

Reference



X2max
www.x2max.com

FileMaker Pro is a registered trademark of FileMaker Inc.
©1997-2012 by X2max Software. All rights reserved.

Table Of Contents

Reference

Syntax	7
Functions	7
Arguments	8
Miscellaneous	11
 Functions	 12
AddArc()	12
AddArrow()	13
AddClipOval()	13
AddClipPolygon()	13
AddClipRect()	14
AddClipReset()	14
AddClipRoundRect()	14
AddClipSlice()	15
AddClipSmoothPolygon()	15
AddEllipse()	15
AddFrame()	16
AddLine()	16
AddOval()	16
AddPath()	17
AddPicture()	17
AddPolygon()	18
AddPolyline()	18
AddRect()	18
AddRoundFrame()	19
AddRoundRect()	19
AddSlice()	20
AddSmoothPolygon()	20
AddSmoothPolyline()	20
AddSymbol()	21
AddText()	21
AreaChart()	21
AreaChart2D()	22
AreaChartOptions()	22
ArrowStyle()	22

AxisLabelBackground()	23
AxisLabelOptions()	23
AxisLabelStyle()	23
AxisLabelText()	24
AxisLine()	24
AxisMajorTickLabelBackground()	24
AxisMajorTickLabelOptions()	25
AxisMajorTickLabelStyle()	25
AxisMajorTickLabelTexts()	25
AxisMajorTicks()	26
AxisMinorTickLabelBackground()	26
AxisMinorTickLabelOptions()	27
AxisMinorTickLabelStyle()	27
AxisMinorTickLabelTexts()	27
AxisMinorTicks()	28
AxisOptions()	28
Background()	28
BackgroundPict()	29
BarChart()	29
BarChart2D()	30
BarChartOptions()	30
BorderStyle()	30
BoxPlot()	31
BoxPlotOptions()	31
BubbleChart()	31
BubbleChart2D()	32
BubbleChartOptions()	32
CandlestickChart()	32
CandlestickChart2D()	33
ChartBackground()	33
ChartBackgroundPict()	33
ChartData()	34
ChartDataLowerLimits()	34
ChartDataOptions()	35
ChartDataRead()	35
ChartDataUpperLimits()	35
ChartDataWrite()	36
CloseChart()	36
CloseDrawing()	36
CloseView()	36
CurveFitting()	37
CurveFittingLineStyle()	37
CurveFittingOptions()	37
DateTimeOptions()	38
DropLineReferenceLine()	38

DropLineReferencePoint()	38
DropLineReferenceSeries()	39
DropLineStyle()	39
ErrorBarData()	39
ErrorBars()	40
ErrorBarStyle()	40
ErrorBarStyle2D()	41
FillStyle()	41
GanttChart()	41
GridFrame()	42
GridLocation()	42
HighLowChart()	42
HighLowChart2D()	43
Histogram()	43
HistogramOptions()	44
HistogramRange()	44
LabelBackground()	44
LabelOptions()	45
LabelStyle()	45
LabelTexts()	46
LegendBackground()	46
LegendOptions()	46
LegendStyle()	47
LegendTexts()	47
LineChart()	48
LineChart2D()	48
LineStyle()	48
MajorGridLineColors()	49
MajorGridLinePatterns()	49
MajorGridLineWidths()	49
MajorGridStripeColors()	49
MajorGridStripePatterns()	50
MinorGridLineColors()	50
MinorGridLinePatterns()	50
MinorGridLineWidths()	51
MinorGridStripeColors()	51
MinorGridStripePatterns()	51
MovingAverage()	52
MovingAverageLineStyle()	52
MovingAverageOptions()	53
OpenChart()	53
OpenDrawing()	54
OpenView()	54
PictureStyle()	55
PieChart()	55

PieChartAuxLines()	56
PieChartCenterLabelBackground()	56
PieChartCenterLabelStyle()	56
PieChartCenterLabelText()	57
PieChartExplodeDepths()	57
PieChartExplodes()	57
PieChartInnerLabelBackground()	58
PieChartInnerLabelStyle()	58
PieChartInnerLabelTexts()	59
PieChartLabelOptions()	59
PolarChart()	59
PolarChartOptions()	60
RadarChart()	60
RadarChartOptions()	60
SaveAsBMPFile()	61
SaveAsEMFFile()	61
SaveAsGIFFile()	61
SaveAsJPGFile()	62
SaveAsPDFFile()	62
SaveAsPICTFile()	62
SaveAsPNGFile()	63
SaveAsSVGFile()	63
SaveAsTIFFFile()	63
Scaling()	64
ScalingOptions()	64
ScatterChart()	64
ScatterChart2D()	65
SendToClipboard()	65
SetDecimalPoint()	65
SetThousandsSep()	65
ShadowStyle()	66
SymbolStyle()	66
TitleBackground()	67
TitleOptions()	67
TitleStyle()	68
TitleSubStyle()	68
TitleText()	69
Constants	70
Appearance constants	70
Arrow head locations	71
Axis indices	71
Axis and grid locations	72
Bubble types	72
Clipping constants	72

Color constants	73
Curve fitting constants	77
Date orders	77
DateTime scaling constants	77
Error bar directions	78
Error bar shapes	78
Error bar types	79
Explode constants	79
File flags	79
Flag constants	80
Frequency line constants	80
Grid shapes	80
Horizontal alignments	81
JPEG compression constants	81
High-Low chart constants	81
Label locations	82
Legend marker types	83
Line shape constants	83
Location constants	84
Mathematical constants	84
Moving average calculations	84
Moving average alignments	85
Pattern constants	85
Picture adjustment constants	87
Picture source constants	87
Plane indices	89
Scaling constants	89
Scan directions	89
Series indices	90
Symbols	90
Text styles	91
Tick mark locations	91
Vertical alignments	91
Format Specifiers	92
Number Format Specifiers	92
General structure	92
Examples	93
DateTime Format Specifiers	94
General structure	94
Examples	95
Error Messages	96
Index	99

Syntax

Functions

- Functions are separated by carriage returns. That means there is one function call per line.

