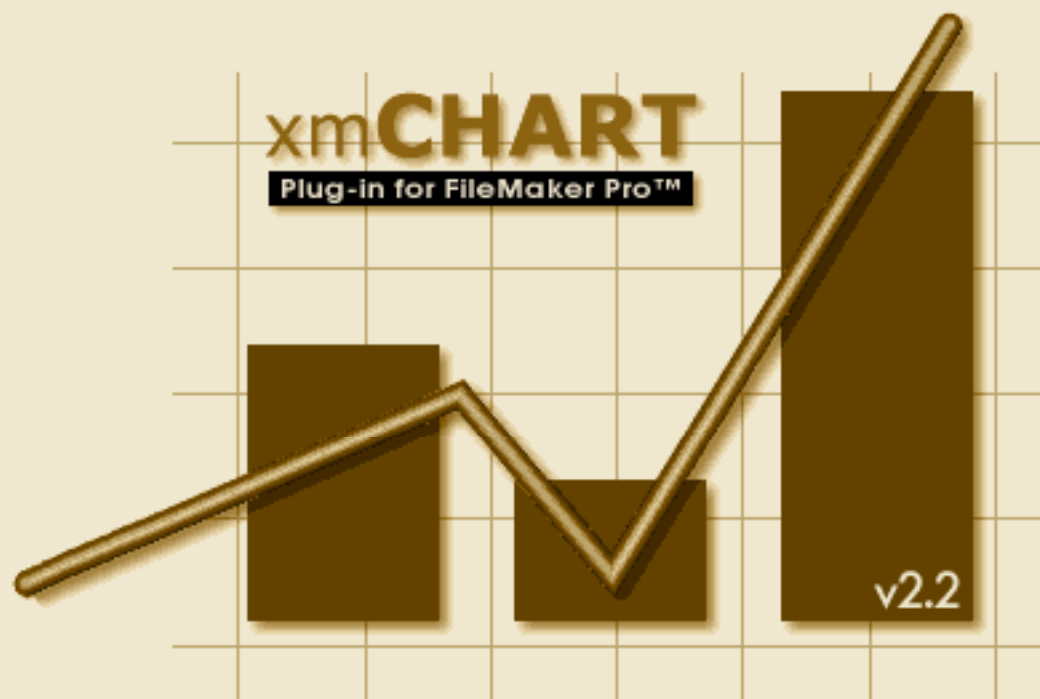


# Reference



*X2max*  
www.x2max.com

FileMaker Pro is a registered trademark of FileMaker, Inc.  
© 1997-2002 by X2max Software. All rights reserved.

# Table Of Contents

## Reference

<b>Syntax</b>	<b>7</b>
Functions	7
Arguments	8
Miscellaneous	9
 <b>Functions</b>	 <b>10</b>
AddArc()	10
AddArrow()	11
AddClipOval()	11
AddClipPolygon()	11
AddClipRect()	12
AddClipReset()	12
AddClipRoundRect()	12
AddClipSlice()	13
AddClipSmoothPolygon()	13
AddEllipse()	13
AddFrame()	14
AddLine()	14
AddOval()	14
AddPicture()	15
AddPolygon()	15
AddPolyline()	15
AddRect()	16
AddRoundFrame()	16
AddRoundRect()	17
AddSlice()	17
AddSmoothPolygon()	17
AddSmoothPolyline()	18
AddSymbol()	18

AddText()	18
AreaChart()	19
AreaChartOptions()	19
ArrowStyle()	19
AxisLabelBackground()	20
AxisLabelOptions()	20
AxisLabelStyle()	20
AxisLabelText()	21
AxisLine()	21
AxisMajorTickLabelBackground()	21
AxisMajorTickLabelOptions()	22
AxisMajorTickLabelStyle()	22
AxisMajorTickLabelTexts()	22
AxisMajorTicks()	23
AxisMinorTickLabelBackground()	23
AxisMinorTickLabelOptions()	24
AxisMinorTickLabelStyle()	24
AxisMinorTickLabelTexts()	24
AxisMinorTicks()	25
AxisOptions()	25
Background()	25
BackgroundPict()	26
BarChart()	26
BarChartOptions()	27
BorderStyle()	27
BoxPlot()	27
BoxPlotOptions()	28
BubbleChart()	28
BubbleChart2D()	28
BubbleChartOptions()	29
CandlestickChart()	29
ChartBackground()	29
ChartBackgroundPict()	30
ChartData()	30
ChartDataLowerLimits()	30
ChartDataOptions()	31
ChartDataRead()	31
ChartDataUpperLimits()	31
ChartDataWrite()	32
CloseChart()	32
CloseDrawing()	32
CloseView()	32
CurveFitting()	33
CurveFittingLineStyle()	33
CurveFittingOptions()	33

DropLineStyle()	34
DropLineReferenceLine()	34
DropLineReferencePoint()	35
DropLineReferenceSeries()	35
ErrorBarData()	35
ErrorBars()	36
ErrorBarStyle()	36
ErrorBarStyle2D()	37
FillStyle()	37
GanttChart()	37
GridFrame()	38
GridLocation()	38
HighLowChart()	38
Histogram()	39
HistogramOptions()	39
HistogramRange()	39
LabelBackground()	40
LabelOptions()	40
LabelStyle()	41
LabelTexts()	41
LegendBackground()	41
LegendOptions()	42
LegendStyle()	42
LegendTexts()	42
LineChart()	43
LineChart2D()	43
LineStyle()	43
MajorGridLineColors()	44
MajorGridLinePatterns()	44
MajorGridLineWidths()	44
MajorGridStripeColors()	45
MajorGridStripePatterns()	45
MinorGridLineColors()	45
MinorGridLinePatterns()	46
MinorGridLineWidths()	46
MinorGridStripeColors()	46
MinorGridStripePatterns()	47
MovingAverage()	47
MovingAverageLineStyle()	47
MovingAverageOptions()	48
OpenChart()	48
OpenDrawing()	49
OpenView()	49
PictureStyle()	49
PieChart()	50

PieChartAuxLines()	50
PieChartCenterLabelBackground()	50
PieChartCenterLabelStyle()	51
PieChartCenterLabelText()	51
PieChartExplodes()	51
PieChartInnerLabelBackground()	52
PieChartInnerLabelStyle()	52
PieChartInnerLabelTexts()	52
PieChartLabelOptions()	53
PolarChart()	53
PolarChartOptions()	53
RadarChart()	54
RadarChartOptions()	54
SaveAsBMPFile()	54
SaveAsEMFFile()	55
SaveAsGIFFile()	55
SaveAsJPGFile()	55
SaveAsPICTFile()	56
SaveAsPNGFile()	56
SaveAsTIFFFile()	56
Scaling()	57
ScalingOptions()	57
ScatterChart()	57
ScatterChart2D()	58
SendToClipboard()	58
SetDecimalPoint()	58
SetThousandsSep()	58
ShadowStyle()	59
SymbolStyle()	59
TitleBackground()	60
TitleOptions()	60
TitleStyle()	61
TitleSubStyle()	61
TitleText()	61
<b>Constants</b>	<b>62</b>
Appearance constants	62
Arrow head locations	63
Axis indices	63
Axis and grid locations	64
Bubble types	64
Clipping constants	64
Color constants	65
Curve fitting constants	66
Error bar directions	66

Error bar shapes	66
Error bar types	67
Explode constants	67
File flags	67
Flag constants	68
Frequency line constants	68
Grid shapes	68
Horizontal alignments	69
JPEG compression constants	69
High-Low chart constants	69
Label locations	70
Legend marker types	71
Line appearance constants	71
Location constants	72
Mathematical constants	72
Moving average calculations	72
Moving average alignments	73
Pattern constants	73
Picture adjustment constants	75
Picture source constants	75
Plane indices	77
Scaling constants	77
Scan directions	77
Series indices	78
Symbols	78
Text styles	79
Tick mark locations	79
<b>Format Specifiers</b>	<b>80</b>
General structure	80
Examples	81
<b>Error Messages</b>	<b>82</b>
<b>Index</b>	<b>84</b>

# Syntax

## Functions

- Functions are separated by carriage returns. That means there is one function call per line.
- Function names are case sensitive and start with an uppercase letter.
- The arguments of the functions are to be placed in round brackets.  
For example:

```
AddSymbol(100;150;bullet;10;1;darkGray)
```

- If a function is called several times, the last one will be used; all previous ones will be ignored.

**Exception:** The graphic primitive functions, like `AddText()`, `AddSymbol()`, `AddPicture()` etc. — they can be called multiple times.

For example:

```
OpenDrawing(400;300)
  ChartData(12 45 23 -10 34) // is ignored
  ChartData(78 -23 56 22 11)
  PieChart(oval+shadow+label) // is ignored
  BarChart(label)
  FillStyle(1;red) // is ignored
  FillStyle(1;blue)
  FillStyle(2;yellow)
  AddText(20;20;"Diagram 1")
  AddText(20;35;"(Variant A)")
  AddText(200;280;"Copyright")
CloseDrawing()
```

## Arguments

- If a function has several arguments, they are separated by semicolons ";".

- Optional arguments — those which are not absolutely necessary — can be skipped in a function call. The default values stored in xmCHART are used in this case. Examples:

```
LegendBackground(white;;2;;3)
LegendBackground( )
```

- `ChartData( )` function:

Values of a data series are separated by spaces, tabs or line feeds, several data series by semicolons ";". For example:

```
ChartData(78 -23 56 22 11; 34 23 -87 18 72)
```

The number of values per series may vary. For example:

```
ChartData(78 -12; 45 7 -23; 0; 12 -34 78 23)
```

- Texts and names of fonts, e.g. "Times", are to be placed in double quotes. For example:

```
TitleText("Chart 1")
TitleStyle("Times";24:bold+underline;darkBlue)
```

If a double quote is to be issued, it must be entered twice or as \".

For example:

```
TitleText(" "A"BC" ") produces "A"BC".
TitleText("\A\BC\" ) produces "A"BC".
```

Double quotes are not to be confused with typographical quotes ("). Typographical quotes can be activated or deactivated in the "*Edit/Preferences/Document...*" menu.

- RGB colors are defined by three integers between 0 and 255, which represent the red, green and blue components. These three color components are separated by spaces. For example:

```
FillStyle(1;255 127 0)
LegendBackground(255 255 0) // yellow
```

Since the handling of RGB colors is a bit cumbersome, xmCHART also has its own palette of 88 colors and 20 easy-to-remember names for the most common colors. Refer to the following *Constants* section for both the color palette and color names.



- 42 built-in backgrounds (gradient fills) can be accessed by entering resource as picture source and a resource ID between 1 and 42, for example:

```
BackgroundPict(resource;"12")
```

Please note: The resource ID is to be placed in double quotes.

A list of the 42 built-in gradient fills can be found in the following section *Constants*.

- A picture in the clipboard is copied to xmCHART by entering the picture source constant clipboard. For example:

```
AddPicture(100;150;;;clipboard)
```

- A picture from a file is copied to xmCHART by entering the picture source constant file and the name of the file. Examples:

```
PictureStyle(2;file;"Macintosh HD:Picts:logo.jpg")
```

```
PictureStyle(1;file;"C:\\Programs\\Pict1.gif")
```

In order to be imported, pictures have to be in PICT, GIF, JPEG, PNG, BMP or TIFF format in MacOS/X and in WMF, EMF, GIF, JPEG, PNG, BMP or TIFF format in Windows

Windows: Please note that the file path separator "\" has to be entered as "\\".

