$Individually	Series data is stored individually

## Lotus Chart: Series property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_SERIES\_DATAPOINTS\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The series in a chart.

### Data type

ChartSeries

### Syntax

*chart.Series(num).attribute = value*

*value = chart.Series(num).attribute*

### Legal values

1..30 for specifying a single series.

---

{button ,AL('H\_CHARTSERIES\_CLASS ',0)} [See related topics](#)

## Lotus Chart: SetNumericFormat method

{button ,AL(^H\_NUMBERFORMAT\_CLASS',0)} [See list of classes](#)

### Syntax

*chart*.SetNumericFormat

### Return values

Void

## Lotus Chart: ShadowPlacement property

{button ,AL(^H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The placement of the shadow in a 3D pie or doughnut chart.

### Data type

Integer (Enumerated)

### Syntax

*chart.Pies.ShadowPlacement* = *placementvalue*

*placementvalue* = *chart.Pies.ShadowPlacement*

### Legal values

<u>Value</u>	<u>Description</u>
\$ShadowNone	Show no shadow
\$ShadowFront	Show a shadow in front of the pie or doughnut chart
\$ShadowRight	Show a shadow to the right of the chart
\$ShadowLeft	Show a shadow to the left of the chart

---

{button ,AL(^H\_SHADOWSIZE\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: ShadowSize property

{button ,AL(^H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The size of the shadow to show in a pie or doughnut chart.

### Data type

Integer (Enumerated)

### Syntax

*chart.Pies.ShadowSize* = *shadowsize*

*shadowsize* = *chart.Pies.ShadowSize*

### Legal values

<u>Value</u>	<u>Description</u>
\$ShortShadow	Display a small shadow
\$MediumShadow	Display a medium-sized shadow
\$LongShadow	Display a large shadow

### Usage

This value takes effect only when the ShadowPlacement property is not set to ShadowNone.

---

{button ,AL(^H\_SHADOWPLACEMENT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: ShowInDataArea property

{button ,AL(`H\_CHARTTABLEGRIDLINES\_CLASS ;H\_CHARTTABLEGRIDLINE\_CLASS ',0)} [See list of classes](#)  
(Read-write) Whether to show gridlines in the data area of a number grid or table under a chart.

### Data type

Integer (Boolean)

### Syntax

*chart.NumberGrid.gridline.ShowInDataArea* = trueorfalse  
or  
*chart.Table.gridline.ShowInDataArea* = trueorfalse

*trueorfalse* = *chart.Table.gridline.ShowInDataArea*  
or  
*trueorfalse* = *chart.NumberGrid.gridline.ShowInDataArea*

### Legal values

True (1) or False (0)

### Usage

The values for *gridline* can be FirstHorizontal, FirstVertical, InteriorHorizontal, or InteriorVertical. For example, to turn off gridlines in the data area for the first vertical line in a table under a chart, you'd use a statement like the following:

```
mychart.Table.FirstVertical.ShowInDataArea = False
```

---

{button ,AL(`H\_SHOWINHEADERAREA\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: ShowInHeaderArea property

{button ,AL('H\_CHARTTABLEGRIDLINES\_CLASS ;H\_CHARTTABLEGRIDLINE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to show gridlines in the header area of a number grid or table under a chart.

### Data type

Integer (Boolean)

### Syntax

*chart.NumberGrid.gridline.ShowInHeaderArea* = *trueorfalse*

or

*chart.Table.gridline.ShowInHeaderArea* = *trueorfalse*

*trueorfalse* = *chart.Table.gridline.ShowInHeaderArea*

or

*trueorfalse* = *chart.NumberGrid.gridline.ShowInHeaderArea*

### Legal values

True (1) or False (0)

### Usage

The values for *gridline* can be FirstHorizontal, FirstVertical, InteriorHorizontal, or InteriorVertical. For example, to turn off gridlines in the header area for the interior vertical lines in a table under a chart, you'd use a statement like the following:

```
mychart.Table.InteriorVerticals.ShowInHeaderArea = False
```

---

{button ,AL('H\_SHOWINDATAAREA\_PROPERTY\_MEMDEF ',0)} [See related topics](#)



## Lotus Chart: ShowLabels property

{button ,AL(^H\_CHARTMAJORGRIDLINES\_CLASS',0)} [See list of classes](#)  
(Read-write) Controls the display of labels for the major gridlines on an axis.

### Data type

Integer (Boolean)

### Syntax

*chart.axis.MajorGridlines.ShowLabels = trueorfalse*

*trueorfalse = chart.axis.MajorGridlines.ShowLabels*

### Legal values

True (1) or False (0)

### Usage

ShowLabels is available for XAxis, YAxis, or SecondYAxis (if defined).

## Lotus Chart: ShowPercentLabels property

{button ,AL(`H\_CHARTSERIES\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to show percent labels for each series in a chart.