- Function names and constants are not case sensitive.

For example:

```
BORDERSTYLE(all;POLY;2;darkblue) // allowed
BorderStyle(All;Poly;3;DARKBLUE) // no error
```

- 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 functions `PieChartExplodes()`, `PieChartExplodeDepths()` and graphic primitive functions, e.g. `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 ";", e.g.: `OpenDrawing(400;300)`

- 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)
```

Entering chart values in scientific notation is allowed. For example:

```
ChartData(-1.2e04 0.2E04 .2e-3)
```

Support of "NULL" for missing charting values. For example:

```
ChartData(12 98.3 null 8 Null NULL 7.23 -0.67)
```

Thousands separators are not permitted. Example:

```
ChartData(1,234.56 12.345,67) // invalid!
```

- 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 "File>File Options...>Text" menu.

Texts may also have several lines. For example:

```
TitleText("Chart 1\nOverview") // "\n"...new line
```


- 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
```

Transparency (alpha value) is supported as optional 4th argument in R G B A. Range: 0 (invisible) ... 255 (opaque) default: 255

For example:

```
FillStyle(1;100 0 188 255) // same as 100 0 188
FillStyle(1;255 0 255 200)
FillStyle(1;#FF8900DE)
FillStyle(1;darkRed 200) // Error, not allowed!
FillStyle(1;darkRed,200) // Error, not allowed!
```

Colors can be entered as hexcodes, with or without alpha value, #RRGGBB, #rrggb, #RRGGBBAA, #rrggbbaa. For example:

```
FillStyle(1;#FF8900DE)
BorderStyle(all;poly;2;#ff12c0)
```

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

New in xmCHART 3.4 are linear and radial gradients. Gradients are available for all xmCHART functions (fills, borders, texts, etc.) and are defined by a list of numbers.

Linear gradients:

```
[ 1 angle isScalableFlag offset1 r1 g1 b1 a1 offset2 r2 g2 b2 a2 ... ]
```

For linear gradients the first value of the list is always 1, the second value defines the direction of the gradient within a range of -360 and +360 degrees. 0 degree defines a horizontal gradient from left to right, 90 degrees a gradient from top to bottom. The 3rd value is either 0 or 1. 1 means the angle of the gradient is linked to the width/height ratio of the bounding box, which is in most cases preferable. For example, for a diagonal gradient set the first three values to 1 45 1. On the other hand if you set the third value 0 (1 45 0 ...) the direction of the gradient is always 45 degrees regardless of the shape of the bounding box. These three values follow a series of *color stops*. A color stop is represented by five elements: offset, red, green, blue and alpha. Offsets are between 0 and 1, the 4 RGBA components between 0 and 255. Example:

```
Background(1 45 1 0.0 255 0 0 255 0.5 255 255 0
           255 1.0 0 0 255 255;;0)
```

Radial gradients:

```
[ 2 cx cy offset1 r1 g1 b1 a1 offset2 r2 g2 b2 a2 ... ]
```

For radial gradients the first value of the list is always 2. The second and third value represent the center point in relative coordinates in respect to the bounding box. For example the center of the bounding box is defined by cx=0.5 and cy=0.5. These three values follows a series of *color stops*. A color stop is represented by five elements: offset, red, green, blue and alpha. Offsets are between 0 and 1, the 4 RGBA components between 0 and 255. For example:

```
FillStyle(1; 2 0.5 0.5 0.4 0 155 155 255 0.8 0
           255 255 255 1.0 0 155 255 255)
```

- 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. Either a complete, absolute file path or only a relative path can be passed. The relative path refers to the folder in which the current FileMaker Pro database file is located. The separator in the file path is a slash "/", with no space before and after the slash.

Examples:

```
BackgroundPict(file;"Pictures/Pict_01.png")
```

```
BackgroundPict(file;"C:/Pictures/Pict_01.png")
```

```
BackgroundPict(file;"Macintosh HD/Picts/Pict_01.pdf")
```

In Mac OS X pictures have to be in PDF, PICT, GIF, JPEG, PNG, BMP, TIFF format in order to be imported. In Windows pictures have to be in WMF, EMF, GIF, JPEG, PNG, BMP, TIFF format in order to be imported.

- A dash pattern can be assigned to the line width by adding a list of dash lengths and gaps. Examples:

```
LineStyle(1;poly;1) // solid line (default)
```

```
BorderStyle(1;poly;1 2 2) // dotted line
```

```
LineStyle(all;poly;1 5 5) // dashed line
```

```
BorderStyle(all;;;2 9 4 2 4) // dash-dotted line
```

Miscellaneous

- Comments:

(1) Comments start with 2 slashes "//". For example:

```
MajorGridLineWidths(all;all;0) // hide grid lines
```

(2) C-style multi-line comments /* ... */. For example:

```
MajorGridLineWidths(all;all;0) /* hide grid lines */
```

- Special characters:

\t.....tab character

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

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

\\.....backslash

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

\uXXXX...Unicode character (hexadecimal)

Examples:

```
TitleText("Chart-1\nSeries AB")
LegendTexts("Category \"A\"") // Category "A"
LegendTexts("Category \"A\"") // Category "A"
TitleText("Group \u03b1") // \u03b1 = "α"
AddText(5;50;"\u00A9X2max Software") // \u00A9 = "©"
```

- Get xmCHART version:

```
xmCH_GetVersion(type) with type = 1: long version string
                        type = 2: short version string
```

Examples:

```
xmCH_GetVersion(1) // returns, for example: "xmCHART 3.4.6"
xmCH_GetVersion(2) // returns, for example: "3.4.6"
```

- Mouse coordinates:

xmCH_GetMouse() returns the current mouse coordinates of the cursor. In doing so, the window scroll offsets, the status area and, starting with FileMaker Pro 8, also the zoom factor will be properly taken into account. The output of the coordinate values can be controlled by a format string. The x-coordinate is represented by the placeholder "%1" and the y-coordinate by the placeholder "%2". If an empty format string is passed to xmCH_GetMouse(""), then the x and y-coordinate will be separated by a space.