## Miscellaneous

- Comments start with 2 slashes "//". For example:

```
MajorGridLineWidths(y;x;0) // hide vert. grid lines
```

- Special characters:

\t.....tab character

\n.....newline (line feed)

\r.....carriage return (\r is equivalent to \n)

\\.....backslash

\"......quote (\\" is equivalent to ")

\xnn...ASCII Code (hexadecimal)

Examples:

```
TitleText("Chart-1\nSeries AB")
```

```
ChartDataRead("C:\\MyData\\data.txt")
```

```
LegendTexts("Category \"A\") // Category "A"
```

```
TitleText("Category \x7CA\x7C") // Category |A|
```

# Functions

The following section lists all available functions in xmCHART 2.2 in alphabetical order. The individual function arguments are listed by type, value range and default value. Examples and notes are added.

```
AddArc(left;top;width;height;startAngle;arcAngle;lineWidth;
        lineColor;linePattern)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
startAngle		num	-360..360	0	Dimension:[deg]
arcAngle		num	-360..360	90	Dimension:[deg]
lineWidth		num	0..100	1	Dimension:[pt]
lineColor		rgb	0..255	black	
linePattern		int	1..128	black	

### Examples:

```
AddArc(50;50;250;150)
```

```
AddArc(50;50;250;150;180;180;2)
```

```
AddArrow(hStart;vStart;hEnd;vEnd;lineWidth;lineColor;  
linePattern;headLocation;headLength;headWidth;  
headInset;hasHollowHead)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
hStart	•	num	-9e99...+9e99		
vStart	•	num	-9e99...+9e99		
hEnd	•	num	-9e99...+9e99		
vEnd	•	num	-9e99...+9e99		
lineWidth		num	0..100	1	Dimension:[pt]
lineColor		rgb	0..255	black	
linePattern		int	1..128	black	
headLocation		int	0..3	end	
headLength		num	0..1000	16	Dimension:[pt]
headWidth		num	0..1000	8	Dimension:[pt]
headInset		num	-1000..1000	0	Dimension:[pt]
hasHollowHead		int	0..1	off	

**Examples:**

```
AddArrow(50;50;200;200;2;blue;;;30;15;10)
```

```
AddArrow(100;150;100;0;1;darkGray;;begin+end)
```

```
AddClipOval(type;left;top;width;height)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
type		int	1..3	sect	
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		

**Examples:**

```
AddClipOval(diff;0;0;400;300)
```

```
AddClipOval(;50;50;150;150)
```

```
AddClipPolygon(type;scanDirection;listOfCoords)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
type		int	1..3	sect	
scanDirection		int	1..2	xyxy	
listOfCoords	•	numList	-9e99...+9e99		max. 1000 points

**Examples:**

```
AddClipPolygon(;xyxy;350 50 250 150 250 100 350 250 350 50)
```

```
AddClipPolygon(union;xyxy;350 50 250 150 250 100 350 250 350 50)
```

**AddClipRect( type; left; top; width; height )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
type		int	1..3	sect	
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		

**Examples:**

```
AddClipRect(sect;0;0;100;100)
```

```
AddClipRect(diff;50;50;250;150)
```

**AddClipReset()**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
-------------------	-------------	-------------	--------------	----------------	-------------

**Examples:**

```
AddClipReset()
```

**AddClipRoundRect( type; left; top; width; height; hCurvature; vCurvature )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
type		int	1..3	sect	
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
hCurvature		int	0..100	12	Dimension:[pt]
vCurvature		int	0..100	12	Dimension:[pt]

**Examples:**

```
AddClipRoundRect(diff;0;0;400;300)
```

```
AddClipRoundRect( ;50;50;150;150;16;16)
```

**AddClipSlice(type;left;top;width;height;startAngle;arcAngle)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
type		int	1..3	sect	
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
startAngle		int	-360..360	0	Dimension:[deg]
arcAngle		int	-360..360	90	Dimension:[deg]

**Examples:**

```
AddClipSlice(union;50;50;250;150)
AddClipSlice(;50;50;250;150;-45;90)
```

**AddClipSmoothPolygon(type;scanDirection;listOfCoords)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
type		int	1..3	sect	
scanDirection		int	1..2	xyxy	
listOfCoords	•	numList	-9e99...+9e99		max. 1000 points

**Examples:**

```
AddClipSmoothPolygon(;xyxy;350 50 250 150 250 100 350 250 350 50)
AddClipSmoothPolygon(;xyxy;350 50 250 150 250 100 350 250 350 50)
```

**AddEllipse(left;top;width;height;lineWidth;lineColor;linePattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
lineWidth		num	0..100	1	Dimension:[pt]
lineColor		rgb	0..255	black	
linePattern		int	1..128	black	

**Examples:**

```
AddEllipse(150;20;150;150)
AddEllipse(150;20;150;150;3;blue)
```

**AddFrame(left;top;width;height;frameWidth;frameColor;framePattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
frameWidth		num	0..100	1	Dimension:[pt]
frameColor		rgb	0..255	black	
framePattern		int	1..128	black	

**Examples:**

AddFrame(50;50;250;150)

AddFrame(50;50;250;150;3;red)

**AddLine(hStart;vStart;hEnd;vEnd;lineWidth;lineColor;linePattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
hStart	•	num	-9e99...+9e99		
vStart	•	num	-9e99...+9e99		
hEnd	•	num	-9e99...+9e99		
vEnd	•	num	-9e99...+9e99		
lineWidth		num	0..100	1	Dimension:[pt]
lineColor		rgb	0..255	black	
linePattern		int	1..128	black	

**Examples:**

AddLine(50;150;250;0)

AddLine(100;150;100;0;1;darkGray)

**AddOval(left;top;width;height;fillColor;fillPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
fillColor		rgb	0..255	black	
fillPattern		int	1..128	black	

**Examples:**

AddOval(50;20;150;150)

AddOval(50;20;150;150;red)

**AddPicture(left;top;width;height;sourceType;sourceName)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width		num	0..10000	pict width	Dimension:[pt]
height		num	0..10000	pict height	Dimension:[pt]
sourceType		int	1..3	clipboard	
sourceName		str	0..1000	" "	max. 1000 chars.

**Examples:**

```
AddPicture(100;100) // use the picture in the clipboard
AddPicture(100;100;;;file;"logo.pct")
AddPicture(100;100;;;file;"Macintosh HD:Pictures:logo.gif")
AddPicture(100;100;;;file;"C:\\Programs\\Plots\\logo.png")
```

**Notes:**

Windows OS: The file path separator "\" has to be entered as "\\".

**AddPolygon(scanDirection;listOfCoords;fillColor;fillPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
scanDirection		int	1..2	xyxy	
listOfCoords	•	numList	-9e99...+9e99		max. 1000 points
fillColor		rgb	0..255	black	
fillPattern		int	1..128	black	

**Examples:**

```
AddPolygon(xyxy;50 50 150 150 150 100 50 250)
AddPolygon(xyxy;50 50 150 150 150 100 50 250;blue)
```

**AddPolyline(scanDirection;listOfCoords;lineWidth;lineColor;linePattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
scanDirection		int	1..2	xyxy	
listOfCoords	•	numList	-9e99...+9e99		max. 1000 points
lineWidth		num	0..100	1	Dimension:[pt]
lineColor		rgb	0..255	black	
linePattern		int	1..128	black	

**Examples:**

```
AddPolyline(xyxy;50 50 150 150 150 100 50 250)
AddPolyline(xyxy;50 50 150 150 150 100 50 250;blue)
```

**AddRect(left;top;width;height;fillColor;fillPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
fillColor		rgb	0..255	black	
fillPattern		int	1..128	black	

**Examples:**

AddRect(50;20;150;150)

AddRect(53;53;250;150;gray)

**AddRoundFrame(left;top;width;height;hCurvature;vCurvature;frameWidth;frameColor;framePattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
hCurvature		int	0..100	12	Dimension:[pt]
vCurvature		int	0..100	12	Dimension:[pt]
frameWidth		num	0..100	1	Dimension:[pt]
frameColor		rgb	0..255	black	
framePattern		int	1..128	black	

**Examples:**

AddRoundFrame(50;50;250;150;;;2;red)

AddRoundFrame(50;50;250;150;16;16;2)



### AddRoundRect(left;top;width;height;hCurvature;vCurvature;fillColor;fillPattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
hCurvature		int	0..100	12	Dimension:[pt]
vCurvature		int	0..100	12	Dimension:[pt]
fillColor		rgb	0..255	black	
fillPattern		int	1..128	black	

#### Examples:

```
AddRoundRect(50;50;250;150;;;2;red)
AddRoundRect(50;50;250;150;16;16;lightGray)
```

### AddSlice(left;top;width;height;startAngle;arcAngle;fillColor;fillPattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	-9e99...+9e99		
height	•	num	-9e99...+9e99		
startAngle		num	-360..360	0	Dimension:[deg]
arcAngle		num	-360..360	90	Dimension:[deg]
fillColor		rgb	0..255	black	
fillPattern		int	1..128	black	

#### Examples:

```
AddSlice(50;50;250;150;-45;90)
AddSlice(50;50;250;150;45;90;red)
```

### AddSmoothPolygon(scanDirection;listOfCoords;fillColor;fillPattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
scanDirection		int	1..2	xyxy	
listOfCoords	•	numList	-9e99...+9e99		max. 1000 points
fillColor		rgb	0..255	black	
fillPattern		int	1..128	black	

#### Examples:

```
AddSmoothPolygon(xyxy;350 50 250 150 250 100 350 250 350 50;red)
AddSmoothPolygon(xyxy;350 50 250 150 250 100 350 250 350 50;;127)
```

### AddSmoothPolyline(scanDirection;listOfCoords;lineWidth;lineColor;linePattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
scanDirection		int	1..2	xxyy	
listOfCoords	•	numList	-9e99...+9e99		max. 1000 points
lineWidth		num	0..100	1	Dimension:[pt]
lineColor		rgb	0..255	black	
linePattern		int	1..128	black	

#### Examples:

```
AddSmoothPolyline(xxyy;350 50 250 150 250 100 350 250 350 50)
AddSmoothPolyline(xyxy;350 50 250 150 250 100 350 250 350 50;3;red)
```

### AddSymbol(hPosition;vPosition;symbolType;symbolSize;lineWidth;symbolColor;symbolPattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
hPosition	•	num	-9e99...+9e99		
vPosition	•	num	-9e99...+9e99		
symbolType		int	0..18	bullet	
symbolSize		int	0..100	9	Dimension:[pt]
lineWidth		num	0..100	1	Dimension:[pt]
symbolColor		rgb	0..255	black	
symbolPattern		int	1..128	black	

#### Examples:

```
AddSymbol(100;150;bullet;10;1;darkGray)
AddSymbol(100;150;circle)
```

### AddText(hPosition;vPosition;text;font;size;style;color;alignment)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
hPosition	•	num	-9e99...+9e99		
vPosition	•	num	-9e99...+9e99		
text		str	0..10000	" "	max. 10000 chars.
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgb	0..255	black	
alignment		int	1..3	left	

#### Examples:

```
AddText(282;295;"©2002 X2max Software";"ApplFont";10:bold;blue)
AddText(205;155;"Element-2";"Times";16:bold+underline;red)
```

**AreaChart( appearanceConstants;doShiftIntervals)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
doShiftIntervals		int	0..1	off	

**Examples:**

```
AreaChart()
AreaChart( shadow+horizontal+label;on)
```

**AreaChartOptions( useLineStyle)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
useLineStyle		int	0..1	off	

**Examples:**

```
AreaChartOptions(on)
AreaChartOptions(0)
```

**ArrowStyle( seriesIndex;lineWidth;lineColor;linePattern;  
headLocation;headLength;headWidth;headInset;  
hasHollowHead)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
lineWidth		num	0..100	1	Dimension:[pt]
lineColor		rgb	0..255	black	
linePattern		int	1..128	black	
headLocation		int	0..3	end	
headLength		num	0..1000	16	Dimension:[pt]
headWidth		num	0..1000	8	Dimension:[pt]
headInset		num	-1000..1000	0	Dimension:[pt]
hasHollowHead		int	0..1	off	

**Examples:**

```
ArrowStyle(2;1;red;;begin+end)
ArrowStyle()
```

**AxisLabelBackground( axisIndex;fillColor;fillPattern;  
borderWidth;borderColor;borderPattern;  
shadowOffset;shadowColor;shadowPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

**Examples:**

AxisLabelBackground(all;;transparent)  
AxisLabelBackground(x;yellow;;0;;2)

**AxisLabelOptions( axisIndex;location;hOffset;vOffset )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
location		int	0..9	automatic	
hOffset		num	-10000..10000	0	Dimension:[pt]
vOffset		num	-10000..10000	0	Dimension:[pt]

**Examples:**

AxisLabelOptions(x;bottomRight)  
AxisLabelOptions(y;topLeft;-5)

**AxisLabelStyle( axisIndex;font;size;style;color;alignment )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgb	0..255	black	
alignment		int	1..3	center	

**Examples:**

AxisLabelStyle(all;"Times";12)  
AxisLabelStyle(x;;;bold)

**AxisLabelText(axisIndex;text1;text2...)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
text1..1000		str	0..1000	" "	max. 1000 chars/text

**Examples:**

```
AxisLabelText(y;"Revenues")
AxisLabelText(x;"A";"B";"C") // labels for radar chart axes
```

**Notes:**

Texts consisting of several lines are possible by entering a line feed "\n".