### Data type

Integer (Boolean)

### Syntax

*chart*.Series(*num*).ShowPercentLabels = *trueorfalse*

*trueorfalse* = *chart*.Series(*num*).ShowPercentLabels

### Legal values

True (1) or False (0)

---

{button ,AL(`H\_SHOWVALUELABELS\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: ShowTextLabels property

{button ,AL('H\_CHARTSERIES\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to show text labels for each series in a chart.

### Data type

Integer (Boolean)

### Syntax

*chart*.Series(*num*).ShowTextLabels = *trueorfalse*

*trueorfalse* = *chart*.Series(*num*).ShowTextLabels

### Legal values

True (1) or False (0)

---

{button ,AL('H\_SHOWPERCENTLABELS\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: ShowUpperLeft property

```
{button ,AL(^H_CHARTNUMBERGRID_CLASS ;H_CHARTTABLEBASE_CLASS ;H_CHARTTABLE_CLASS ',0)}
```

[See list of classes](#)

(Read-write) Whether to show the lines around the upper-left corner of the number grid or table under a chart.

### Data type

Integer (Boolean)

### Syntax

*chart.NumberGrid.ShowUpperLeft = trueorfalse*

or

*chart.Table.ShowUpperLeft = trueorfalse*

*trueorfalse = chart.NumberGrid.ShowUpperLeft*

or

*trueorfalse = chart.Table.ShowUpperLeft*

### Legal values

True (1) or False (0)

## Lotus Chart: ShowValueLabels property

{button ,AL('H\_CHARTSERIES\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to show value labels for each series in a chart.

### Data type

Integer (Boolean)

### Syntax

*chart*.Series(*num*).ShowValueLabels = *trueorfalse*

*trueorfalse* = *chart*.Series(*num*).ShowValueLabels

### Legal values

True (1) or False (0)

---

{button ,AL('H\_SHOWPERCENTLABELS\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

### **Lotus Chart: Size property**

{button ,AL('H\_FONT\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_SIZE\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The size of the text string.

### **Data type**

Integer

### **Syntax**

*object*.Font.Size = *size*

*size* = *object*.Font.Size

### **Usage**

The integer value specifies the font size in points.

## Lotus Chart: SliceDirection property

{button ,AL('H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The direction of pie slices or doughnut slices.

### Data type

Integer (Enumerated)

### Syntax

*chart.Pies.SliceDirection* = *direction*

*direction* = *chart.Pies.SliceDirection*

### Legal values

<u>Value</u>	<u>Description</u>
\$CounterClockwise	Display slices counter-clockwise
\$Clockwise	Display slices clockwise

---

{button ,AL('H\_STARTANGLE\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: SliceExplosion property

{button ,AL(`H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The amount to explode the slices in a pie or doughnut chart.

### Data type

Integer

### Syntax

*chart.Pies.SliceExplosion* = *explodevalue*

*explodevalue* = *chart.Pies.SliceExplosion*

### Legal values

0..100, with 0 meaning the slices are not exploded at all, and 100 showing the slices completely exploded (with the innermost tips of the slices at the outer edge of the pie or doughnut).

---

{button ,AL(`H\_EXPLODEPERCENT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)



## Lotus Chart: SlicesBeforeOther property

{button ,AL(`H\_CHARTPIE\_CLASS',0)} [See list of classes](#)

The number of uncombined slices in a pie chart.

### Data type

Integer

### Syntax

*chart.Pies(num).SlicesBeforeOther = slices*

### Legal values

For a legible chart, you should limit the uncombined slices to x or fewer.

---

{button ,AL(`H\_CHARTPIES\_CLASS',0)} [See related topics](#)

## Lotus Chart: SliceSort property

{button ,AL(^H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) How to sort the slices by size in a pie or doughnut chart.

### Data type

Integer (Enumerated)

### Syntax

*chart.Pies.SliceSort = sortmethod*

*sortmethod = chart.Pies.SliceSort*

### Legal values

<u>Value</u>	<u>Description</u>
\$Unsorted	Do not sort the pies or doughnuts (show slices in the order used in the data)
\$SortIndividually	Sort each pie or doughnut separately by size
\$SortByFirst	Sort all pies or doughnuts by the size order in the first chart
\$SortByLast	Sort all pies or doughnuts by the size order in the last chart

## Lotus Chart: Slices property

{button ,AL('H\_CHARTPIE\_CLASS',0)} [See list of classes](#)

(Read-write) Slices of an individual pie.

### Data type

[ChartPieSlices](#)

### Syntax

*chart.Pies(n).Slices(n).attribute = attribute*

*attribute = chart.Pies(n).Slices(n).attribute*

### Usage

Where *attribute* is a property or contained property of the ChartPieSlices instance.

## Lotus Chart: Slope property

{button ,AL(^H\_CHARTREGRESSION\_CLASS ',0)} [See list of classes](#)

(Read-only) The slope of the regression line.

### Data type

Double

### Syntax

*slopevalue* = **chart.Series(num).Regression.Slope**

## Lotus Chart: Source property

{button ,AL('H\_CHARTDATASOURCE\_CLASS ',0)} [See list of classes](#)

(Read-write) The name of the file and the link containing the data.