Examples:

```
xmCH_GetMouse("") // returns x y, e.g.: 419 253
xmCH_GetMouse("%1 %2") // returns x y, e.g.: 419 253
xmCH_GetMouse("%2") // returns only y-coordinate
xmCH_GetMouse("(x=%1; y=%2)") // returns e.g. (x=419; y=253)
```

Functions

The following section lists all available functions in xmCHART 3.4 in alphabetical order. The individual function arguments are listed by type, value range and default value. Examples and notes are added.

Type:

num number
 num[] list of numbers
 int integer
 int[] list of integers
 str text string
 rgba RGB-color (with optional alpha component)

**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	0..+9e99		
height	•	num	0..+9e99		
startAngle		num	-360..360	0	Dimension:[deg]
arcAngle		num	-360..360	90	Dimension:[deg]
lineWidth		num[]	0..100	1	Dimension:[pt]
lineColor		rgba	0..255	black	
linePattern		int	1..128	black	

Examples:

AddArc(50;50;250;150)
 AddArc(50;50;250;150;180;180;2)
 AddArc(50;50;250;150;180;180;3 10 5;violet)

```
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		rgba	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	0..+9e99		
height	•	num	0..+9e99		

Examples:

```
AddClipOval(diff;0;0;400;300)
AddClipOval(;50;50;150;150)
```

Notes:

Clipping functions are only supported in Windows OS.

```
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	•	num[]	-9e99..+9e99		

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	0..+9e99		
height	•	num	0..+9e99		

Examples:

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

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

Notes:

Clipping functions are only supported in Windows OS.

AddClipReset()

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

Examples:

```
AddClipReset()
```

Notes:

Clipping functions are only supported in Windows OS.

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	0..+9e99		
height	•	num	0..+9e99		
hCurvature		num	0..100	12	Dimension:[pt]
vCurvature		num	0..100	12	Dimension:[pt]

Examples:

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

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

Notes:

Clipping functions are only supported in Windows OS.

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

Arguments:	req.	type	range	default	note
type		int	1..3	sect	
left	•	num	-9e99..+9e99		
top	•	num	-9e99..+9e99		
width	•	num	0..+9e99		
height	•	num	0..+9e99		
startAngle		num	-360..360	0	Dimension:[deg]
arcAngle		num	-360..360	90	Dimension:[deg]
innerRadius		num	0..100	0	[%] of slice radius

Examples:

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

Notes:

Clipping functions are only supported in Windows OS.

AddClipSmoothPolygon(type;scanDirection;listOfCoords)

Arguments:	req.	type	range	default	note
type		int	1..3	sect	
scanDirection		int	1..2	xyxy	
listOfCoords	•	num[]	-9e99..+9e99		

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)
```

Notes:

Clipping functions are only supported in Windows OS.

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

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

Example:

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

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

Arguments:	req.	type	range	default	note
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	0...+9e99		
height	•	num	0...+9e99		
frameWidth		num[]	0..100	1	Dimension:[pt]
frameColor		rgba	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)

Arguments:	req.	type	range	default	note
hStart	•	num	-9e99...+9e99		
vStart	•	num	-9e99...+9e99		
hEnd	•	num	-9e99...+9e99		
vEnd	•	num	-9e99...+9e99		
lineWidth		num[]	0..100	1	Dimension:[pt]
lineColor		rgba	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)

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

Examples:

AddOval(50;20;150;150)

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

AddOval(50;50;250;150;0 100 100 200)

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

Arguments:	req.	type	range	default	note
pathData	•	num[]	-9e99...+9e99		
fillColor		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	0..255	gray	
shadowPattern		int	1..128	black	

Example:

```
AddPath(2 50 50 // move to 50 50
3 100 50 // line to 100 50
3 100 100 // line to 100 100
1; // close path
darkYellow;; // fill
3;darkRed) // border
```

Notes:

Path constants: 1...close path, 2...move to,
3...line to, 4...quadratic Bézier to,
5...cubic Bézier to, 6...elliptical arc to

**AddPicture(left;top;width;height;sourceType;sourceName
location;adjustment;isProportional)**

Arguments:	req.	type	range	default	note
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.
location		int	1..9	centerCenter	
adjustment		int	1..5	reduceOrEnlarge	
isProportional		int	0..1	off	

Examples:

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

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	xxyy	
listOfCoords	•	num[]	-9e99..+9e99		
fillColor		rgba	0..255	black	
fillPattern		int	1..128	black	

Examples:

```
AddPolygon(xxyy;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	xxyy	
listOfCoords	•	num[]	-9e99..+9e99		
lineWidth		num[]	0..100	1	Dimension:[pt]
lineColor		rgba	0..255	black	
linePattern		int	1..128	black	