**AxisLine(axisIndex;width;color;pattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	black	
pattern		int	1..128	black	

**Examples:**

```
AxisLine(all;0) // hide axis lines
AxisLine(x;;;gray)
```

**AxisMajorTickLabelBackground(axisIndex;fillColor;fillPattern;borderWidth;borderColor;borderPattern;shadowOffset;shadowColor;shadowPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

**Examples:**

```
AxisMajorTickLabelBackground(all;;;0) // white background, no border
AxisMajorTickLabelBackground(y;yellow) // yellow bkgrd, black border
```

**Notes:**

shadowOffset > 0: shadow bottom right  
shadowOffset < 0: shadow top left

### AxisMajorTickLabelOptions(axisIndex;location;hOffset;vOffset;labelEveryNthTickMark;startAtTickMark)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
location		int	1..3	out	
hOffset		num	-10000..10000	0	Dimension:[pt]
vOffset		num	-10000..10000	0	Dimension:[pt]
labelEveryNthTickMark			1..1000000	1	
startAtTickMark		int	1..1000000	1	

#### Examples:

```
AxisMajorTickLabelOptions(y;-3)
AxisMajorTickLabelOptions(x;;;5)
```

### AxisMajorTickLabelStyle(axisIndex;font;size;style;color;alignment)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgb	0..255	black	
alignment		int	1..3	center	

#### Examples:

```
AxisMajorTickLabelStyle(x;"Times";12)
AxisMajorTickLabelStyle(all;;;bold)
```

### AxisMajorTickLabelTexts(axisIndex;text1;text2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
text1..1000		str	0..1000	" u "	max. 1000 chars/text

#### Examples:

```
AxisMajorTickLabelTexts(x;"Q1";"Q2";"Q3";"Q4")
AxisMajorTickLabelTexts(all;"")
```

#### Notes:

Texts consisting of several lines are possible by entering a line feed "\n".

**AxisMajorTicks(axisIndex;length;width;color;pattern;location)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
length		num	0..100	5	Dimension:[pt]
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	black	
pattern		int	1..128	black	
location		int	1..3	center	

**Examples:**

```
AxisMajorTicks(all;0) // hide tick marks
AxisMajorTicks(all;3;;;out)
```

**AxisMinorTickLabelBackground(axisIndex;fillColor;fillPattern;borderWidth;borderColor;borderPattern;shadowOffset;shadowColor;shadowPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

**Examples:**

```
AxisMinorTickLabelBackground(all;lightGray)
AxisMinorTickLabelBackground(x;;;0;;;2)
```

**Notes:**

shadowOffset > 0: shadow bottom right  
shadowOffset < 0: shadow top left

**AxisMinorTickLabelOptions(axisIndex;location;hOffset;vOffset;  
labelEveryNthTickMark;startAtTickMark)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
location		int	1..3	out	
hOffset		num	-10000..10000	0	Dimension:[pt]
vOffset		num	-10000..10000	0	Dimension:[pt]
labelEveryNthTickMark			1..1000000	1	
startAtTickMark		int	1..1000000	1	

**Examples:**

```
AxisMinorTickLabelOptions(x;;;2)
AxisMinorTickLabelOptions(y;;;2)
```

**AxisMinorTickLabelStyle(axisIndex;font;size;style;color;  
alignment)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgb	0..255	black	
alignment		int	1..3	center	

**Examples:**

```
AxisMinorTickLabelStyle(all;;8)
AxisMinorTickLabelStyle(y;"Courier";9;gray)
```

**AxisMinorTickLabelTexts(axisIndex;text1;text2...)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
text1..1000		str	0..1000	" u "	max. 1000 chars/text

**Examples:**

```
AxisMinorTickLabelTexts(x;"|u|")
AxisMinorTickLabelTexts(all;"")
```

**Notes:**

Texts consisting of several lines are possible by entering a line feed "\n".



**AxisMinorTicks(axisIndex;length;width;color;pattern;location)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
length		num	0..100	3	Dimension:[pt]
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	black	
pattern		int	1..128	black	
location		int	1..3	center	

**Examples:**

```
AxisMinorTicks(all;2;;;out)
AxisMinorTicks(y;3;1;gray;;in)
```

**AxisOptions(axisIndex,axisLocation;doShiftAxis)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
axisLocation		int	0..2	back	
doShiftAxis		int	0..1	off	

**Examples:**

```
AxisOptions(x;on)
AxisOptions(all;front)
```

**Background(fillColor;fillPattern;borderWidth;borderColor;borderPattern;shadowOffset;shadowColor;shadowPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

**Examples:**

```
Background(lightYellow)
Background(;transparent;2;red)
```

### BackgroundPict (sourceType;sourceName;location;adjustment;isProportional)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
sourceType		int	1..3	clipboard	
sourceName		str	0..1000	" "	max. 1000 chars.
location		int	1..9	centerCenter	
adjustment		int	1..5	reduceOrEnlarge	
isProportional		int	0..1	off	

#### Examples:

```
BackgroundPict() // use the picture in the clipboard as background
BackgroundPict(file;"Macintosh HD:Images:BackGrd-1.png")
BackgroundPict(file;"C:\\Images\\Gradient-1.gif")
```

#### Notes:

Windows OS: The file path separator "\" has to be entered as "\\".

### BarChart (appearanceConstants;categoryGap;seriesGap;barDepth)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
categoryGap		num	0..1000	100	in % of bar width
seriesGap		num	-100..1000	see notes	in % of bar width
barDepth		num	0..1000	0	not yet implemented

#### Examples:

```
BarChart()
BarChart(stacked;25)
BarChart(shadow+proportional;0;0)
BarChart(shadow+horizontal+label;;-50)
```

#### Notes:

default series gap for non-stacked bar charts: 0 [%]

default series gap for stacked bar charts: -100 [%]

### BarChartOptions(showConnectingLines;referenceValue; makeColorSplit)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
showConnectingL.		int	0..1	off	
referenceValue		num	-9e99..+9e99	0	
makeColorSplit		int	0..1	off	

#### Examples:

```
BarChartOptions(on)
BarChartOptions(;;on)
```

#### Notes:

BarChartOptions() should be entered after the BarChart() function.

### BorderStyle(seriesIndex;appearance;width;color;pattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
appearance		int	0..4	poly	
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	black	
pattern		int	1..128	black	

#### Examples:

```
BorderStyle(1;poly;2)
BorderStyle(all;smooth;2;black)
```

### BoxPlot(appearanceConstants;upperBoxPercentile; lowerBoxPercentile;upperWhiskerPercentile; lowerWhiskerPercentile)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
upperBoxPercent.		num	0..100	75	in %
lowerBoxPercent.		num	0..100	25	in %
upperWhiskerPer.		num	0..100	90	in %
lowerWhiskerPer.		num	0..100	10	in %

#### Examples:

```
BoxPlot(horizontal)
BoxPlot(;80;20;95;5)
```

**BoxPlotOptions( itemGap;isPercentileGraph;doFillBox;showMean;  
showMedian;showOutliersOnly;showCapsOnly;  
capLength)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
itemGap		num	-100..1000	100	in % of box width
isPercentileGraph		int	0..1	off	
doFillBox		int	0..1	off	
showMean		int	0..1	off	
showMedian		int	0..1	off	
showOutliersOnly		int	0..1	off	
showCapsOnly		int	0..1	off	
capLength		num	0..1000	50	in % of box width

**Examples:**

BoxPlotOptions(;;;on;on;on)

BoxPlotOptions(50;on)

**Notes:**

BoxPlotOptions( ) should be entered after the BoxPlot( ) function.

**BubbleChart( appearanceConstants;doShiftIntervals)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
doShiftIntervals		int	0..1	off	

**Examples:**

BubbleChart( shadow)

BubbleChart( label+horizontal;on)

**BubbleChart2D( appearanceConstants)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	

**Examples:**

BubbleChart2D( )

BubbleChart2D( label+shadow)

**BubbleChartOptions(maxDiameter;bubbleType)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
maxDiameter		num	0..1000	30	Dimension:[pt]
bubbleType		int	1..2	areaProp	

**Examples:**

```
BubbleChartOptions(50)
BubbleChartOptions(30;diameterProp)
```

**Notes:**

BubbleChartOptions() should be entered after the BubbleChart() function.

**CandlestickChart(appearanceConstants;itemGap;highTickMarkLength;highTickMarkAlignment;lowTickMarkLength;lowTickMarkAlignment)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
itemGap		num	0..1000	100	in % of box width
highTickMarkLen.		num	0..100	0	in % of box width
highTickMarkAl.		int	1..3	center	
lowTickMarkLen.		num	0..100	0	in % of box width
lowTickMarkAl.		int	1..3	center	

**Examples:**

```
CandlestickChart(;5;right;5;left)
CandlestickChart(horizontal+shadow;50)
```

**ChartBackground(planeIndex;fillColor;fillPattern;borderWidth;borderColor;borderPattern;shadowOffset;shadowColor;shadowPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
planeIndex		int	0..3	all	
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

**Examples:**

```
ChartBackground(all;yellow)
ChartBackground(xy;lightGray;;0)
```

**ChartBackgroundPict(planeIndex;sourceType;sourceName)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
planeIndex		int	0..3	all	
sourceType		int	1..3	clipboard	
sourceName		str	0..1000	" "	max. 1000 chars.

**Examples:**

```
ChartBackgroundPict(xy;clipboard)
ChartBackgroundPict(all;resource;"27")
ChartBackgroundPict(;file;"BackGrdPict12.gif")
ChartBackgroundPict(xy;file;"C:\\Images\\Gradient-1.bmp")
```

**Notes:**

Windows OS: The file path separator "\" has to be entered as "\\".

**ChartData(dataSeries1;dataSeries2...)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
dataSeries		• numList	-9e99..+9e99		max. 1000 series

**Examples:**

```
ChartData(12 56 -34 67 22)
ChartData(2 3 1.32 9; -0,2 5 0 3 8; 1 4 .2)
```

**ChartDataLowerLimits(minValue1;minValue2...)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
minValue1..1000		num	-9e99..+9e99	-9e99	max. 1000 values

**Examples:**

```
ChartDataLowerLimits(0)
ChartDataLowerLimits(0;0.1)
```

**Notes:**

ChartDataLowerLimits() should be entered after the ChartData() function.

**ChartDataOptions(scanDirection)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
scanDirection		int	1..2	xyxy	

**Examples:**

```
ChartDataOptions(xyxy)
ChartData(23 45;34 67;11 76;12 56;44 21)
```

**Notes:**

ChartDataOptions() should be entered before the ChartData() function.

**ChartDataRead(fileName;doTranspose;seriesSeparator;elementSeparator)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fileName	•	str	1..1000		max. 1000 chars.
doTranspose		int	0..1	off	
seriesSeparator		char	1..1	"\n"	
elementSeparator		char	1..1	"\t"	

**Examples:**

```
ChartDataRead("Macintosh HD:Data:PlotData.dat")
ChartDataRead("C:\\Programs\\Data\\plotdata.txt";";";" " )
```

**Notes:**

Windows OS: The file path separator "\" has to be entered as "\\".

**ChartDataUpperLimits(maxValue1;maxValue2...)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
maxValue1..1000		num	-9e99..+9e99	+9e99	max. 1000 values

**Examples:**

```
ChartDataUpperLimits()
ChartDataUpperLimits(0)
```

**Notes:**

ChartDataUpperLimits() should be entered after the ChartData() function.

```
ChartDataWrite( fileName;fileFlag;creatorType;doTranspose;  
                seriesSeparator;elementSeparator;  
                formatSpecifier1;formatSpecifier2...)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fileName	•	str	1..1000		max. 1000 chars.
fileFlag		int	1..3	addCounter	
creatorType		str	4..4	"ttx"t"	MacOS/X only
doTranspose		int	0..1	off	
seriesSeparator		char	1..1	"\n"	
elementSeparator		char	1..1	"\t"	
formatSpecifier1		str	0..1000	" u "	max.1000 chars/spec.

**Examples:**

```
ChartDataWrite("Macintosh HD:Data:Plots:myData")
ChartDataWrite("ChartData";replace;;;";";" ";"|i0|";"|f2|")
ChartDataWrite("C:\\Programs\\Data\\exportdata.txt")
```

**Notes:**

Windows OS: The file path separator "\" has to be entered as "\\".

### **CloseChart( )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
-------------------	-------------	-------------	--------------	----------------	-------------

**Examples:**

```
CloseChart( )
```

### **CloseDrawing( )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
-------------------	-------------	-------------	--------------	----------------	-------------

**Examples:**

```
CloseDrawing( )
```

**Notes:**

Not necessary, only for completeness.

### **CloseView( )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
-------------------	-------------	-------------	--------------	----------------	-------------

**Examples:**

```
CloseView( )
```



**CurveFitting(seriesIndex;type)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
type		int	-4..10	linear	

**Examples:**

```
CurveFitting(all;linear)
CurveFitting(2;3)// cubic curve
```

**CurveFittingLineStyle(seriesIndex;type;width;color;pattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
type		int	-4..10	linear	
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	black	
pattern		int	1..128	black	

**Examples:**

```
CurveFittingLineStyle(1;linear;2)
CurveFittingLineStyle(all;2;;blue)
```

**CurveFittingOptions(seriesIndex;type;doSwitchAxes;  
doExtrapolate;doForceThruZero;numOfPoints)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
type		int	-4..10	linear	
doSwitchAxes		int	0..1	off	
doExtrapolate		int	0..1	off	
doForceThruZero		int	0..1	off	
numOfPoints		int	10..1000	25	

**Examples:**

```
CurveFittingOptions(all;1;;on)
CurveFittingOptions(1;2;;;;20)
```

### DropLineStyle(seriesIndex;dropLineAxisIndex;width;color;pattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
dropLineAxisIndex		int	0..3	all	
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	gray	
pattern		int	1..128	black	

#### *Examples:*