### Data type

String

### Syntax

*object*.Source = *link*

*link* = *object*.Source

---

{button ,AL('H\_DATA LINK\_PROPERTY\_MEMDEF;H\_TICKLABELSLINK\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: StackedTotalsAngle property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The angle at which to display totals on a stacked bar chart.

### Data type

Integer

### Syntax

*chart*.StackedTotalsAngle = *degrees*

*degrees* = *chart*.StackedTotalsAngle

### Legal values

0..360 degrees

## Lotus Chart: StackedTotalsFormat property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The format to use when showing totals values for bars in a stacked bar chart.

### Data type

NumberFormat

### Syntax

*chart*.**StackedTotalsFormat**.*attribute* = *value*

*value* = *chart*.**StackedTotalsFormat**.*attribute*

---

{button ,AL('H\_NUMBERFORMAT\_CLASS ',0)} [See related topics](#)

## Lotus Chart: StackedTotalsLabels property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The labels used to mark the totals in a stacked bar chart.

### Data type

ChartDataLabels

### Syntax

*chart*.**StackedTotalsLabels**.*attribute* = *value*

*value* = *chart*.**StackedTotalsLabels**.*attribute*

---

{button ,AL('H\_CHARTDATALABELS\_CLASS ',0)} [See related topics](#)



## Lotus Chart: StartAngle property

{button ,AL(`H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) The location at which the first slice in the pie displays.

### Data type

Integer

### Syntax

*chart.Pies.StartAngle* = *angle*

*angle* = *chart.Pies.StartAngle*

### Legal values

0..360 degrees

---

{button ,AL(`H\_SLICEDIRECTION\_PROPERTY\_MEMDEF',0)} [See related topics](#)

## Lotus Chart: StartPoint property

{button ,AL('H\_CHARTREGRESSION\_CLASS ',0)} [See list of classes](#)

(Read-write) The first point at which to calculate the trend (regression) line.

### Data type

Integer

### Syntax

*chart*.Series(*num*).Regression.StartPoint = *startingindex*

*startingindex* = *chart*.Series(*num*).Regression.StartPoint

### Legal values

1..8192

### Usage

The value you use for the StartPoint is the data point index for the starting data value.

---

{button ,AL('H\_ENDPOINT\_PROPERTY\_MEMDEF ;H\_XAXISMINIMUM\_PROPERTY\_MEMDEF ;H\_XAXISMAXIMUM\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: StrikeThrough property

{button ,AL(^H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether text displays with a strike-through line.

### Data type

Boolean

### Syntax

*object.Font.StrikeThrough = trueorfalse*

*trueorfalse = object.FontStrikeThrough*

### Legal values

True (1) or False (0)

## Lotus Chart: String property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

### Data type

String

### Syntax

*object.Format.String* = *value*

*value* = *object.Format.String*

---

{button ,AL('H\_NUMBERFORMAT\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Style property

{button ,AL('H\_BORDER\_CLASS ',0)} [See list of classes](#)

(Read-write) The line style of the border around an object.

### Data type

Integer (Enumerated)

### Syntax

*object*.Edge.Style = *style*

*style* = *object*.Edge.Style

### Legal values

Value	Value
\$Solid	\$DashShortShortShort
\$Dash	\$LongLongDashShort
\$Dot	\$SpacedLongDash
\$LongDash	\$SpacedDashShort
\$DashDotDot	\$DashDotDotDash
\$DashDot	\$ExtraLongDash
\$DotDot	\$ShortLongDash
\$DashDashDash	\$DashDashSpace
\$DashDotDotDot	\$DotShortDashDashLong
\$SpacedDash	\$DoubleLine
\$DashDash	\$SpacedDot
\$DashDashShortShort	\$Invisible
\$LongDashDash	\$DefaultLineStyle
\$DashShortDash	

## Lotus Chart: SubtitleIsManual property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether the axis subtitle is specified manually or derived from the data.

### Data type

Integer (Boolean)

### Syntax

*chart.axis.SubtitleIsManual* = *trueorfalse*

*trueorfalse* = *chart.axis.SubtitleIsManual*

### Legal values

True (1) or False (0)

---

{button ,AL('H\_EXPONENT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: SubtitlePlacement property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_SUBTITLEPLACEMENT\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The placement of axis subtitles.

### Data type

Integer (Enumerated)

### Syntax

*chart.axis.SubtitlePlacement* = *placement*

*placement* = *chart.axis.SubtitlePlacement*

### Legal values

<u>Values</u>	<u>Description</u>
\$SubtitleBeneath	Place subtitle beneath the title
\$SubtitleAdjacent	Place subtitle next to the title

## **Lotus Chart: SubtitlesAreManual property**

{button ,AL(^H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) Whether the subtitles for a pie chart or doughnut chart are specified manually or are derived from the data.

### **Data type**

Integer (Boolean)

### **Syntax**

*chart.Pies.SubtitlesAreManual = trueorfalse*

*trueorfalse = chart.Pies.SubtitlesAreManual*

### **Legal values**

True (1) or False (0)



## Lotus Chart: Subtitles property

{button ,AL(^H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The subtitles showing the totals for a pie chart.