Examples:

```
AddPolyline(xxyy;50 50 150 150 150 100 50 250)
AddPolyline(xyxy;50 50 150 150 150 100 50 250;2;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	0..+9e99		
height	•	num	0..+9e99		
fillColor		rgba	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	0...+9e99		
height	•	num	0...+9e99		
hCurvature		num	0..100	12	Dimension:[pt]
vCurvature		num	0..100	12	Dimension:[pt]
frameWidth		num[]	0..100	1	Dimension:[pt]
frameColor		rgba	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	0...+9e99		
height	•	num	0...+9e99		
hCurvature		num	0..100	12	Dimension:[pt]
vCurvature		num	0..100	12	Dimension:[pt]
fillColor		rgba	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; innerRadius;fillColor;fillPattern)

Arguments:	req.	type	range	default	note
left	•	num	-9e99...+9e99		
top	•	num	-9e99...+9e99		
width	•	num	0...+9e99		
height	•	num	0...+9e99		
startAngle		num	-360...360	0	Dimension:[deg]
arcAngle		num	-360...360	90	Dimension:[deg]
innerRadius		num	0...100	0	[%] of slice radius
fillColor		rgba	0..255	black	
fillPattern		int	1..128	black	

Examples:

```
AddSlice(50;50;250;150;-45;90)
AddSlice(50;50;250;150;45;90;20;red)
```

AddSmoothPolygon(scanDirection;listOfCoords;fillColor; fillPattern)

Arguments:	req.	type	range	default	note
scanDirection		int	1..2	xxyy	
listOfCoords	•	num[]	-9e99...+9e99		
fillColor		rgba	0..255	black	
fillPattern		int	1..128	black	

Examples:

```
AddSmoothPolygon(xxyy;350 50 250 150 250 100 350 250 350 50;red)
AddSmoothPolygon(xxyy;350 50 250 150 250 100 350 250 350 50;0 90 200)
```

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

Arguments:	req.	type	range	default	note
scanDirection		int	1..2	xxyy	
listOfCoords	•	num[]	-9e99...+9e99		
lineWidth		num[]	0...100	1	Dimension:[pt]
lineColor		rgba	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)

Arguments:	req.	type	range	default	note
hPosition	•	num	-9e99..+9e99		
vPosition	•	num	-9e99..+9e99		
symbolType		int	0..18	bullet	
symbolSize		num	0..100	9	Dimension:[pt]
lineWidth		num[]	0..100	1	Dimension:[pt]
symbolColor		rgba	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;hAlignment;vAlignment;orientation;maxWidth;maxHeight;ellipsisPosition)

Arguments:	req.	type	range	default	note
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		rgba	0..255	black	
hAlignment		int	1..3	left	
vAlignment		int	1..4	baseline	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-1..10000	-1	Dimension:[pt]
maxHeight		num	-1..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

Examples:

AddText(282;295;"X2max Software";"Times";10:bold;blue)

AddText(205;155;"Element-2";"Verdana";16:bold+underline;red;;;-90)

AreaChart(appearanceConstants;doShiftIntervals)

Arguments:	req.	type	range	default	note
appearanceConst.		int	0..127	default	
doShiftIntervals		int	0..1	off	

Examples:

AreaChart()

AreaChart(shadow+horizontal+label;on)

AreaChart2D(appearanceConstants;fillAreaToAxis)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
appearanceConst.	int		0..127	default	
fillAreaToAxis	int		0..2	0	

Examples:

```
AreaChart2D()
```

```
AreaChart2D(shadow;1)
```

AreaChartOptions(useLineStyle;referenceValue;splitPosNegStacks)

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

Examples:

```
AreaChartOptions(on)
```

```
AreaChartOptions(on;;off)
```

Notes:

AreaChartOptions() should be entered after the AreaChart() function.

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..10000	all	
lineWidth	num[]		0..100	1	Dimension:[pt]
lineColor	rgba		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..10000	all	
fillColor		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	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..10000	all	
location		int	0..9	(autom.)	
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;
orientation;maxWidth;maxHeight;ellipsisPosition)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..10000	all	
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgba	0..255	black	
alignment		int	1..3	center	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-1..10000	-1	Dimension:[pt]
maxHeight		num	-1..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

Examples:

```
AxisLabelStyle(all;"Times";12)
AxisLabelStyle(x;;;bold;;;-45)
```

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..10000	all	
text1..10000		str	0..1000	" "	max. 1000 chars/text

Examples:

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

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..10000	all	
width		num[]	0..100	1	Dimension:[pt]
color		rgba	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..10000	all	
fillColor		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	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)**

Arguments:	req.	type	range	default	note
axisIndex		int	0..10000	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;orientation;maxWidth;
maxHeight;ellipsisPosition)**

Arguments:	req.	type	range	default	note
axisIndex		int	0..10000	all	
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgba	0..255	black	
alignment		int	1..3	center	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-1..10000	-1	Dimension:[pt]
maxHeight		num	-1..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

Examples:

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

AxisMajorTickLabelTexts(axisIndex;text1;text2...)

Arguments:	req.	type	range	default	note
axisIndex		int	0..10000	all	
text1..10000		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..10000	all	
length		num	0..100	5	Dimension:[pt]
width		num[]	0..100	1	Dimension:[pt]
color		rgba	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..10000	all	
fillColor		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	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;
doRepeatLabelPattern)**

Arguments:	req.	type	range	default	note
axisIndex		int	0..10000	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	
doRepeatLabelPattern			0..1	on	

Examples:

```
AxisMinorTickLabelOptions(x;out;;;2)
AxisMinorTickLabelOptions(y;in;;;2;1;off)
```

**AxisMinorTickLabelStyle(axisIndex;font;size;style;color;
alignment;orientation;maxWidth;
maxHeight;ellipsisPosition)**

Arguments:	req.	type	range	default	note
axisIndex		int	0..10000	all	
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgba	0..255	black	
alignment		int	1..3	center	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-1..10000	-1	Dimension:[pt]
maxHeight		num	-1..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

Examples:

```
AxisMinorTickLabelStyle(all;;;8)
AxisMinorTickLabelStyle(y;"Courier";9;;gray;;-45)
```

AxisMinorTickLabelTexts(axisIndex;text1;text2...)

Arguments:	req.	type	range	default	note
axisIndex		int	0..10000	all	
text1..10000		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..10000	all	
length		num	0..100	3	Dimension:[pt]
width		num[]	0..100	1	Dimension:[pt]
color		rgba	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..10000	all	
axisLocation		int	0..2	back	
doShiftAxis		int	0..1	off	

Examples:

```
AxisOptions(x;;on)
AxisOptions(y;;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		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	0..255	gray	
shadowPattern		int	1..128	black	

Examples:

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

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

Arguments:	req.	type	range	default	note
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(resource;"41")
BackgroundPict(file;"Backgrounds/Image012.jpg")
BackgroundPict(file;"C:/Images/Gradient_01.jpg")
BackgroundPict(file;"Macintosh HD/Images/BackGrd_01.png")
```

Notes:

BarChart (appearanceConstants;categoryGap;seriesGap;barDepth)

Arguments:	req.	type	range	default	note
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	in % of bar width

Examples:

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

Notes:

default series gap for non-stacked bar charts: 0 [%]
 default series gap for stacked bar charts: -100 [%]

BarChart2D(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	in % of bar width

Examples:

```
BarChart2D()
BarChart2D(stacked;25)
BarChart2D(shadow+proportional;0;0;50)
BarChart2D(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;shape;width;color;pattern)

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

Examples:

```
BorderStyle(1;poly;2)
BorderStyle(2;smooth;2;black)
BorderStyle(all;none)
```

**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;confidenceInterval)**

<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
confidenceInterv.		num	0..99.9999		in % probability

Examples:

BoxPlotOptions(;;;on;on;on)
BoxPlotOptions(50;on)
BoxPlotOptions(;on;on;on;on;off;off;;95)

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)
```


**CandlestickChart2D(appearanceConstants;itemGap;
highTickMarkLength;highTickMarkAlignment;
lowTickMarkLength;lowTickMarkAlignment)**

Arguments:	req.	type	range	default	note
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:

CandlestickChart2D(;;5;right;5;left)

CandlestickChart2D(horizontal+shadow;50)

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

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

Examples:

ChartBackground(all;yellow)

ChartBackground(xy;lightGray;;0)

**ChartBackgroundPict(planeIndex;sourceType;sourceName;location;
adjustment;isProportional)**

Arguments:	req.	type	range	default	note
planeIndex	int		0..3	all	
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:

```

ChartBackgroundPict(xy;clipboard)
ChartBackgroundPict(all;resource;"27")
ChartBackgroundPict(;file;"Backgrounds/Pict012.tif")
ChartBackgroundPict(xy;file;"C:/Images/Gradient_01.bmp")
ChartBackgroundPict(;file;"Macintosh HD/Picts/Gradient_01.jpg")

```

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

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
dataSeries1...		num[]	-9e99...+9e99		max. 10000 series

Examples:

```

ChartData(12 56 -34 67 22)
ChartData(2 3 1.32 9; -0.2 5 0 3 8; 1 4 .2)
ChartData(2008-12-31 2009-01-22; 2009-12-12)
ChartData(12/31/2008&14:00 12/31/2008&18:15 1/1/2009&7:53)

```

ChartDataLowerLimits(minValue1;minValue2...)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
minValue1..10000		num	-9e99...+9e99	-9e99	max. 10000 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		str	1..1	"\n"	
elementSeparator		str	1..1	"\t"	

Examples:

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

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

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
maxValue1..10000		num	-9e99..+9e99	+9e99	max. 10000 values

Examples:

```
ChartDataUpperLimits(0)
ChartDataUpperLimits(9999)
```

Notes:

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

```
ChartDataWrite(fileName;fileFlag;creatorType;doTranspose;
seriesSeparator;elementSeparator;
format1;format2...)
```

<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	"????"	Mac OS X only
doTranspose		int	0..1	off	
seriesSeparator		str	1..1	"\n"	
elementSeparator		str	1..1	"\t"	
format1..10000		str	0..1000	" u "	max.1000 chars/form.

Examples:

```
ChartDataWrite("C:/Programs/Data/exportdata.txt")
ChartDataWrite("Macintosh HD/Data/Plots/data_01")
ChartDataWrite("ChartData";replace;;;";";" ";"|i0|";"|f2|")
```

Notes:

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()
```

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..10000	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..10000	all	
type		int	-4..10	linear	
width		num[]	0..100	1	Dimension:[pt]
color		rgba	0..255	black	
pattern		int	1..128	black	

Examples:

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

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

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..10000	all	
type		int	-4..10	linear	
doSwitchAxes		int	0..1	off	
doExtrapolate		int	0..1	off	
doForceThruZero		int	0..1	off	

Examples:

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

DateTimeOptions(dateOrder;startingDay)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
dateOrder		int	1..3	mdy	mdy = month day year
startingDay		int	1..7	1	

Examples:

```

DateTimeOptions(ymd) // year|month|day
DateTimeOptions(mdy;1) // US date format
DateTimeOptions(;2) // week begins on Monday (ISO 8601)

```

Notes:

Place DateTimeOptions() before ChartData()!