```
DropLineStyle(all;x)
DropLineStyle(1;all;1;blue;gray)
```

### DropLineReferenceLine(seriesIndex;xStart;yStart;xEnd;yEnd;width;color;pattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
xStart	•	num	-9e99..+9e99		
yStart	•	num	-9e99..+9e99		
xEnd	•	num	-9e99..+9e99		
yEnd	•	num	-9e99..+9e99		
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	black	
pattern		int	1..128	black	

#### *Examples:*

```
DropLineReferenceLine(1;0;0;10;20;;red)
DropLineReferenceLine(all;10;20;10;100;1;red;gray)
```

**DropLineReferencePoint(seriesIndex;xCenter;yCenter;symbolType;  
symbolSize;lineWidth;symbolColor;  
symbolPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
xCenter	•	num	-9e99..+9e99		
yCenter	•	num	-9e99..+9e99		
symbolType		int	0..18	bullet	
symbolSize		num	0..100	9	Dimension:[pt]
lineWidth		num	0..100	1	Dimension:[pt]
symbolColor		rgb	0..255	black	
symbolPattern		int	1..128	black	

**Examples:**

```
DropLineReferencePoint(;0;0)
DropLineReferencePoint(all;40;60;bullet;5;;red)
```

**DropLineReferenceSeries(seriesIndex;refSeriesIndex1;  
refSeriesIndex2...)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
refSeries1..100		int	0..1000	all	max. 100 series

**Examples:**

```
DropLineReferenceSeries(1;all)
DropLineReferenceSeries(4;2;3;1)
```

**ErrorBarData(seriesIndex;valueList1;valueList2;valueList3;  
valueList4)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
valueList1		numList	-9e99..+9e99	0	pos. x-error values
valueList2		numList	-9e99..+9e99	0	neg. x-error values
valueList3		numList	-9e99..+9e99	0	pos. y-error values
valueList4		numList	-9e99..+9e99	0	neg. y-error values

**Examples:**

```
ErrorBarData(1;1 1.1 1.2 1 1.1;0.5 0.6 1.0 0.9 0.8) // x-errors
ErrorBarData(all;;;1 1.1 1.2 1 1.1;0.5 0.6 1.0 0.9 0.8 // y-errors
```

**ErrorBars( seriesIndex;axisIndex;errorDirection;type;addlData1;  
addlData2;addlData3;addlData4)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
axisIndex		int	0..3	all	
errorDirection		int	0..3	both	
type		int	0..5	stdError	
addlData1		num	0..+9e99	see notes	data for pos.x-error
addlData2		num	0..+9e99	see notes	data for neg.x-error
addlData3		num	0..+9e99	see notes	data for pos.y-error
addlData4		num	0..+9e99	see notes	data for neg.y-error

**Examples:**

```
ErrorBars(all;y:both;percent;;;10;10)
ErrorBars(1;x:both;stdDev;1.5;1.5)
```

**Notes:**

if type=stdDev: addlData contains standard deviations, default: 1  
 if type=percent: addlData contains percent values, default: 5 [%]  
 if type=constant: addlData contains constant values, default: 1

**ErrorBarStyle( seriesIndex;axisIndex;showCapsOnly;capLength;  
barWidth;barColor;barPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
axisIndex		int	0..3	all	
showCapsOnly		int	0..1	off	
capLength		num	0..100	5	Dimension:[pt]
barWidth		num	0..100	1	Dimension:[pt]
barColor		rgb	0..255	black	
barPattern		int	1..128	black	

**Examples:**

```
ErrorBarStyle(all;all;;0) // no caps
ErrorBarStyle(2;y:on;2;2)
```

**ErrorBarStyle2D(seriesIndex;shapeType;fillColor;fillPattern;borderWidth;borderColor;borderPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
shapeType		int	0..2	oval	
fillColor		rgb	0..255	gray	
fillPattern		int	1..128	transparent	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	

**Examples:**

```
ErrorBarStyle2D(1;;red;gray;0)
ErrorBarStyle2D(all;oval;;;2;green)
```

**FillStyle(seriesIndex;color;pattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
color		rgb	0..255	see notes	
pattern		int	1..128	black	

**Examples:**

```
FillStyle(1;red;gray)
FillStyle(all;;transparent)
```

**Notes:**

The first 16 default colors are: 37 42 38 16 57 9 74 50 43 2 82 70 41 28 71 66. Colors are repeated periodically if the number of series is greater than the number of predefined colors. An overview of the predefined colors can be found in the *Constants* section.

**GanttChart(appearanceConstants;categoryGap;barDepth)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
categoryGap		num	0..1000	100	in % of bar width
barDepth		num	0..1000	0	not yet implemented

**Examples:**

```
GanttChart(label)
GanttChart(shadow+horizontal;50)
```

**GridFrame(planeIndex;width;color;pattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
planeIndex		int	0..3	all	
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	black	
pattern		int	1..128	black	

**Examples:**

```
GridFrame(all;2;gray)
```

```
GridFrame(xy;3)
```

**GridLocation(planeIndex;gridLocation)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
planeIndex		int	0..3	all	
gridLocation		int	0..2	back	

**Examples:**

```
GridLocation(xy;front)
```

```
GridLocation(all;none)
```

**HighLowChart(appearanceConstants;doShiftIntervals;  
highLowChartType;highTickMarkLength;  
highTickMarkAlignment;lowTickMarkLength;  
lowTickMarkAlignment;closeTickMarkLength;  
closeTickMarkAlignment;openTickMarkLength;  
openTickMarkAlignment)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
doShiftIntervals		int	0..1	off	
highLowChartType		int	1..3	highLow	
highTickMarkLen.		num	0..100	25	% of interval width
highTickMarkAl.		int	1..3	center	
lowTickMarkLen.		num	0..100	25	% of interval width
lowTickMarkAl.		int	1..3	center	
closeTickMarkLen.		num	0..100	25	% of interval width
closeTickMarkAl.		int	1..3	center	
openTickMarkLen.		num	0..100	25	% of interval width
openTickMarkAl.		int	1..3	center	

**Examples:**

```
HighLowChart(;on)
```

```
HighLowChart(horizontal;on;highLowClose)
```

**Histogram( appearanceConstants ; categoryGap ; seriesGap )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
categoryGap		num	0..1000	100	in % of bar width
seriesGap		num	-100..1000	see notes	in % of bar width

**Examples:**

```
Histogram( ;100)
Histogram(horizontal)
```

**Notes:**

default series gap for non-stacked histograms: 0 [%]

default series gap for stacked histograms: -100 [%]

**HistogramOptions( doCountData ; doMoveHigher ; doIncludeEnds ; frequencyLineOptions )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
doCountData		int	0..1	off	
doMoveHigher		int	0..1	on	
doIncludeEnds		int	0..1	on	
frequencyLineOpt.		int	0..3	none	

**Examples:**

```
HistogramOptions(on)
HistogramOptions( ; ; ; ogive)
```

**Notes:**

HistogramOptions( ) should be entered after the Histogram( ) function.

**HistogramRange( minValue ; maxValue ; numOfBins )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
minValue		num	-9e99..+9e99	autom.	
maxValue		num	-9e99..+9e99	autom.	
numOfBins		int	1..1000	10	

**Examples:**

```
HistogramRange(10;20;20)
HistogramRange(0;50)
```

**Notes:**

HistogramRange( ) should be entered after the Histogram( ) function.

**LabelBackground(seriesIndex;fillColor;fillPattern;borderWidth;  
borderColor;borderPattern;shadowOffset;  
shadowColor;shadowPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	-1..1000	all	
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

**Examples:**

LabelBackground(all;;transparent)  
LabelBackground(2;yellow)

**Notes:**

shadowOffset > 0: shadow bottom right  
shadowOffset < 0: shadow top left

**LabelOptions(seriesIndex;location;hOffset;vOffset;lowerLimit;  
upperLimit)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	-1..1000	all	
location		int	0..9	automatic	
hOffset		num	-10000..10000	0	Dimension:[pt]
vOffset		num	-10000..10000	0	Dimension:[pt]
lowerLimit		num	-9e99..+9e99	-9e99	
upperLimit		num	-9e99..+9e99	+9e99	

**Examples:**

LabelOptions(all;centerCenter)  
LabelOptions(1;;2;-2)



**LabelStyle(seriesIndex;font;size;style;color;alignment)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	-1..1000	all	
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgb	0..255	black	
alignment		int	1..3	center	

**Examples:**

```
LabelStyle(all;;;bold)
LabelStyle(2;"Courier";10;plain;gray)
```

**LabelTexts(seriesIndex;text1;text2...)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	-1..1000	all	
text1..1000		str	0..1000	" u "	max. 1000 chars/text

**Examples:**

```
LabelTexts(2;"")
LabelTexts(all;"|2f1|%" )
```

**Notes:**

Texts consisting of several lines are possible by entering a line feed "\n".

**LegendBackground(fillColor;fillPattern;borderWidth;borderColor;borderPattern;shadowOffset;shadowColor;shadowPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

**Examples:**

```
LegendBackground( )
LegendBackground(lightYellow;;2;white;;3)
```

**Notes:**

shadowOffset > 0: shadow bottom right  
shadowOffset < 0: shadow top left

```
LegendOptions( location;placeInside;hOffset;vOffset;  
numOfRows;markerType;markerWidth;  
markerHeight;markerGap;rowGap;columnGap)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
location		int	0..9	centerRight	
placeInside		int	0..1	off	
hOffset		num	-10000..10000	0	Dimension:[pt]
vOffset		num	-10000..10000	0	Dimension:[pt]
numOfRows		int	1..1000	10	
markerType		int	0..7	automatic	
markerWidth		num	0..10000	12	Dimension:[pt]
markerHeight		num	0..10000	12	Dimension:[pt]
markerGap		num	-10000..10000	6	Dimension:[pt]
rowGap		num	-10000..10000	4	Dimension:[pt]
columnGap		num	-10000..10000	8	Dimension:[pt]

**Examples:**

```
LegendOptions(bottomRight;on;;;1)  
LegendOptions(;;;;15;15)
```

```
LegendStyle( font;size;style;color;alignment )
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgb	0..255	black	
alignment		int	1..3	left	

**Examples:**

```
LegendStyle("Times")  
LegendStyle("Times";12:bold)  
LegendStyle(;;bold+underline;blue)
```

```
LegendTexts( text1;text2... )
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
text1..1000		str	0..1000	" "	max. 1000 chars/text

**Examples:**

```
LegendTexts("Turnover\n2001";"Turnover\n2002")
```

**Notes:**

Texts consisting of several lines are possible by entering a line feed "\n".

**LineChart( appearanceConstants;doShiftIntervals )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
doShiftIntervals		int	0..1	off	

**Examples:**

```
LineChart( shadow+symbol )
LineChart( shadow+horizontal+label;on)
```

**LineChart2D( appearanceConstants )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	

**Examples:**

```
LineChart2D( )
LineChart2D( label+symbol+shadow)
```

**LineStyle( seriesIndex;appearance;width;color;pattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
appearance		int	0..4	poly	
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	see notes	
pattern		int	1..128	black	

**Examples:**

```
LineStyle(1;smooth;2)
LineStyle(all;step)
LineStyle(all;;2;red)
```

**Notes:**

The first 16 default colors are: 37 42 38 16 57 9 74 50 43 2 82 70 41 28 71 66. Colors are repeated periodically if the number of series is greater than the number of predefined colors. An overview of the predefined colors can be found in the *Constants* section.