### Data type

ChartPieTitles (the collection of all subtitles)

ChartPieTitle (a single subtitle)

### Syntax

*chart.Pies.Subtitles.attribute = value*

or

*chart.Pies.Subtitles(num).attribute = value*

*value = chart.Pies.Subtitles.attribute*

or

*value = chart.Pies.Subtitles(num).attribute*

---

{button ,AL(^H\_CHARTPIETITLES\_CLASS ;H\_CHARTPIETITLE\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Subtitle property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_TITLE\_SUBTITLE\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies the attributes of the axis subtitle.

### Data type

ChartAxisTitle

### Syntax

*chart.axis.Subtitle.attribute = value*

*value = chart.axis.Subtitle.attribute*

---

{button ,AL('H\_CHARTTEXTENTRY\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Table property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) A data table that displays under a chart.

### Data type

ChartTable

### Syntax

*chart.Table.attribute = attribute*

*attribute = chart.Table.attribute*

---

{button ,AL('H\_NUMBERGRID\_PROPERTY\_MEMDEF ;H\_CHARTTABLE\_CLASS',0)} [See related topics](#)

## Lotus Chart: TextLabelPlacement property

{button ,AL(^H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) The placement of text labels for a slice in a pie chart or doughnut chart.

### Data type

Integer (Enumerated)

### Syntax

*chart.Pies.TextLabelPlacement* = *placementvalue*

*placementvalue* = *chart.Pies.TextLabelPlacement*

### Legal values

<u>Value</u>	<u>Description</u>
\$LabelsOutside	Places labels outside slices.
\$LabelsInside	Places labels inside slices, where they will fit.

## Lotus Chart: TextLabelsLink property

{button ,AL(^H\_CHARTBASE\_CLASS',0)} [See list of classes](#)

(Read-write) The link source of a chart label.

### Data type

String

### Syntax

*chart.Series(num).TextLabelsLink.Source = value*

*value = chart.Series(num).TextLabelsLink.Source*

## Lotus Chart: TextLabels property

{button ,AL('H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) The text labels associated with a pie chart.

### Data type

ChartDataLabels

### Syntax

*chart.Pies.TextLabels.attribute = attribute*

*attribute = chart.Pies.TextLabels.attribute*

---

{button ,AL('H\_CHARTDATALABELS\_CLASS',0)} [See related topics](#)

### **Lotus Chart: Text property**

{button ,AL('H\_CHARTPIETITLES\_CLASS ;H\_CHARTPIETITLE\_CLASS ;H\_CHARTTEXTENTRY\_CLASS ;H\_CHARTTEXTLABEL\_CLASS ',0)} [See list of classes](#)

(Read-write) The string in a line of text.

### **Data type**

String

### **Syntax**

*object.Text = textvalue*

*textvalue = object.Text*

## Lotus Chart: ThreeDElevation property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read and write) The amount of elevation used to display a 3D chart. Available only for bar charts, area charts, line charts, and mixed charts.

### Data type

Integer

### Syntax

*chart.ThreeDElevation = elevationvalue*

*elevationvalue = chart.ThreeDElevation*

### Legal values

5..85 degrees

### Usage

The ThreeDElevation is the angle between the viewer's eye and the x-y plane on which the chart is drawn.



## Lotus Chart: ThreeDLateralRotation property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The clockwise rotation about the chart's y-axis. Available only for 3D bar charts, area charts, line charts, and mixed charts.

### Data type

Integer

### Syntax

*chart*.ThreeDLateralRotation = *rotation*

*rotation* = *chart*.ThreeDLateralRotation

### Legal values

5..85 degrees

## Lotus Chart: ThreeDLightSource property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The direction of the light source in a 3D chart or with-depth chart. Available only for 3D or with-depth bar charts, area charts, line charts, and mixed charts.

### Data type

Integer (Enumerated)

### Syntax

*chart*.ThreeDLightSource = *direction*

*direction* = *chart*.ThreeDLightSource

### Legal values

<u>Value</u>	<u>Description</u>
\$LightLeft	Places the light source to the left of the chart.
\$LightRight	Places the light source to the right of the chart.

## Lotus Chart: ThreeDPlatform property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The size of the platform in a 3D chart. Available only for 3D bar charts, area charts, line charts, and mixed charts.

### Data type

Integer (Enumerated)

### Syntax

*chart*.ThreeDPlatform = *size*

*size* = *chart*.ThreeDPlatform

### Legal values

<u>Value</u>	<u>Description</u>
\$NoFloor	Do not display a platform floor
\$SmallFloor	Display a small platform floor
\$MediumFloor	Display a medium-sized platform floor
\$LargeFloor	Display a large platform floor

## Lotus Chart: ThreeDRowGapPercent property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The space between rows in a 3D line chart, bar chart, area chart, or mixed chart. You specify the gap between rows (series) as a percent of the width of markers in one row.