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..10000	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		rgba	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..10000	all	
xCenter	•	num	-9e99..+9e99		
yCenter	•	num	-9e99..+9e99		
symbolType		int	0..18	bullet	
symbolSize		num	0..100	9	
lineWidth		num[]	0..100	1	Dimension:[pt]
symbolColor		rgba	0..255	black	Dimension:[pt]
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..10000	all	
refSeries1..1000		int	0..10000	all	max. 1000 series

Examples:

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

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..10000	all	
dropLineAxisIndex		int	0..3	all	
width		num[]	0..100	1	Dimension:[pt]
color		rgba	0..255	gray	
pattern		int	1..128	black	

Examples:

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

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..10000	all	
valueList1		num[]	-9e99..+9e99	0	pos. x-error values
valueList2		num[]	-9e99..+9e99	0	neg. x-error values
valueList3		num[]	-9e99..+9e99	0	pos. y-error values
valueList4		num[]	-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;add1Data1;
add1Data2;add1Data3;add1Data4)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..10000	all	
axisIndex		int	0..3	all	
errorDirection		int	0..3	both	
type		int	0..5	stdError	
add1Data1		num	0..+9e99	see notes	data for pos.x-error
add1Data2		num	0..+9e99	see notes	data for neg.x-error
add1Data3		num	0..+9e99	see notes	data for pos.y-error
add1Data4		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: add1DataN contains standard deviations, default: 1

if type=percent: add1DataN contains percent values, default: 5 [%]

if type=constant: add1DataN 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..10000	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		rgba	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)

Arguments:	req.	type	range	default	note
seriesIndex		int	0..10000	all	
shapeType		int	0..2	oval	
fillColor		rgba	0..255	gray	
fillPattern		int	1..128	transparent	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	

Examples:

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

FillStyle(seriesIndex;color;pattern)

Arguments:	req.	type	range	default	note
seriesIndex		int	0..10000	all	
color		rgba	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)

Arguments:	req.	type	range	default	note
appearanceConst.		int	0..127	default	
categoryGap		num	0..1000	100	in % of bar width
barDepth		num	0..1000	0	in % of bar width

Examples:

```
GanttChart(label)
GanttChart(shadow+horizontal;50;30)
```

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		rgba	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) // hide grid
```

**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)
```

```
HighLowChart2D(appearanceConstants;  
                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	
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:

```
HighLowChart2D()  
HighLowChart2D(horizontal;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..10000	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..10000	all	
fillColor		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	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..10000	all	
location		int	0..9	(autom.)	
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;
orientation;maxWidth;maxHeight;ellipsisPosition)**

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	-1..10000	all	
font		str	0..255	"ApplFont"	Mac:Geneva/Win:Arial
size		int	0..127	9	Dimension:[pt]
style		int	0..7	plain	
color		rgba	0..255	black	
alignment		int	1..3	center	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-2..10000	-1	Dimension:[pt]
maxHeight		num	-2..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

Examples:

```
LabelStyle(all;;;bold)
LabelStyle(2;"Courier";10;plain;gray;center;-45)
```

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..10000	all	
text1..10000		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		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	0..255	gray	
shadowPattern		int	1..128	black	

Examples:

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

LegendOptions(location;placeInside;hOffset;vOffset;distribution;markerType;markerWidth;markerHeight;markerGap;rowGap;columnGap;textLocation;markerShape)

<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]
distribution		int[]	-1..1000	10	see notes
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]
textLocation		int	1..9	centerRight	
markerShape		int	1..3	1	rect,oval,rdRect

Examples:

```

LegendOptions(bottomRight;on;;;1;15;15)
LegendOptions(;;;2) // 2 rows (variable num of columns)
LegendOptions(;;;2 -1) // 2 rows (same as above)
LegendOptions(;;;-1 2) // 2 columns (variable num of rows)
LegendOptions(;;;3 -1 1) // 3 rows with layout column by col.
LegendOptions(;;;-1 1 0 1) // 1 column, bottom to top

```

Notes: distribution flags:

[1]: numRows, range: -1..1000, default: 10, (-1...variable)
 [2]: numColumns, range: -1..1000, default: -1 (-1...variable)
 [3]: doArrangeColumnByColumn, range: 0..1, default: 0
 [4]: doArrangeReversed, range 0..1, default: 0
 [5]: useEquidstantColumnWidths, range: 0..1, default: 0
 [6]: useEquidstantRowHeights, range: 0..1, default: 0

LegendStyle(font;size;style;color;alignment;orientation;maxWidth;maxHeight;ellipsisPosition)

<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		rgba	0..255	black	
alignment		int	1..3	left	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-1..10000	-1	Dimension:[pt]
maxHeight		num	-1..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

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..10000		str	0..1000	" "	max. 1000 chars/text

Examples:

```

LegendTexts("Turnover\nDepartment A";"Turnover\nDepartment B")

```

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;shape;width;color;pattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..10000	all	
shape		int	0..4	poly	
width		num[]	0..100	1	Dimension:[pt]
color		rgba	0..255	see notes	
pattern		int	1..128	black	

Examples:

```
LineStyle(1;smooth;2)
LineStyle(2;step)
LineStyle(all;poly;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..1000		rgba	0..255	gray	max. 1000 colors

Examples:

```
MajorGridLineColors(all;all;black)
MajorGridLineColors(x;y;lightGray;gray)
```

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..1000		int	1..128	black	max. 1000 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..1000		num[]	0..100	1	max.1000 line widths

Examples:

```
MajorGridLineWidths(x;y;1;2) // horizontal grid lines
MajorGridLineWidths(y;x;0) // hide vertical grid lines
MajorGridLineWidths(y;x;1 2 2) // dotted 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..1000		rgba	0..255	gray	max. 1000 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..1000		int	1..128	black	max. 1000 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..1000		rgba	0..255	gray	max. 1000 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..1000		int	1..128	black	max. 1000 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..1000		num[]	0..100	1	max.1000 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..1000		rgba	0..255	lightGray	max. 1000 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..1000		int	1..128	black	max. 1000 patterns

Examples:

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

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

MovingAverage(seriesIndex;numOfIntervals; calculationMethod;weightList)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..10000	all	
numOfIntervals		int	1..10000	2	
calculationMethod		int	0..3	average	
weightList		num[]	-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; calculationMethod;shape;width; color;pattern)

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
seriesIndex		int	0..10000	all	
numOfIntervals		int	1..10000	2	
calculationMethod		int	0..3	average	
shape		int	0..4	poly	
width		num[]	0..100	1	Dimension:[pt]
color		rgba	0..255	black	
pattern		int	1..128	black	

Examples:

MovingAverageLineStyle(1;50;average;poly;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..10000	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)           // extrapolate  
MovingAverageOptions(1;100;;;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;antialiasing)

<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	-1..1	0	see note 1
antialiasing		int	0..3	3	see note 2

Examples:

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

```
OpenDrawing(400;300;1;2)
```

Note 1:

type=0: xmCHART creates a vector graphic (EMF on Windows,PDF on Mac OS X)

type=1: xmCHART creates a bitmap graphic

type=-1: Windows OS: xmCHART creates a PDF vector graphic.

type=-1 is not required for SaveAsPDFFile().