### MajorGridLineColors(directionAxis;distributionAxis;color1; color2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
color1..100		rgb	0..255	gray	max. 100 colors

#### *Examples:*

```
MajorGridLineColors(all;all;black)
```

```
MajorGridLineColors(x;y;blue)
```

### MajorGridLinePatterns(directionAxis;distributionAxis;pattern1; pattern2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
pattern1..100		int	1..128	black	max. 100 patterns

#### *Examples:*

```
MajorGridLinePatterns(all;all;gray)
```

```
MajorGridLinePatterns(y;x;gray;black)
```

### MajorGridLineWidths(directionAxis;distributionAxis;width1; width2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
width1..100		num	0..100	1	max. 100 line widths

#### *Examples:*

```
MajorGridLineWidths(x;y;1;2) // horizontal grid lines
```

```
MajorGridLineWidths(y;x;0) // hide vertical grid lines
```

### MajorGridStripeColors(directionAxis;distributionAxis;color1; color2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
color1..100		rgb	0..255	gray	max. 100 colors

#### *Examples:*

MajorGridStripeColors(x;y;lightGray;gray)

MajorGridStripeColors(all;all;lightGray)

### MajorGridStripePatterns(directionAxis;distributionAxis; pattern1;pattern2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
pattern1..100		int	1..128	black	max. 100 patterns

#### *Examples:*

MajorGridStripePatterns(all;all;gray)

MajorGridStripePatterns(x;y;black;darkGray;gray;lightGray)

### MinorGridLineColors(directionAxis;distributionAxis;color1; color2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
color1..100		rgb	0..255	gray	max. 100 colors

#### *Examples:*

MinorGridLineColors(all;all;lightGray)

MinorGridLineColors(x;y;lightGray;gray)

### MinorGridLinePatterns(directionAxis;distributionAxis;pattern1;pattern2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
pattern1..100		int	1..128	black	max. 100 patterns

#### *Examples:*

```
MinorGridLinePatterns(all;all;gray)
```

```
MinorGridLinePatterns(y;x;black;darkGray;gray;lightGray)
```

### MinorGridLineWidths(directionAxis;distributionAxis;width1;width2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
width1..100		num	0..100	1	max. 100 line widths

#### *Examples:*

```
MinorGridLineWidths(x;y;0;1)
```

```
MinorGridLineWidths(y;x;0)
```

### MinorGridStripeColors(directionAxis;distributionAxis;color1;color2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
color1..100		rgb	0..255	lightGray	max. 100 colors

#### *Examples:*

```
MinorGridStripeColors(x;y;lightGray;gray)
```

```
MinorGridStripeColors(all;all;gray)
```

### MinorGridStripePatterns(directionAxis;distributionAxis;pattern1;pattern2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
directionAxis		int	0..3	all	
distributionAxis		int	0..3	all	
pattern1..100		int	1..128	black	max. 100 patterns

#### Examples:

```
MinorGridStripePatterns(x;y;black;gray;transparent)
```

```
MinorGridStripePatterns(y;x;transparent)
```

### MovingAverage(seriesIndex;numOfIntervals;weightList)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
numOfIntervals		int	1..10000	2	
weightList		numList	-9e99..+9e99	1	

#### Examples:

```
MovingAverage(all;50)
```

```
MovingAverage(2;20;1.1 1.05 1.03 1.025 1.02 1.015 1.013 1.01)
```

### MovingAverageLineStyle(seriesIndex;numOfIntervals;appearance;width;color;pattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
numOfIntervals		int	1..10000	2	
appearance		int	0..4	poly	
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	black	
pattern		int	1..128	black	

#### Examples:

```
MovingAverageLineStyle(1;50;;2;red)
```

```
MovingAverageLineStyle(all;10;smooth;;blue)
```

```
MovingAverageOptions(seriesIndex;numOfIntervals;  
                      calculationMethod;alignment;  
                      doExtrapolate;hShift;vShift;  
                      isRelativeHShift;isRelativeVShift)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
numOfIntervals		int	1..10000	2	
calculationMethod		int	0..3	average	
alignment		int	1..4	backward	
doExtrapolate		int	0..1	off	
hShift		num	-9e99..+9e99	0	
vShift		num	-9e99..+9e99	0	
isRelativeHShift		int	0..1	off	
isRelativeVShift		int	0..1	off	

**Examples:**

```
MovingAverageOptions(all;50;;;on)
```

```
MovingAverageOptions(1;100;;centeredForward;on;;5;;on) // 5% vShift
```

```
OpenChart(left;top;width;height;isPlotArea)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-10000..10000		Dimension:[pt]
top	•	num	-10000..10000		Dimension:[pt]
width	•	num	1..10000		Dimension:[pt]
height	•	num	1..10000		Dimension:[pt]
isPlotArea		int	0..1	off	for overlay graphs

**Examples:**

```
OpenChart(0;0;400;300)
```

```
OpenChart(100;150;400;250;on)
```

**Notes:**

left and top are relative to the enclosing view.



**OpenDrawing(width; height; type)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
width	•	num	1..10000		Dimension:[pt]
height	•	num	1..10000		Dimension:[pt]
type		int	0..1	0	see notes

**Examples:**

```
OpenDrawing(300;200)
```

**Notes:**

type=0: xmCHART creates a vector graphic (Mac: PICT/Win: EMF)

type=1: xmCHART creates a bitmap graphic

**OpenView(left; top; width; height)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
left	•	num	-10000..10000		Dimension:[pt]
top	•	num	-10000..10000		Dimension:[pt]
width	•	num	1..10000		Dimension:[pt]
height	•	num	1..10000		Dimension:[pt]

**Examples:**

```
OpenView(0;0;300;200)
```

```
OpenView(200;-50;300;400)
```

**Notes:**

In case of a nested view, left and top are relative to the enclosing view.

**PictureStyle(seriesIndex; sourceType; sourceName; stackAndScaleAt)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
sourceType		int	1..3	clipboard	
sourceName		str	0..1000	" "	max. 1000 chars.
stackAndScaleAt		num	0..+9e99	0	

**Examples:**

```
PictureStyle(2;resource;"17")
```

```
PictureStyle(1;clipboard)
```

```
PictureStyle(3;file;"Macintosh HD:Fills:Gradient-1.tiff")
```

```
PictureStyle(all;file;"C:\\Programs\\Fills\\Gradient-1.png")
```

**Notes:**

Windows OS: The file path separator "\" has to be entered as "\\\".

### **PieChart(appearanceConstants;pieDepth;innerRadius;startAngle;arcAngle)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
pieDepth		num	0..1000	0	<i>not yet implemented</i>
innerRadius		num	0..100	0	in % of pie radius
startAngle		num	-360..360	0	Dimension:[deg]
arcAngle		num	-360..360	360	Dimension:[deg]

#### **Examples:**

```
PieChart(shadow+label+oval;;30)
```

### **PieChartAuxLines(horizontalLength;extensionLength;vAlignment;width;color;pattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
horizontalLength		num	0..1000	10	in % of pie radius
extensionLength		num	0..1000	0	<i>not implemented.</i>
vAlignment		int	1..3	center	
width		num	0..100	1	Dimension:[pt]
color		rgb	0..255	gray	
pattern		int	1..128	black	

#### **Examples:**

```
PieChartAuxLines(10;;bottom)
```

```
PieChartAuxLines(;;bottom;;black)
```

#### **Notes:**

PieChartAuxLines() should be entered after the PieChart() function.

### **PieChartCenterLabelBackground(fillColor;fillPattern;borderWidth;borderColor;borderPattern;shadowOffset;shadowColor;shadowPattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

#### **Examples:**

```
PieChartCenterLabelBackground(;transparent)
```

```
PieChartCenterLabelBackground(yellow;;;;;2)
```

**PieChartCenterLabelStyle(font;size;style;color;alignment)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgb	0..255	black	
alignment		int	1..3	left	

**Examples:**

```
PieChartCenterLabelStyle("Times";14)
PieChartCenterLabelStyle(;;bold)
```

**Notes:**

PieChartCenterLabelStyle() should be entered after the PieChart() function.

**PieChartCenterLabelText(text)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
text		str	0..1000	" "	max. 1000 chars.

**Examples:**

```
PieChartCenterLabelText("Distribution\nA")
PieChartCenterLabelText("|u|") // show total
```

**Notes:**

PieChartCenterLabelText() should be entered after the PieChart() function. Texts consisting of several lines are possible by entering a line feed "\n".

**PieChartExplodes(explodeOffset;sliceIndex1;sliceIndex2...)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
explodeOffset		num	0..100	20	in % of pie radius
sliceIndex1..1000		int	-3..1000	none	

**Examples:**

```
PieChartExplodes(15;all)
PieChartExplodes(20;max)
PieChartExplodes(20;2;4;6;8)
```

**Notes:**

PieChartExplodes() should be entered after the PieChart() function.

```
PieChartInnerLabelBackground( fillColor;fillPattern;  
                               borderWidth;borderColor;  
                               borderPattern;shadowOffset;  
                               shadowColor;shadowPattern)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

**Examples:**

```
PieChartInnerLabelBackground(yellow)
PieChartInnerLabelBackground(;transparent)
```

**Notes:**

PieChartInnerLabelBackground() should be entered after PieChart().

```
PieChartInnerLabelStyle( font;size;style;color;alignment)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgb	0..255	black	
alignment		int	1..3	left	

**Examples:**

```
PieChartInnerLabelStyle("Times";14)
PieChartInnerLabelStyle(;12:bold+underline)
```

**Notes:**

PieChartInnerLabelStyle() should be entered after PieChart().

```
PieChartInnerLabelTexts(text1;text2...)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
text1..1000		str	0..1000	" "	max. 1000 chars/text

**Examples:**

```
PieChartInnerLabelTexts("||f1|%") // show percent values
PieChartInnerLabelTexts("u|n(|f1|%)") // show abs. & percent values
```

**Notes:**

PieChartInnerLabelTexts() should be entered after PieChart().

**PieChartLabelOptions(useRelativeLimits;outerLabelOffset;  
innerLabelOffset)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
useRelativeLimits	int		0..1	off	
outerLabelOffset	num		-100..100	5	in % of pie radius
innerLabelOffset	num		-100..100	5	in % of pie radius

**Examples:**

```
PieChartLabelOptions(on)
PieChartLabelOptions(;10;-5)
```

**Notes:**

PieChartLabelOptions() should be entered after the PieChart() function.

**PolarChart(appearanceConstants;startAngle;arcAngle)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.	int		0..127	default	
startAngle	num		-360..360	0	Dimension:[deg]
arcAngle	num		-360..360	360	Dimension:[deg]

**Examples:**

```
PolarChart(oval)
PolarChart(symbol+label+shadow)
```

**PolarChartOptions(scalingAxisIndex;gridShape;doAddArrows;  
doNotClosePolygon;numOfAxes)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
scalingAxisIndex	int		0..1000	1	0...no scaling
gridShape	int		0..3	oval	
doAddArrows	int		0..1	off	
doNotClosePolygon	int		0..1	off	
numOfAxes	int		1..1000	12	

**Examples:**

```
PolarChartOptions(1;poly)
PolarChartOptions(;;;36)
```

**Notes:**

PolarChartOptions() should be entered after the PolarChart() function.

**RadarChart( appearanceConstants; startAngle; arcAngle )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.	int		0..127	default	
startAngle	num		-360..360	0	Dimension:[deg]
arcAngle	num		-360..360	360	Dimension:[deg]

**Examples:**

```
RadarChart( label+oval;90)
RadarChart( symbol+shadow;120;-240)
```

**RadarChartOptions( scalingAxisIndex; gridShape; doAddArrows; doNotClosePolygon )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
scalingAxisIndex	int		0..1000	1	0...no scaling
gridShape	int		0..3	oval	
doAddArrows	int		0..1	off	
doNotClosePolygon	int		0..1	off	

**Examples:**

```
RadarChartOptions(4;poly)
RadarChartOptions(0;oval;on)
```

**Notes:**

RadarChartOptions() should be entered after the RadarChart() function.

**SaveAsBMPFile( fileName; fileFlag; creatorType )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fileName	•	str	1..1000		max. 1000 chars.
fileFlag		int	1..3	addCounter	
creatorType		str	4..4	"ogle"	MacOS/X only

**Examples:**

```
SaveAsBMPFile( "Chart.bmp" )
SaveAsBMPFile( "Chart-1.bmp";replace)
SaveAsBMPFile( "Macintosh HD:Programs:Plots:Plot-1.bmp" )
SaveAsBMPFile( "C:\\Programs\\Plots\\Plot-1.bmp" )
```

**Notes:**

Windows OS: The file path separator "\" has to be entered as "\\".

**SaveAsEMFFile( fileName;fileFlag)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fileName	•	str	1..1000		max. 1000 chars.
fileFlag		int	1..3	addCounter	

**Examples:**

```
SaveAsEMFFile("Chart.emf";replace)
SaveAsEMFFile("C:\\Programs\\Plots\\Plot-1.emf")
```

**Notes:**

Windows only. The file path separator "\" has to be entered as "\\"

**SaveAsGIFFile( fileName;fileFlag)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fileName	•	str	1..1000		max. 1000 chars.
fileFlag		int	1..3	addCounter	

**Examples:**

```
SaveAsGIFFile("Chart.gif")
SaveAsGIFFile("C:\\Programs\\Plots\\Plot-1.gif")
```

**Notes:**

Windows only: The file path separator "\" has to be entered as "\\".

**SaveAsJPGFile( fileName;fileFlag;creatorType;compression)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fileName	•	str	1..1000		max. 1000 chars.
fileFlag		int	1..3	addCounter	
creatorType		str	4..4	"ogle"	MacOS/X only
compression		int	1..5	normal	

**Examples:**

```
SaveAsJPGFile("Chart.jpg")
SaveAsJPGFile("Chart-1.jpg";replace)
SaveAsJPGFile("Macintosh HD:Programs:Plots:Plot-1.jpg")
SaveAsJPGFile("C:\\Programs\\Plots\\Plot-1.jpg")
```

**Notes:**

Windows OS: The file path separator "\" has to be entered as "\\".

**SaveAsPICTFile( fileName;fileFlag;creatorType)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fileName	•	str	1..1000		max. 1000 chars.
fileFlag		int	1..3	addCounter	
creatorType		str	4..4	"ogle"	MacOS/X only

**Examples:**