### Data type

Integer

### Syntax

*chart.ThreeDRowGapPercent* = *percentage*

*percentage* = *chart.ThreeDRowGapPercent*

### Legal values

0..100

## Lotus Chart: TickLabelCharWidth property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The maximum number of characters of a single tick label. Used only if TickLabelWidthIsManual is True.

### Data type

Integer

### Syntax

*chart.axis.TickLabelCharWidth* = *charwidth*

*charwidth* = *chart.axis.TickLabelCharWidth*

---

{button ,AL('H\_TICKLABELWIDTHISMANUAL\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: TickLabelPlacement property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The placement of the tick labels.

### Data type

Integer (Enumerated)

### Syntax

*chart.axis.TickLabelPlacement* = *placementvalue*

*placementvalue* = *chart.axis.TickLabelPlacement*

### Legal values

<u>Value</u>	<u>Description</u>
\$Automatic	Use automatic layout
\$Staggered	Stagger tick labels so they fit, if necessary
\$Rotated	Rotate tick labels so they fit, if necessary
\$Skip	Skip a specified number of tick labels
\$Slant	Slant tick labels so they fit, if necessary
\$Scaled	Scale tick labels so they all fit

### Usage

To specify the number of tick labels to skip when using the Skip enumeration value, use the TickLabelSkip property.

---

{button ,AL('H\_TICKLABELSKIP\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: TickLabelSkip property

{button ,AL('H\_CHARTAXIS\_CLASS ;H\_CHARTZAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) The number of tick marks to skip, when display tick mark labels.

### Data type

Integer

### Syntax

*chart.axis.TickLabelSkip* = *skipnumber*

*skipnumber* = *chart.axis.TickLabelSkip*

---

{button ,AL('H\_TICKLABELPLACEMENT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: TickLabelsLink property

{button ,AL('H\_CHARTTEXTLABELS\_CLASS',0)} [See list of classes](#)

(Read-write) The link source of a tick label.

### Data type

ChartDataSource

### Syntax

*mychart.axis.TickLabelsLink.attribute = value*

*value = mychart.axis.TickLabelsLink.attribute*

---

{button ,AL('H\_CHARTDATASOURCE\_CLASS ;H\_SOURCE\_PROPERTY\_MEMDEF ;H\_DATALINK\_PROPERTY\_MEMDEF',0)} [See related topics](#)



## Lotus Chart: TickLabels property

{button ,AL('H\_CHARTZAXIS\_CLASS ;H\_CHARTAXIS\_CLASS',0)} [See list of classes](#)

(Read-write) The labels for the axis tick marks.

### Data type

ChartTextLabels

### Syntax

*chart.axis.TickLabels.attribute = attribute*

*attribute = chart.axis.TickLabels.attribute*

---

{button ,AL('H\_CHARTTEXTLABELS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: TickLabelWidthsManual property

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether the width of the tick mark labels is set manually. When True, then the TickLabelCharWidth property applies.

### Data type

Integer (Boolean)

### Syntax

*chart.axis.TickLabelWidthsManual = trueorfalse*

*trueorfalse = chart.axis.TickLabelWidthsManual*

### Legal values

True (1) or False (0)

---

{button ,AL('H\_TICKLABELCHARWIDTH\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: TitlePlacement property

{button ,AL('H\_CHARTAXIS\_CLASS',0)} [See list of classes](#)

(Read-write) Controls the orientation of the chart title.

### Data type

Integer (Enumerated)

### Syntax

*chart.axis.TitlePlacement = placementvalue*

*placementvalue = chart.axis.TitlePlacement*

### Legal values

<u>Value</u>	<u>Description</u>
\$TitleParallel	Displays the title parallel to the axis.
\$TitlePerpendicular	Displays the title perpendicular to the axis.

## Lotus Chart: Titles property

{button ,AL('H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The titles for pies in a pie chart.

### Data type

ChartPieTitles (the collection of all pie titles in the pie chart)

ChartPieTitle (a single pie title)

### Syntax

*chart.Pies.Titles.attribute = attribute*

or

*chart.Pies.Titles(num).attribute = attribute*

*attribute = chart.Pies.Titles.attribute*

or

*attribute = chart.Pies.Titles(num).attribute*

---

{button ,AL('H\_CHARTPIETITLES\_CLASS ;H\_CHARTPIETITLE\_CLASS ',0)} [See related topics](#)

### **Lotus Chart: Title property (ChartAxis class)**

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_TITLE\_SUBTITLE\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies the attributes of the title of a chart axis.

### **Data type**

ChartTextEntry

### **Syntax**

*chart.axis.Title.attribute = value*

*value = chart.axis.Title.attribute*

---

{button ,AL('H\_CHARTTEXTENTRY\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Title property (ChartBase class)

{button ,AL(`H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

{button ,AL(`H\_EXAMPLE\_TITLE\_CHART\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies the attributes of the chart title.