Mac OS X: xmCHART creates a PDF vector graphic (same as type=0)

Note 2:

antialiasing=0: no antialiasing

antialiasing=1: antialiasing geometric objects

antialiasing=2: antialiasing texts

antialiasing=3: antialiasing geometric objects and texts (default)

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..10000	all	
sourceType		int	1..3	clipboard	
sourceName		str	0..1000	" "	max. 1000 chars.
stackAndScaleAt		num	0..+9e99	0	

Examples:

```
PictureStyle(1;clipboard)
PictureStyle(2;resource;"17")
PictureStyle(3;file;"Images/Gradient_03.tif")
PictureStyle(all;file;"C:/Programs/Fills/Gradient003.png")
PictureStyle(1;file;"Macintosh HD/Fills/Gradient_01.pdf")
```

**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..200	0	in % of pie radius
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(label+oval;;80)
PieChart(label;20;80;-135;270)
```

**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	not implemented.
vAlignment		int	1..3	center	
width		num[]	0..100	1	Dimension:[pt]
color		rgba	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		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	0..255	gray	
shadowPattern		int	1..128	black	

Examples:

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

**PieChartCenterLabelStyle(font;size;style;color;alignment;
orientation;maxWidth;maxHeight;
ellipsisPosition)**

<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		rgba	0..255	black	
alignment		int	1..3	center	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-1..10000	-1	Dimension:[pt]
maxHeight		num	-1..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

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|") // shows total
```

Notes:

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

PieChartExplodeDepths(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 depth
sliceIndex1..1000		int	-3..1000	none	

Examples:

```
PieChartExplodeDepths(15;2)
PieChartExplodeDepths(20;max)
PieChartExplodeDepths(20;2;4;6;8)
```

Notes:

The functions PieChartExplodes() and PieChartExplodeDepths() can be entered several times and combined as well. Therefore, it is possible to move segments outwards radially and/or upwards vertically, just as you wish. PieChartExplodeDepths() should be entered after the PieChart() function.

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:

The functions PieChartExplodes() and PieChartExplodeDepths() can be entered several times and combined as well. Therefore, it is possible to move segments outwards radially and/or upwards vertically, just as you wish. 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		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	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;
orientation;maxWidth;maxHeight;
ellipsisPosition)**

<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		rgba	0..255	black	
alignment		int	1..3	left	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-1..10000	-1	Dimension:[pt]
maxHeight		num	-1..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

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..10000		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..10000	1	0...no scaling
gridShape	int		0..3	oval	
doAddArrows	int		0..1	off	
doNotClosePolygon	int		0..1	off	
numOfAxes	int		2..100	12	

Examples:

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

Notes:

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

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

<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]
doShiftIntervals	int		0..1	off	

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..10000	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	"????"	Mac OS X only

Examples:

```
SaveAsBMPFile("Chart.bmp";replace)
SaveAsBMPFile("Graphs/Graph_A.bmp")
SaveAsBMPFile("C:/Programs/Plots/Plot_01.bmp")
SaveAsBMPFile("Macintosh HD/Plots/Plot_01.bmp")
```

*Notes:***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("Graphs/Graph_A.emf")
SaveAsEMFFile("C:/Programs/Plots/Plot_01.emf")
```

Notes:

Windows only.

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";replace)
SaveAsGIFFile("Graphs/Graph_A.gif")
SaveAsGIFFile("C:/Programs/Plots/Plot_01.gif")
```

Notes:

Windows only.

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	"????"	Mac OS X only
compression		int	1..5	normal	

Examples:

```
SaveAsJPGFile("Chart.jpg";replace)
SaveAsJPGFile("Graphs/Graph_A.jpg")
SaveAsJPGFile("C:/Programs/Plots/Plot_01.jpg";replace;;max)
SaveAsJPGFile("Macintosh HD/Plots/Plot_01.jpg")
```

*Notes:***SaveAsPDFFile(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	"????"	Mac OS X only

Examples:

```
SaveAsPDFFile("Chart.pdf";replace)
SaveAsPDFFile("C:/Programs/Diagrams/Diagram.pdf";replace)
SaveAsPDFFile("Macintosh HD/Plots/Plot_01.pdf")
```

*Notes:***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	"????"	Mac OS X only

Examples:

```
SaveAsPICTFile("Chart.pct";replace;"8BIM")
SaveAsPICTFile("Graphs/Graph_A.pct")
SaveAsPICTFile("Macintosh HD/Plots/Plot_01.pct")
```

Notes:

Mac OS 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	"????"	Mac OS X only

Examples:

```
SaveAsPNGFile("Chart.png";replace)
SaveAsPNGFile("Graphs/Chart_A.png")
SaveAsPNGFile("C:/Programs/Plots/Plot_01.png")
SaveAsPNGFile("Macintosh HD/Plots/Plot_01.png")
```

*Notes:***SaveAsSVGFile(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	"????"	Mac OS X only

Examples:

```
SaveAsSVGFile("Chart.svg";replace)
SaveAsSVGFile("Graphs/Chart_A.svg")
SaveAsSVGFile("C:/Programs/Plots/Plot_01.svg")
SaveAsSVGFile("Macintosh HD/Plots/Plot_01.svg")
```