```
SaveAsPICTFile("Chart")
SaveAsPICTFile("Chart.pict";replace;"8BIM")
SaveAsPICTFile("Macintosh HD:Programs:Plots:Plot-1.pct")
```

**Notes:**

MacOS/X only

**SaveAsPNGFile( fileName;fileFlag;creatorType)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fileName	•	str	1..1000		max. 1000 chars.
fileFlag		int	1..3	addCounter	
creatorType		str	4..4	"ogle"	MacOS/X only

**Examples:**

```
SaveAsPNGFile("Chart.png")
SaveAsPNGFile("Chart-1.png";replace)
SaveAsPNGFile("Macintosh HD:Programs:Plots:Plot-1.png")
SaveAsPNGFile("C:\\Programs\\Plots\\Plot-1.png")
```

**Notes:**

Windows OS: The file path separator "\" has to be entered as "\\".

**SaveAsTIFFFile( fileName;fileFlag;creatorType)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fileName	•	str	1..1000		max. 1000 chars.
fileFlag		int	1..3	addCounter	
creatorType		str	4..4	"ogle"	MacOS/X only

**Examples:**

```
SaveAsTIFFFile("Chart.tif")
SaveAsTIFFFile("Chart-1.tif";replace)
SaveAsTIFFFile("Macintosh HD:Programs:Plots:Plot-1.tiff")
SaveAsTIFFFile("C:\\Programs\\Plots\\Plot-1.tiff")
```

**Notes:**

Windows OS: The file path separator "\" has to be entered as "\\".



**Scaling(axisIndex;type;minValue;maxValue;numOfMajorIntervals;  
numOfMinorIntervals;baseValue;useEquidistantLogScaling)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
type		int	1..3	linear	
minValue		num	-9e99..+9e99	autom.	
maxValue		num	-9e99..+9e99	autom.	
numOfMajorInt.		int	1..1000	autom.	
numOfMinorInt.		int	1..1000	1	
baseValue		num	-9e99..+9e99	10	log. base
useEquidistantLogScaling			0..1	off	

**Examples:**

```
Scaling(x;linear;0;100;10)
Scaling(y;log;;;;2;on)
Scaling(y;log)
```

**ScalingOptions(axisIndex;doReverseScaling;useIntegersOnly)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..1000	all	
doReverseScaling		int	0..1	off	
useIntegersOnly		int	0..1	off	

**Examples:**

```
ScalingOptions(x;on)
ScalingOptions(all;;on)
```

**ScatterChart(appearanceConstants;doShiftIntervals)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	
doShiftIntervals		int	0..1	off	

**Examples:**

```
ScatterChart()
ScatterChart(shadow+horizontal+label;on)
```

**ScatterChart2D( appearanceConstants )**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.		int	0..127	default	

**Examples:**

```
ScatterChart2D()
ScatterChart2D(label+shadow)
```

**SendToClipboard()**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
-------------------	-------------	-------------	--------------	----------------	-------------

**Examples:**

```
SendToClipboard()
```

**Notes:**

The clipboard is the default output target.

**SetDecimalPoint(char)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
char		str	1..1	"."	

**Examples:**

```
SetDecimalPoint(",")
SetDecimalPoint(".")
```

**Notes:**

For number formatting only. Decimal numbers can be entered using a decimal point or decimal comma.

**SetThousandsSep(char)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
char		str	0..1	" "	

**Examples:**

```
SetThousandsSep(",")
SetThousandsSep(" ")
```

**Notes:**

For number formatting only. A thousands separator is not permitted for data entry.

**ShadowStyle(seriesIndex;offset;color;pattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
offset		num	-100..100	3	Dimension:[pt]
color		rgb	0..255	gray	
pattern		int	1..128	black	

**Examples:**

```
ShadowStyle(all;;lightGray)
ShadowStyle(all;1)
```

**Notes:**

offset > 0: shadow bottom right  
offset < 0: shadow top left

**SymbolStyle(seriesIndex;type;size;lineWidth;color;pattern)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..1000	all	
type		int	0..18	cross	
size		int	0..100	9	Dimension:[pt]
lineWidth		num	0..100	1	Dimension:[pt]
color		rgb	0..255	see notes	
pattern		int	1..128	black	

**Examples:**

```
SymbolStyle(1;bullet;5)
SymbolStyle(2;none)
```

**Notes:**

The first 16 default colors are: 37 42 38 16 57 9 74 50 43 2 82 70 41 28 71 66. Colors are repeated periodically if the number of series is greater than the number of predefined colors. An overview of the predefined colors can be found in the *Constants* section.

```
TitleBackground(fillColor;fillPattern;borderWidth;  
borderColor;borderPattern;shadowOffset;  
shadowColor;shadowPattern)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
fillColor		rgb	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num	0..100	1	Dimension:[pt]
borderColor		rgb	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgb	0..255	gray	
shadowPattern		int	1..128	black	

**Examples:**

```
TitleBackground()  
TitleBackground(255 255 153)  
TitleBackground(yellow;;2;white;;3)
```

**Notes:**

shadowOffset > 0: shadow bottom right  
shadowOffset < 0: shadow top left

```
TitleOptions(location;placeInside;hOffset;vOffset;  
vSubtitleOffset;titleAlignment)
```

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
location		int	0..9	topCenter	
placeInside		int	0..1	off	
hOffset		num	-10000..10000	0	Dimension:[pt]
vOffset		num	-10000..10000	0	Dimension:[pt]
vSubtitleOffset		num	-10000..10000	0	Dimension:[pt]
titleAlignment		int	1..3	center	

**Examples:**

```
TitleOptions(bottomCenter)  
TitleOptions(;on;-10)
```

**TitleStyle(font;size;style;color;alignment)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	bold	
color		rgb	0..255	black	
alignment		int	1..3	center	

**Examples:**

```
TitleStyle("Times";12:bold+underline;blue)
TitleStyle(;;plain;;left)
```

**TitleSubStyle(font;size;style;color;alignment)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgb	0..255	black	
alignment		int	1..3	center	

**Examples:**

```
TitleSubStyle("Times";10)
TitleSubStyle(;;;gray)
```

**TitleText(title;subTitle)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
title		str	0..1000	" "	max. 1000 chars.
subTitle		str	0..1000	" "	max. 1000 chars.

**Examples:**

```
TitleText("Turnover")
TitleText("Turnover";"1999-2002")
TitleText("Turnover\n1999-2002")
```

**Notes:**

Texts consisting of several lines are possible by entering a line feed "\n".

# Constants

The following section lists all constants available in xmCHART 2.2. Most constants can be accessed by using an easy-to-remember name. For example, the constant *yellow* can be used instead of the RGB value (255 255 0).

Constant names always begin with a lowercase letter and are case sensitive. Several constants such as appearance constants or text styles can be combined by a plus sign "+".

## Appearance constants

All appearance options can be accessed by name and combined. See Fig. 1 for which options are available for which types of charts.

<i>Chart function</i>	shadow	label	symbol	horizontal	stacked	proportional	oval
AreaChart()	•	•	•	•	•	•	
BarChart()	•	•	•	•	•	•	
BoxPlot()	•		•	•			
BubbleChart()	•	•	•	•			
BubbleChart2D()	•	•	•				
CandlestickChart()	•			•			
GanttChart()	•	•		•			
HighLowChart()	•	•	•	•			
Histogram()	•	•	•	•	•		
LineChart()	•	•	•	•			
LineChart2D()	•	•	•				
PieChart()	•	•					•
PolarChart()	•	•	•				•
RadarChart()	•	•	•				•
ScatterChart()	•	•		•			
ScatterChart2D()	•	•					

Fig. 1

<i>Constants</i>	<i>Value</i>
default	0
shadow	1
label	2
symbol	4
horizontal	8
stacked	16
proportional	32
oval	64

**Examples:** (the following two examples are equal)

```
BarChart(shadow+label+proportional)
BarChart(35)
```

### Arrow head locations

Arrow head location constants can be combined.

<i>Constants</i>	<i>Value</i>
none	0
begin	1
end	2

**Examples:**

```
ArrowStyle(2;1;red;;begin+end)
AddArrow(100;150;100;0;1;darkGray;;begin)
```

### Axis indices

<i>Constants</i>	<i>Value</i>
all	0
x	1
y	2
z	3

**Examples:**

```
AxisLine(all;0) // hide all axis lines
MajorGridLineWidths(y;x;0) // hide vertical grid lines
```

### Axis and grid locations

<i>Constants</i>	<i>Value</i>
none	0
back	1
front	2

**Examples:**

```
AxisOptions(all;front)
GridLocation(all;none) // hide grid
```

### Bubble types

<i>Constants</i>	<i>Value</i>
areaProp	1
diameterProp	2

**Examples:**

```
BubbleChartOptions(;diameterProp)
BubbleChartOptions(50;areaProp)
```

### Clipping constants

<i>Constants</i>	<i>Value</i>
sect	1
diff	2
union	3





















**Examples:**

```
AddClipRect(sect;0;0;100;100)
AddClipOval(diff;0;0;400;300)
```



## Color constants

In addition to defining the color by using RGB components, there is also a palette of 88 colors which are referenced by giving a value between 1 and 88. Several frequently used colors can be accessed by name. (see Fig. 2)

	<i>Constants</i>	<i>Value</i>	<i>Red</i>	<i>Green</i>	<i>Blue</i>
	white	1	255	255	255
	black	2	0	0	0
	lightGray	67	221	221	221
	gray	12	136	136	136
	darkGray	68	102	102	102
	lightRed	69	255	153	153
	red	37	255	51	0
	darkRed	26	204	51	0
	lightGreen	72	153	255	153
	green	39	0	136	0
	darkGreen	28	0	102	0
	lightBlue	86	153	204	255
	blue	42	0	0	221
	darkBlue	20	0	0	119
	lightYellow	71	255	255	153
	yellow	38	255	255	0
	darkYellow	27	255	204	0
	lightPurple	77	255	153	255
	purple	44	255	0	153
	darkPurple	33	204	0	153

































































































1													11
12													22
23													33
34													44
45													55
56													66
67													77
78													88

Fig. 2

### Examples:

```
FillStyle(1;darkPurple)
LabelBackground(2;yellow)
```

**Curve fitting constants**

<i>Constants</i>	<i>Value</i>
log	-4
exp	-3
pow	-2
none	0
linear	1

**Examples:**

```
CurveFitting(all;log)
CurveFittingLineStyle(1;linear;2)
```

**Error bar directions**

<i>Constants</i>	<i>Value</i>
none	0
plus	1
minus	2
both	3

**Examples:**

```
ErrorBars(all;y;both;percent;;;10;10)
ErrorBars(1;x;plus;stdDev;1.5;1.5)
```

**Error bar shapes**

<i>Constants</i>	<i>Value</i>
none	0
rect	1
oval	2

**Examples:**

```
ErrorBarStyle2D(all;oval;;;2;green)
ErrorBarStyle2D(1;rect;red;gray;0)
```

### Error bar types

<i>Constants</i>	<i>Value</i>
none	0
stdError	1
stdDev	2
percent	3
constant	4
valueList	5

**Examples:**

```
ErrorBars(all;y:both;percent;;;10;10)
ErrorBars(1;x;plus;stdDev;1.5;1.5)
```

### Explode constants

<i>Constants</i>	<i>Value</i>
none	0
all	-1
max	-2
min	-3

**Examples:**

```
PieChartExplodes(15;all)
PieChartExplodes(20;max)
```

### File flags

<i>Constants</i>	<i>Value</i>
addCounter	1
replace	2
throwError	3

**Examples:**

```
SaveAsBMPFile("Chart.bmp";replace)
SaveAsPICTFile("Macintosh HD:Plots:Plot-1.pct";throwError)
```

### Flag constants

<i>Constants</i>	<i>Value</i>
off	0
on	1

**Examples:**

```
LineChart(;on)
BoxPlotOptions(;;;on;on;on)
```

### Frequency line constants

<i>Constants</i>	<i>Value</i>
none	0
frequency	1
ogive	2
reverseOgive	3

**Examples:**

```
HistogramOptions(;;;ogive)
HistogramOptions(;;;frequency)
```

### Grid shapes

<i>Constants</i>	<i>Value</i>
none	0
rect	1
poly	2
oval	3

**Examples:**

```
PolarChartOptions(1;poly)
RadarChartOptions(0;oval;on)
```

### Horizontal alignments

<i>Constants</i>	<i>Value</i>
left	1
center	2
right	3

**Examples:**

```
TitleStyle("Times";12:bold+underline;blue;right)
LabelStyle(all;;;bold;;left)
```

### JPEG compression constants

<i>Constants</i>	<i>Value</i>
min	1
low	2
normal	3
high	4
max	5

**Examples:**

```
SaveAsJPGFile("Macintosh HD:Programs:Plots:Plot-1.jpg";;;low)
SaveAsJPGFile("Chart.jpg";replace;;max)
```

### High-Low chart constants

<i>Constants</i>	<i>Value</i>
highLow	1
highLowClose	2
highLowCloseOpen	3

**Examples:**

```
HighLowChart(horizontal;on;highLowClose)
HighLowChart(;;highLowCloseOpen)
```

**Label locations**

<i>Constants</i>	<i>Value</i>
automatic	0
topLeft	1
topCenter	2
topRight	3
centerLeft	4
centerCenter	5
centerRight	6
bottomLeft	7
bottomCenter	8
bottomRight	9

**Bar charts, Gantt charts & Histograms:**

smartBegin	1
smartCenter	2
smartEnd	3
begin	4
center	5
end	6
edge	7
smartOut	8
out	9

**Stacked and proportional bar and area charts:**

totalsOut	1
totalsEdge	2
runningTotalsOut	3
runningTotalsEdge	4

**Examples:**

```
LabelOptions(all:centerCenter)
LabelOptions(all:smartOut)
LabelOptions(stacked:totalsOut)
LabelOptions(-1:runningTotalsEdge)
```

## Legend marker types

Legend markers can be combined.