### Data type

ChartTitleBox

### Syntax

*value = chart.Title.attribute*

*chart.Title.attribute = value*

---

{button ,AL(`H\_CHARTTITLEBOX\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Type property (ChartRegression class)

{button ,AL('H\_CHARTREGRESSION\_CLASS ',0)} [See list of classes](#)

(Read-write) The type of regression (trend) line for the series.

### Data type

Integer (Enumerated)

### Syntax

*chart*.Series(*num*).Regression.Type = *type*

*type* = *chart*.Series(*num*).Regression.Type

### Legal values

<u>Value</u>	<u>Description</u>
\$NoRegression	No regression line
\$LinearRegression	Linear regression line
\$ExponentialRegression	Exponential regression line
\$LogRegression	Logarithmic regression line
\$PowerRegression	Power regression line

## Lotus Chart: Type property (ChartBase and ChartSeries classes)

{button ,AL('H\_CHARTBASE\_CLASS ;H\_CHARTSERIES\_CLASS ',0)} [See list of classes](#)

{button ,AL('H\_EXAMPLE\_TYPE\_CHART\_PROPERTY\_EXSCRIPT',1)} [See example](#)

(Read-write) The type of the chart or series.

### Data type

Integer (Enumerated)

### Syntax

*chart.Type* = *type*

or

*chart.Series(num).Type* = *type*

*type* = *chart.Type*

or

*type* = *chart.Series(num).Type*

### Legal values

<u>Value</u>	<u>Description</u>
\$Line	Line chart
\$Bar	Bar chart
\$Area	Area chart
\$Scatter	Scatter (XY) chart (not available as a series type)
\$HLCO	HLCO chart (not available as a series type)
\$Mixed	Mixed chart (not available as a series type)
\$Pie	Pie chart (single) (not available as a series type)
\$MultiplePie	Pie chart (multiple) (not available as a series type)
\$Doughnut	Doughnut chart (single) (not available as a series type)
\$MultipleDoughnut	Doughnut chart (multiple) (not available as a series type)
\$Radar	Radar chart (not available as a series type)
\$NumberGrid	Number grid (not available as a series type)

The default chart and series type is a bar chart.



## Lotus Chart: Type property (NumberFormat class)

{button ,AL(^H\_NUMBERFORMAT\_CLASS ',0)} [See list of classes](#)

(Read-write) The type of formatting to use on a number.

### Data type

Integer (Enumerated)

### Syntax

*object.Type* = *formattype*

*formattype* = *object.Type*

### Legal values

<u>Value</u>	<u>Value</u>
\$General	\$Scientific
\$Number	\$Fixed
\$Custom	\$Comma
\$Date	\$Percent
\$String	\$Currency
\$Time	\$ISO Currency

## **Lotus Chart: Underline property**

{button ,AL(^H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether text displays with an underline.

### **Data type**

Boolean

### **Syntax**

*object.Font.Underline = trueorfalse*

*trueorfalse = object.Font.Underline*

### **Legal values**

True (1) or False (0)

## Lotus Chart: UpdateLinks method

{button ,AL(`H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

Updates all link values in the chart.

### Syntax

*chart*.UpdateLinks

### Returns

Void

### Usage

Use this method when the LinkUpdateMethod property is set to \$LinkManual.

---

{button ,AL(`H\_LINKUPDATEMETHOD\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: Use2YAxis property

{button ,AL(^H\_CHARTSERIES\_CLASS',0)} [See list of classes](#)

(Read-write) Whether to display the series on the 2nd y-axis.

### Data type

Integer (Boolean)

### Syntax

*chart.Series(num).Use2YAxis = trueorfalse*

*trueorfalse = chart.Series(num).Use2YAxis*

### Legal values

True (1) or False (0)

## Lotus Chart: UseCandlestickForHLCO property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to create a candlestick HLCO chart.

### Data type

Boolean

### Syntax

*chart.UseCandlestickForHLCO = trueorfalse*

*trueorfalse = chart.UseCandlestickForHLCO*

### Legal values

True (1) or False (0)

## Lotus Chart: UseSeparateLabelRanges property

{button ,AL(^H\_CHARTPIES\_CLASS',0)} [See list of classes](#)

(Read-write) Whether to match each pie or doughnut with a separate range of data labels. For multiple pies or multiple doughnut charts only.

### Data type

Integer (Boolean)

### Syntax

*chart.Pies.UseSeparateLabelRanges = trueorfalse*

*trueorfalse = chart.Pies.UseSeparateLabelRanges*

### Legal values

True (1) or False (0)

## Lotus Chart: UseSmartLookColors property

{button ,AL(^H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Whether to use colors specified in SmartMasters in the chart.

### Data type

Boolean

### Syntax

*chart*.UseSmartLookColors = *trueorfalse*

*trueorfalse* = *chart*.UseSmartLookColors

### Legal values

True (1) or False (0)

## Lotus Chart: ValueLabelFormat property

{button ,AL('H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The format to use for value labels of pie chart slices.

### Data type

NumberFormat

### Syntax

*chart.Pies.ValueLabelFormat.attribute = value*

*value = chart.Pies.ValueLabelFormat.attribute*

---

{button ,AL('H\_NUMBERFORMAT\_CLASS ',0)} [See related topics](#)



## Lotus Chart: ValueLabelPlacement property

{button ,AL('H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The placement of slice value labels in pie and doughnut charts.

### Data type

Integer (Enumerated)

### Syntax

*chart.Pies.ValueLabelPlacement = placement*

*placement = chart.Pies.ValueLabelPlacement*

### Legal values

<u>Value</u>	<u>Description</u>
\$LabelsOutside	Put labels outside of slices
\$LabelsInside	Put labels inside slices, where possible

---

{button ,AL('H\_VALUELABELS\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: ValueLabels property

{button ,AL('H\_CHARTPIES\_CLASS ',0)} [See list of classes](#)

(Read-write) The value labels in a pie chart.

### Data type

ChartDataLabels

### Syntax

*chart.Pies.ValueLabels.attribute* = value

*value* = *chart.Pies.ValueLabels.attribute*

---

{button ,AL('H\_CHARTDATALABELS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: Value property

{button ,AL(^H\_CHARTDATAPOINT\_CLASS ;H\_CHARTMAJORGRIDLINE\_CLASS ',0)} [See list of classes](#)

(Read-write) The value of the specified data point or major grid line.

### Data type

Double

### Syntax

*chart*.**Series**(*num*).**Datapoints**(*num*).**Value** = *pointValue*

or

*chart*.*axis*.**MajorGridlines**.**Value** = *pointValue*

**Lotus Chart: VertAlign property**

{button ,AL(^H\_FONT\_CLASS ',0)} [See list of classes](#)

(Read-write) The vertical alignment of the text.

**Note** Provided for compatibility; to be implemented in a future release.

## Lotus Chart: Visible property

```
{button ,AL(^H_CHARTAXIS_CLASS ;H_CHARTDATALABELS_CLASS ;H_CHARTGRIDLINES_CLASS ;H_CHARTLEGEND_CLASS ;H_CHARTMAJORGRIDLINES_CLASS ;H_CHARTPIESLICE_CLASS ;H_CHARTPIETITLES_CLASS ;H_CHARTPLOT_CLASS ;H_CHARTTABLEGRIDLINES_CLASS ;H_CHARTTABLEGRIDLINE_CLASS ;H_CHARTTABLE_CLASS ;H_CHARTTEXTENTRY_CLASS ;H_CHARTTEXTLABELS_CLASS ;H_CHARTTEXTLABEL_CLASS ;H_CHARTTITLEBOX_CLASS ;H_CHARTZAXIS_CLASS ;H_CHARTSERIES_CLASS ;H_CHARTREGRESSION_CLASS ',0)} See list of classes
```

```
{button ,AL(^H_EXAMPLE_VISIBLE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Whether the object displays in the chart.

### Data type

Integer (Boolean)

### Syntax

*object.Visible* = *trueorfalse*

*trueorfalse* = *object.Visible*

### Legal values

True (1) or False (0)

## Lotus Chart: Width property (Border class)

```
{button ,AL(`H_BORDER_CLASS ;H_CHARTLEGEND_CLASS ;H_CHARTNUMBERGRID_CLASS ;H_CHARTPIES_CLASS ;H_CHARTPLOT_CLASS ;H_CHARTTITLEBOX_CLASS '0)} See list of classes
```

```
{button ,AL(`H_EXAMPLE_WIDTH_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) The width of the object's border.

### Data type

Integer (Enumerated)

### Syntax

*object*.Edge.Width = *value*

*value* = *object*.Edge.Width

### Legal values

<u>Value</u>	<u>Description</u>
\$NoBorder	Do not display a border
\$ThinBorder	Show a thin border
\$ThickBorder	Show a thick border

---

```
{button ,AL(`H_HEIGHT_PROPERTY_MEMDEF ;H_X_PROPERTY_MEMDEF ;H_Y_PROPERTY_MEMDEF '0)}  
See related topics
```

## Lotus Chart: Width property (ChartBase, ChartLegend, ChartNumberGrid, ChartPies, ChartPlot, and ChartTitleBox classes)

{button ,AL('H\_CHARTBASE\_CLASS ;H\_CHARTLEGEND\_CLASS ;H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTPIES\_CLASS ;H\_CHARTPLOT\_CLASS ;H\_CHARTTITLEBOX\_CLASS ',0)} [See list of classes](#)

(Read-write) The width of the object.

### Data type

Long

### Syntax

*object.Width* = *value*

*value* = *object.Width*

### Usage

For the ChartBase class, this property is read-only.

---

{button ,AL('H\_HEIGHT\_PROPERTY\_MEMDEF ;H\_X\_PROPERTY\_MEMDEF ;H\_Y\_PROPERTY\_MEMDEF ',0)}  
[See related topics](#)

## Lotus Chart: XAxisMaximum property

{button ,AL('H\_CHARTREGRESSION\_CLASS ',0)} [See list of classes](#)

(Read-write) The maximum value at which to display the trend (regression) line.