<i>Constants</i>	<i>Value</i>
automatic	0
rect	1
symbol	2
line	4

### *Examples:*

```
LegendOptions(bottomRight;on;;;1;rect)
LegendOptions(;off;0;0;;line+symbol)
```

## Line appearance constants

<i>Constants</i>	<i>Value</i>
none	0
jump	1
step	2
poly	3
smooth	4

### *Examples:*

```
BorderStyle(1;poly;2)
LineStyle(all;smooth)
```

**Location constants**

<i>Constants</i>	<i>Value</i>
topLeft	1
topCenter	2
topRight	3
centerLeft	4
centerCenter	5
centerRight	6
bottomLeft	7
bottomCenter	8
bottomRight	9

**Examples:**

```
TitleOptions(bottomCenter)
LegendOptions(topCenter;on)
```

**Mathematical constants**

<i>Constants</i>	<i>Value</i>
e	2.7182818284590452
pi	3.1415926535897932

**Example:**

```
Scaling(y;log;1;1000;3;10;e;on)
```

**Moving average calculations**

<i>Constants</i>	<i>Value</i>
none	0
average	1
median	2
exponential	3

**Examples:**

```
MovingAverageOptions(all;50;average)
MovingAverageOptions(all;50;exponential)
```



### Moving average alignments

<i>Constants</i>	<i>Value</i>
backward	1
forward	2
centeredBackward	3
centeredForward	4

#### *Examples:*

```
MovingAverageOptions(1;100;;centeredForward;on;;5;;on)
MovingAverageOptions(1;50;;forward)
```

### Pattern constants

There are basically two types of patterns: black&white and color patterns.

#### • **Black&White Patterns**

64 black&white patterns are available and can be referenced by entering a value between 1 and 64. Several frequently used patterns can also be accessed by name. (Fig. 3)

<i>Constants</i>	<i>Value</i>
transparent	1
black	2
darkGray	6
gray	7
lightGray	8

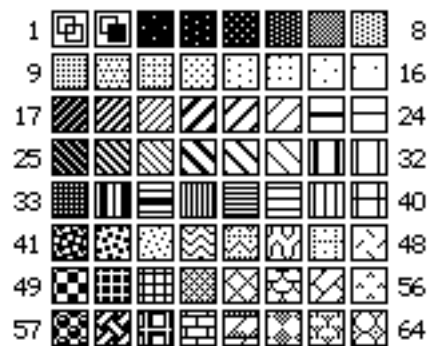


Fig. 3

#### *Examples:* (the following two examples are equal)

```
FillStyle(all;;53)
BorderStyle(1;poly;3;red;lightGray)
```

**• Color Patterns**

64 color patterns can be accessed by entering a value between 65 and 128. (Fig. 4)

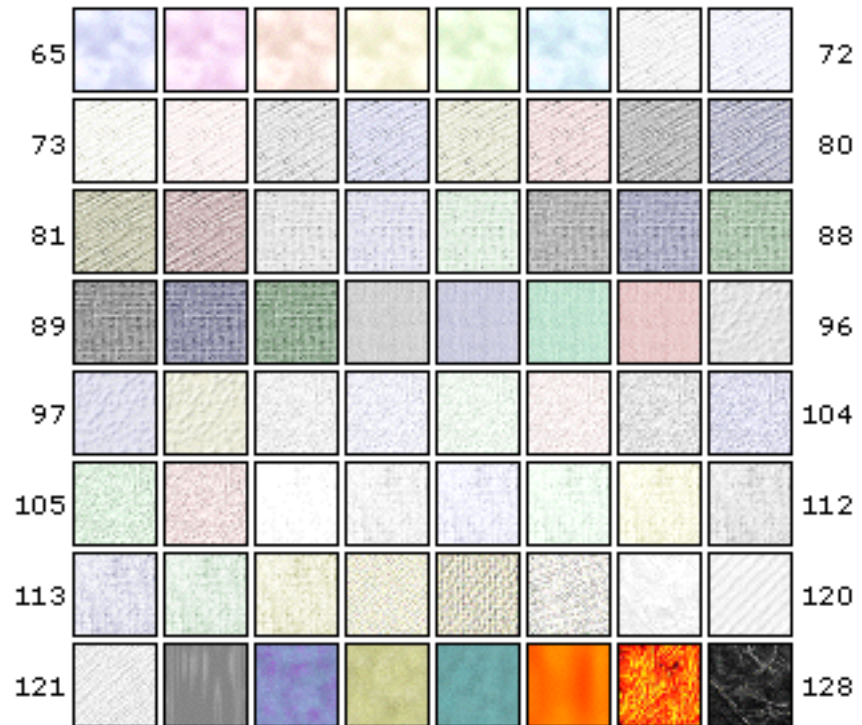


Fig. 4

***Examples:***

Background( ;119;0)

TitleBackground( ;128)

### Picture adjustment constants

Five constants are available for adjusting an optional background picture.

<i>Constants</i>	<i>Value</i>
crop	1
reduce	2
enlarge	3
reduceOrEnlarge	4
tile	5

**Examples:**

```
BackgroundPict(file;"C:\\Images\\Gradient-1.jpg";;crop)
BackgroundPict(clipboard;;tile)
```

### Picture source constants

<i>Constants</i>	<i>Value</i>
clipboard	1
resource	2
file	3

**Examples:**

```
AddPicture(10;10;;;clipboard)
AddPicture(10;10;;;file;"logo.png")
PictureStyle(1;resource;"27")
PictureStyle(3;file;"C:\\Images\\Gradient-1.jpg")
```

Presently there are 42 built-in gradient backgrounds to choose from which can be accessed by entering a resource ID between "1" and "42". (see Fig. 5). Please note, the resource ID is to be placed in double quotes, for example:

```
BackgroundPict(resource;"25")
```

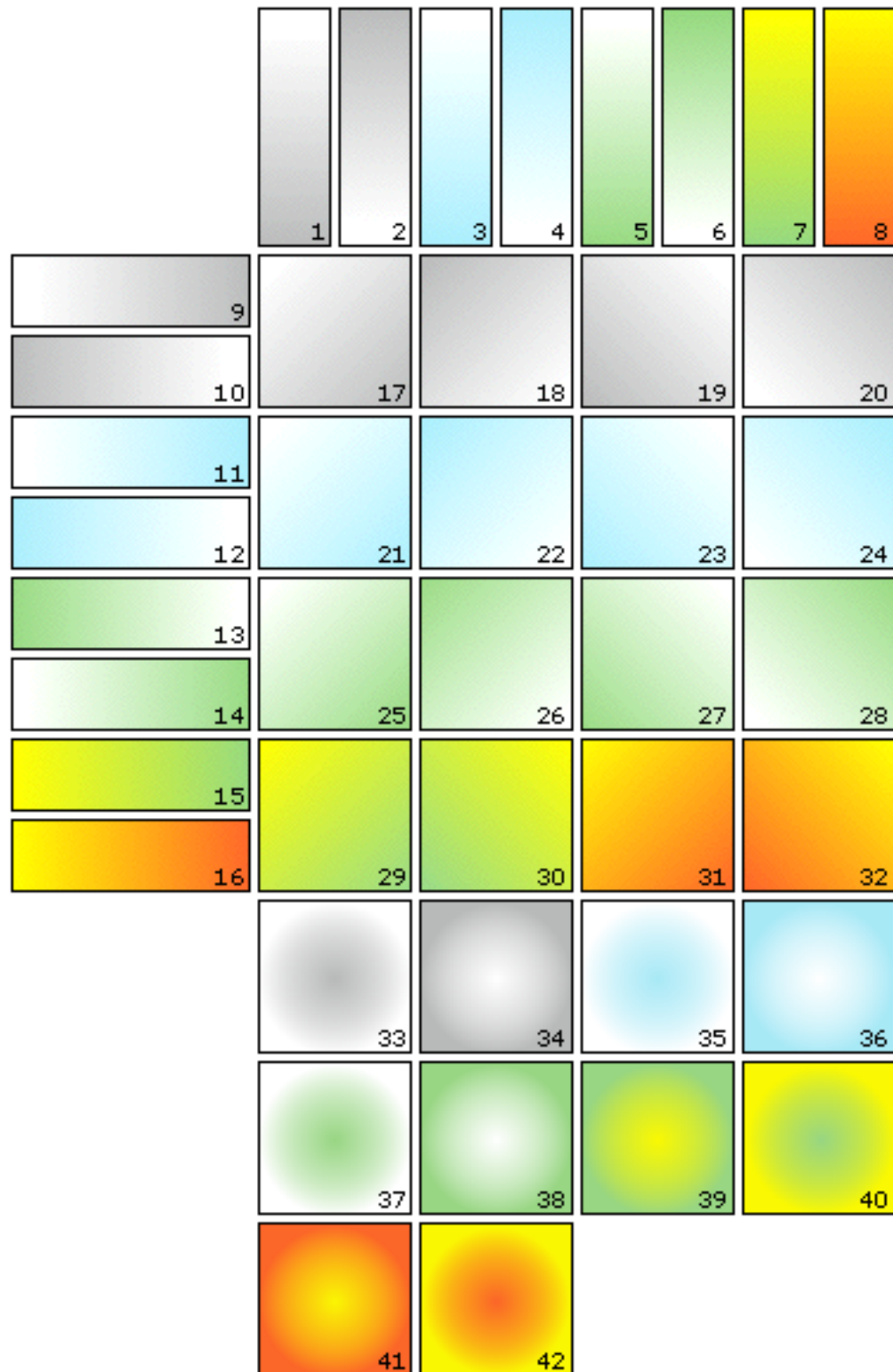


Fig. 5

### Plane indices

<i>Constants</i>	<i>Value</i>
all	0
xy	1
xz	2
yz	3

**Examples:**

```
ChartBackgroundPict(xy;clipboard)
GridFrame(all;2;gray)
```

### Scaling constants

<i>Constants</i>	<i>Value</i>
linear	1
percent	2
log	3

**Examples:**

```
Scaling(x;log;1;256;8;1;2) // logarithmic scaling
Scaling(y;percent)
```

### Scan directions

<i>Constants</i>	<i>Value</i>
xxyy	1
xyxy	2

**Examples:**

```
ChartDataOptions(xyxy)
ChartData(23 45;34 67;11 76;12 56;44 21)
```

### Series indices

<i>Constants</i>	<i>Value</i>	
stacked	-1	(available for stacked charts only)
all	0	

#### *Examples:*

```
LineStyle(all;2)
```

```
LabelOptions(stacked;totalsOut)
```

### Symbols

At the moment 18 symbols (Fig. 6) are provided by xmCHART, all of which can be accessed by name.

<i>Constants</i>	<i>Value</i>
none	0
cross	1
bullet	2
square	3
diamond	4
downTriangle	5
upTriangle	6
plus	7
circle	8
hollowSquare	9
hollowDiamond	10
hollowDownTriangle	11
hollowUpTriangle	12
hBar	13
vBar	14
leftBar	15
rightBar	16
topBar	17
bottomBar	18

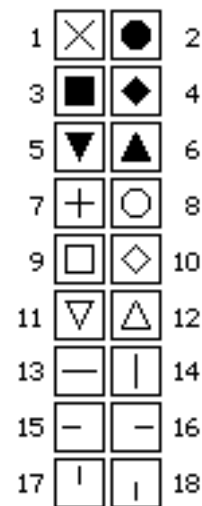


Fig. 6

#### *Examples:*

```
SymbolStyle(1;bullet;5)
```

```
SymbolStyle(2;none)
```

## Text styles

All style constants can be accessed by name and combined.

<i>Constants</i>	<i>Value</i>
plain	0
bold	1
italic	2
underline	4

**Examples:** (the following two examples are equal)

```
LegendStyle("Times";12:bold+underline)
```

```
LegendStyle("Times";12;5)
```

## Tick mark locations

<i>Constants</i>	<i>Value</i>
in	1
center	2
out	3

**Examples:**

```
AxisMajorTickLabelOptions(x;out)
```

```
AxisMinorTickLabelOptions(all;in)
```

# Format Specifiers

## General structure

"[text] | [±][power]specifier precision | [text]"

- Format specifiers are to be placed in vertical bars "|" (pipe character). Leading and trailing texts can be attached as an option.
- The decimal point character can be changed with the function `SetDecimalPoint()`.  
For example: By using the function `SetDecimalPoint(",")` the number 0.00123 is formatted as 0,00123.
- A thousands separator can be inserted with the function `SetThousandsSep()`.  
For example: By using the function `SetThousandsSep(",")` the number 1234567 is formatted as 1,234,567.
- *Specifiers:* (required)
 

u	Default format. (max. 6 digits)
i	Integer format.
f	Floating point format.
e/E	Scientific notation.
g/G	Scientific notation is used if the exponent is less than -4 or greater than or equal to the precision; otherwise f-format is used.
h/H	Engineering notation. (Engineering notation shows all exponents in multiples of three).
- *Precision:* 0 . . 9 (required, except for u-format)
- *±:* Number always displayed with a sign. (optional)
- *Power:* -9 . . 9 Multiply number by 10th power. (optional)



**Examples**

<i>number</i>	<i>format specifier</i>	<i>result</i>
1234.5678	"  u  "	1234.57
	"  f5  "	1234.56780
	"  f4  "	1234.5678
	"  f3  "	1234.568
	"  f2  "	1234.57
	"  f1  "	1234.6
	"  f0  "	1235.
	"  i0  "	1235
	"  i1  "	1230
	"  i2  "	1200
	"  i3  "	1000
	"  i4  "	0
	"  2f1  "	123456.8
	"  -2f3  "	12.346
12345.678	"  +f1  "	+1,234.6
	"  +-2f1  "	+12.3
	"  e3  "	1.235e+03
	"  +1E3  "	+1.235E+04
	"  +H1  "	+12.3E+03
	"  g4  "	0.1235e-06
	"  g6  "	0.000123
	"  2f1  %"	12.3%
	"  -6f2  Mill \$"	1.23 Mill \$
	" (  +f2  ) "	(+12.35)
0.12345		
1234567.89		
12.3456		

# Error Messages

Internally, xmCHART distinguishes between two groups of errors. On the one hand, there are so-called *parsing errors* such as typing errors, missing brackets or invalid arguments. They are located exactly by displaying the line number and, if possible, the function name and argument index so that the error can be found and corrected quickly. On the other hand, there are so-called *runtime errors*, which occur while creating the chart, for example a chart frame that is too small, or memory is running low.

The error message is stored in the return string of the external function `xmCH-DrawChart()`. For example:

```
Set Field['gError','External("xmCH-DrawChart";gFunctions)']
```

If no error occurs, the function `xmCH-DrawChart()` returns an empty string (`""`).

As the default, the error messages are displayed in English. By calling the external function `xmCH-SetLanguage()` the language of the error messages can be controlled. Currently the languages, English and German, are supported.

```
English:  External("xmCH-SetLanguage","0")
German:   External("xmCH-SetLanguage","1")
```

## **xmCHART Parsing Errors:**

```
"Syntax error."
"Invalid function name."
"Invalid number of arguments."
"Value out of range."
"String too long."
"String too short."
"Value error."
"Clipboard contains invalid data."
"Invalid constant."
```

"Not enough data."  
"Can't find resource."  
"Invalid font name."  
"Function OpenDrawing(...) missing."  
"Multiple call of function: OpenDrawing()"  
"Multiple call of function: CloseDrawing()"  
"Can't open new view."  
"Function OpenView(...) missing."  
"Nested chart definition."  
"Function OpenChart(...) missing."  
"No chart function defined."  
"Invalid axis index."

**xmCHART Runtime Errors:**

"Frame too small to draw chart."  
"Invalid scaling values."  
"Invalid number format."  
"Invalid chart data."  
"Function ChartData() not defined."  
"Invalid scaling of X-Axis."  
"Invalid scaling of Y-Axis."  
"Invalid picture dimension."  
"Curve fitting: Not enough data."  
"Division by zero."  
"Invalid logarithmic value."  
"QuickTime™ not available."  
"Image compression not available."  
"File not found."  
"File exist error."  
"File creation error."  
"File open error."  
"File read error."  
"File write error."  
"File close error."  
"File exchange error."  
"File delete error."  
"File search error."  
"File error. No such volume."  
"File name error."  
"Out of memory. Try to increase FileMaker Pro's memory size."  
"Runtime error (*errorNumber*)"

# Index

//	9	Appearance constants	62
\"	9	AreaChart	19
\\	9	AreaChartOptions	19
\n	9	areaProp	64
\r	9	Arguments	8
\t	9	Arrow head locations	63
\xnn	9	ArrowStyle	19
<b>A</b>		ASCII Code	9
AddArc	10	automatic	70,71
AddArrow	11	average	72
AddClipOval	11	Axis indices	63
AddClipPolygon	11	Axis and grid locations	64
AddClipRect	12	AxisLabelBackground	20
AddClipReset	12	AxisLabelOptions	20
AddClipRoundRect	12	AxisLabelStyle	20
AddClipSlice	13	AxisLabelText	21
AddClipSmoothPolygon	13	AxisLine	21
addCounter	67	AxisMajorTickLabelBackground	21
AddEllipse	13	AxisMajorTickLabelOptions	22
AddFrame	14	AxisMajorTickLabelStyle	22
AddLine	14	AxisMajorTickLabelTexts	22
AddOval	14	AxisMajorTicks	23
AddPicture	15	AxisMinorTickLabelBackground	23
AddPolygon	15	AxisMinorTickLabelOptions	24
AddPolyline	15	AxisMinorTickLabelStyle	24
AddRect	16	AxisMinorTickLabelTexts	24
AddRoundFrame	16	AxisMinorTicks	25
AddRoundRect	17	AxisOptions	25
AddSlice	17	<b>B</b>	
AddSmoothPolygon	17	back	64
AddSmoothPolyline	18	Background	25
AddSymbol	18	BackgroundPict	26
AddText	18	backslash	9
all	63,67,77,78	backward	73

BarChart	26	Color constants	65
BarChartOptions	27	Color Patterns	74
begin	63,70	Comments	9
black	65,73	constant	67
Black&White Patterns	73	crop	75
blue	65	cross	78
bold	79	Curve fitting constants	66
BorderStyle	27	CurveFitting	33
both	66	CurveFittingLineStyle	33
bottomBar	78	CurveFittingOptions	33
bottomCenter	70,72		
bottomLeft	70,72	<b>D</b>	
bottomRight	70,72	darkBlue	65
BoxPlot	27	darkGray	65,73
BoxPlotOptions	28	darkGreen	65
Bubble types	64	darkPurple	65
BubbleChart	28	darkRed	65
BubbleChart2D	28	darkYellow	65
BubbleChartOptions	29	default	63
built-in backgrounds	9	Default format	80
bullet	78	diameterProp	64
		diamond	78
<b>C</b>		diff	64
CandlestickChart	29	Double quotes	8
carriage return	9	downTriangle	78
center	69,70,79	DropLineReferenceLine	34
centerCenter	70,72	DropLineReferencePoint	35
centeredBackward	73	DropLineReferenceSeries	35
centeredForward	73	DropLineStyle	34
centerLeft	70,72		
centerRight	70,72	<b>E</b>	
ChartBackground	29	e	72
ChartBackgroundPict	30	edge	70
ChartData	8,30	end	63,70
ChartDataLowerLimits	30	Engineering notation	80
ChartDataOptions	31	enlarge	75
ChartDataRead	31	Error bar directions	66
ChartDataUpperLimits	31	Error bar shapes	66
ChartDataWrite	32	Error bar types	67
circle	78	ErrorBarData	36
clipboard	9,75	ErrorBars	36
Clipping constants	64	ErrorBarStyle	36
CloseChart	32	ErrorBarStyle2D	37
CloseDrawing	32	exp	66
CloseView	32	Explode constants	67

exponential	72	J	
<b>F</b>		JPEG Compression Constants	69
file	9,75	jump	71
File flags	67	<b>L</b>	
FillStyle	37	label	63
Flag constants	68	Label locations	70
Floating point format	80	LabelBackground	40
forward	73	LabelOptions	40
frequency	68	LabelStyle	41
Frequency line constants	68	LabelTexts	41
front	64	left	69
<b>G</b>		leftBar	78
GanttChart	37	Legend marker types	71
gray	65,73	LegendBackground	41
green	65	LegendOptions	42
Grid shapes	68	LegendStyle	42
GridFrame	38	LegendTexts	42
GridLocation	38	lightBlue	65
<b>H</b>		lightGray	65,73
hBar	78	lightGreen	65
hexadecimal	9	lightPurple	65
high	69	lightRed	65
High-Low chart constants	69	lightYellow	65
highLow	69	line	71
HighLowChart	38	Line appearance constants	71
highLowClose	69	linear	66,77
highLowCloseOpen	69	LineChart	43
Histogram	39	LineChart2D	43
HistogramOptions	39	LineStyle	43
HistogramRange	39	Location constants	72
hollowDiamond	78	log	66,77
hollowDownTriangle	78	low	69
hollowSquare	78	<b>M</b>	
hollowUpTriangle	78	MajorGridLineColors	44
horizontal	63	MajorGridLinePatterns	44
Horizontal alignments	69	MajorGridLineWidths	44
<b>I</b>		MajorGridStripeColors	45
in	79	MajorGridStripePatterns	45
Integer format	80	Mathematical constants	72
italic	79	max	67,69
		median	72
		min	67,69

MinorGridLineColors	45	Plane indices	77
MinorGridLinePatterns	46	plus	66,78
MinorGridLineWidths	46	PolarChart	53
MinorGridStripeColors	46	PolarChartOptions	53
MinorGridStripePatterns	47	poly	68,71
minus	66	pow	66
Moving average alignments	73	proportional	63
Moving average calculations	72	purple	65
MovingAverage	47		
MovingAverageLineStyle	47	Q	
MovingAverageOptions	48	quote	9
<b>N</b>		<b>R</b>	
newline	9	RadarChart	54
none	63,64,66-68,71,72,78	RadarChartOptions	54
normal	69	RBG components	64
<b>O</b>		rect	66,68,71
off	68	red	65
ogive	68	reduce	75
on	68	reduceOrEnlarge	75
OpenChart	48	replace	67
OpenDrawing	49	resource	9,75
OpenView	49	reverseOgive	68
out	70,79	RGB colors	8
oval	63,66,68	right	69
<b>P</b>		rightBar	78
Pattern constants	73	runningTotalsEdge	70
percent	67,77	runningTotalsOut	70
pi	72	<b>S</b>	
Picture adjustment constants	75	SaveAsBMPFile	54
Picture source constants	75	SaveAsEMFFile	55
PictureStyle	49	SaveAsGIFFile	55
PieChart	50	SaveAsJPGFile	55
PieChartAuxLines	50	SaveAsPCTFile	56
PieChartCenterLabelBackground	50	SaveAsPNGFile	56
PieChartCenterLabelStyle	51	SaveAsTIFFFile	56
PieChartCenterLabelText	51	Scaling	57
PieChartExplodes	51	Scaling constants	77
PieChartInnerLabelBackground	52	ScalingOptions	57
PieChartInnerLabelStyle	52	Scan directions	77
PieChartInnerLabelTexts	52	ScatterChart	57
PieChartLabelOptions	53	ScatterChart2D	58
plain	79	Scientific notation	80
		sect	64

SendToClipboard	58	<b>V</b>	
Series indices	78	valueList	67
SetDecimalPoint	58	vBar	78
SetThousandsSep	58		
shadow	63	<b>W</b>	
ShadowStyle	59	white	65
smartBegin	70		
smartCenter	70	<b>X</b>	
smartEnd	70	x	63
smartOut	70	xmCH-DrawChart	82
smooth	71	xmCH-SetLanguage	82
square	78	xyyy	77
stacked	63,78	xy	77
stdDev	67	xyxy	77
stdError	67	xz	77
step	71		
symbol	63,71	<b>Y</b>	
Symbols	78	y	63
SymbolStyle	59	yellow	65
		yz	77
<b>T</b>			
tab character	9	<b>Z</b>	
Text styles	79	z	63
throwError	67		
Tick mark locations	79		
tile	75		
TitleBackground	60		
TitleOptions	60		
TitleStyle	61		
TitleSubStyle	61		
TitleText	61		
topBar	78		
topCenter	70,72		
topLeft	70,72		
topRight	70,72		
totalsEdge	70		
totalsOut	70		
transparent	73		
Typographical quotes	8		
<b>U</b>			
underline	79		
union	64		
upTriangle	78		