### Data type

Double

### Syntax

*chart*.Series(*num*).Regression.XAxisMaximum = *maximum*

*maximum* = *chart*.Series(*num*).Regression.XAxisMaximum

---

{button ,AL('H\_XAXISMINIMUM\_PROPERTY\_MEMDEF ;H\_ENDPOINT\_PROPERTY\_MEMDEF ;H\_STARTPOINT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)



## Lotus Chart: XAxisMinimum property

{button ,AL('H\_CHARTREGRESSION\_CLASS ',0)} [See list of classes](#)

(Read-write) The minimum value at which to display the trend (regression) line.

### Data type

Double

### Syntax

*chart*.Series(*num*).Regression.XAxisMinimum = *minimum*

*minimum* = *chart*.Series(*num*).Regression.XAxisMinimum

---

{button ,AL('H\_XAXISMAXIMUM\_PROPERTY\_MEMDEF ;H\_ENDPOINT\_PROPERTY\_MEMDEF ;H\_STARTPOINT\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: XAxis property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) Specifies the attributes of the x-axis in a chart.

### Data type

ChartAxis

### Syntax

*chart.XAxis.attribute = value*

*value = chart.XAxis.attribute*

---

{button ,AL('H\_CHARTAXIS\_CLASS ',0)} [See related topics](#)

## Lotus Chart: XValueLink property

{button ,AL('H\_CHARTLEGENDENTRIES\_CLASS ;H\_CHARTPIETITLE\_CLASS ;H\_CHARTTEXTENTRIES\_CLASS ;H\_CHARTTEXTENTRY\_CLASS ;H\_CHARTSERIES\_CLASS ;H\_CHARTTITILEBOX\_CLASS ;H\_CHARTLEGEND\_CLASS ',0)} [See list of classes](#)

### Data type

[ChartDataSource](#)

### Syntax

*chart.Series(num).XValueLink.attribute = value*

*value = chart.Series(num).XValueLink.attribute*

---

{button ,AL('H\_CHARTDATASOURCE\_CLASS ;H\_SOURCE\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: XValue property

{button ,AL(^H\_CHARTDATAPOINT\_CLASS',0)} [See list of classes](#)

(Read-write) Sets X values for an XY (scatter) chart using separate X values.

### Data type

Double

### Syntax

*chart*.**Series**(*num*).**Datapoints**(*num*).**XValue** = *pointValue*

*pointValue* = *chart*.**Series**(*num*).**Datapoints**(*num*).**XValue**

## Lotus Chart: X property

{button ,AL(`H\_CHARTBASE\_CLASS ;H\_CHARTLEGEND\_CLASS ;H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTPIES\_CLASS ;H\_CHARTPLOT\_CLASS ;H\_CHARTTITLEBOX\_CLASS ',0)} [See list of classes](#)

(Read-write) The x-placement of the top-left corner of the object.

### Data type

Long

### Syntax

*object.X = placementvalue*

*placementvalue = object.X*

### Usage

For the ChartBase class, this property is read-only.

---

{button ,AL(`H\_HEIGHT\_PROPERTY\_MEMDEF ;H\_WIDTH\_PROPERTY\_MEMDEF ;H\_Y\_PROPERTY\_MEMDEF ',0)} [See related topics](#)

## Lotus Chart: YAxis property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The y-axis in a chart.

### Data type

ChartAxis

### Syntax

*chart.YAxis.attribute = value*

*value = chart.YAxis.attribute*

## Lotus Chart: YIntercept property

{button ,AL(^H\_CHARTREGRESSION\_CLASS ',0)} [See list of classes](#)

(Read-only) The Y-intercept of the trend (regression) line.

### Data type

Double

### Syntax

*interceptvalue* = *chart*.**Series**(*num*).**Regression.YIntercept**

## Lotus Chart: Y property

{button ,AL(`H\_CHARTBASE\_CLASS ;H\_CHARTLEGEND\_CLASS ;H\_CHARTNUMBERGRID\_CLASS ;H\_CHARTPIES\_CLASS ;H\_CHARTPLOT\_CLASS ;H\_CHARTTITLEBOX\_CLASS ',0)} [See list of classes](#)

(Read-write) The y-placement of the top-left corner of the object.

### Data type

Long

### Syntax

*object.Y = placementvalue*

*placementvalue = object.Y*

### Usage

For the ChartBase class, this property is read-only.

---

{button ,AL(`H\_HEIGHT\_PROPERTY\_MEMDEF ;H\_X\_PROPERTY\_MEMDEF ;H\_WIDTH\_PROPERTY\_MEMDEF ',0)} [See related topics](#)



## Lotus Chart: ZAxis property

{button ,AL('H\_CHARTBASE\_CLASS ',0)} [See list of classes](#)

(Read-write) The z-axis in a 3D chart.

### Data type

ChartAxis

### Syntax

*chart.ZAxis.attribute = value*

*value = chart.ZAxis.attribute*