*Notes:***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	"????"	Mac OS X only

Examples:

```
SaveAsTIFFFile("Chart.tif";replace)
SaveAsTIFFFile("Graphs/Chart_A.tif")
SaveAsTIFFFile("C:/Programs/Plots/Plot_01.tif")
SaveAsTIFFFile("Macintosh HD/Plots/Plot_01.tif")
```

Notes:

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

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..10000	all	
type		int	1..3	linear	
minValue		num	-9e99..+9e99	(autom.)	
maxValue		num	-9e99..+9e99	(autom.)	
numOfMajorInt.		int	-18..1000	(autom.)	
numOfMinorInt.		int	-18..1000	1	
logBaseValue		num	1.0001..+9e99	10	
useEquidistantLogScaling			0..1	off	

Examples:

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

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

<i>Arguments:</i>	<i>req.</i>	<i>type</i>	<i>range</i>	<i>default</i>	<i>note</i>
axisIndex		int	0..10000	all	
doReverseScaling		int	0..1	off	
useIntegersOnly		int	0..1	off	
hideZeroLabel		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()
```

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("") // no thousands separator (default)
SetThousandsSep(",") // e.g.: 1234567 -> 1,234,567
SetThousandsSep("'") // e.g.: 1234567 -> 1'234'567
```

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..10000	all	
offset		num	-100..100	3	Dimension:[pt]
color		rgba	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..10000	all	
type		int	0..18	cross	
size		num	0..100	9	Dimension:[pt]
lineWidth		num[]	0..100	1	Dimension:[pt]
color		rgba	0..255	see notes	
pattern		int	1..128	black	

Examples:

```
SymbolStyle(all;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		rgba	0..255	white	
fillPattern		int	1..128	black	
borderWidth		num[]	0..100	1	Dimension:[pt]
borderColor		rgba	0..255	black	
borderPattern		int	1..128	black	
shadowOffset		num	-100..100	0	Dimension:[pt]
shadowColor		rgba	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;orientation;
maxWidth;maxHeight;ellipsisPosition)**

<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		rgba	0..255	black	
alignment		int	1..3	center	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-1..10000	-1	Dimension:[pt]
maxHeight		num	-1..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

Examples:

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

**TitleSubStyle(font;size;style;color;alignment;orientation;
maxWidth;maxHeight;ellipsisPosition)**

<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		rgba	0..255	black	
alignment		int	1..3	center	
orientation		num	-360..360	0	Dimension:[deg]
maxWidth		num	-1..10000	-1	Dimension:[pt]
maxHeight		num	-1..10000	-1	Dimension:[pt]
ellipsisPos.		int	0..4	3	

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";"Department A")  
TitleText("Turnover\nDepartment A")
```

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

Constants

The following section lists all constants available in xmCHART 3.4. Most constants can be accessed by using an easy-to-remember name. For example, the constant *indigo* can be used instead of the RGB value (275 0 130).

Constant names are not 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.

Chart function	shadow	label	symbol	horizontal	stacked	proportional	oval
AreaChart()	•	•	•	•	•	•	
AreaChart2D()	•	•	•				
BarChart()	•	•	•	•	•	•	
BarChart2D()	•	•	•	•	•	•	
BoxPlot()	•		•	•			
BubbleChart()	•	•	•	•			
BubbleChart2D()	•	•	•				
CandlestickChart()	•			•			
CandlestickChart2D()	•			•			
GanttChart()	•	•		•			
HighLowChart()	•	•	•	•			
HighLowChart2D()	•	•	•	•			
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 (see Fig. 2).

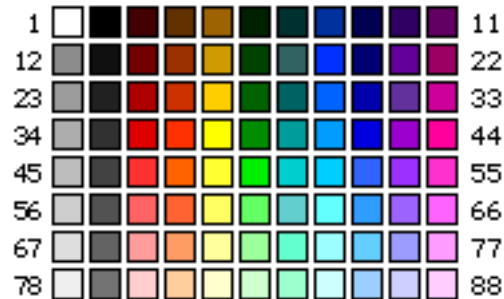


Fig. 2

Frequently used colors can be accessed by name.

<i>Constants</i>	<i>RedGreen Blue</i>			<i>hexCode</i>
aliceblue	240	248	255	#f0f8ff
antiquewhite	250	235	215	#faebd7
aqua	0	255	255	#00ffff
aquamarine	127	255	212	#7fffd4
azure	240	255	255	#f0ffff
beige	245	245	220	#f5f5dc
bisque	255	228	196	#ffe4c4
black	0	0	0	#000000
blanchedalmond	255	235	205	#ffebcd
blue	0	0	221	#0000dd
blueviolet	138	43	226	#8a2be2
brown	165	42	42	#a52a2a
burlywood	222	184	135	#deb887
cadetblue	95	158	160	#5f9ea0
chartreuse	127	255	0	#7fff00
chocolate	210	105	30	#d2691e
coral	255	127	80	#ff7f50
cornflowerblue	100	149	237	#6495ed
cornsilk	255	248	220	#fff8dc
crimson	220	20	60	#dc143c
cyan	0	255	255	#00ffff
darkblue	0	0	119	#000077
darkcyan	0	139	139	#008b8b

darkgoldenrod	184	134	11	#b8860b
darkgray	102	102	102	#666666
darkgreen	0	102	0	#006600
darkkhaki	189	183	107	#bdb76b
darkmagenta	139	0	139	#8b008b
darkolivegreen	85	107	47	#556b2f
darkorange	255	140	0	#ff8c00
darkorchid	153	50	204	#9932cc
darkpurple	204	0	153	#cc0099
darkred	204	51	0	#cc3300
darksalmon	233	150	122	#e9967a
darkseagreen	143	188	143	#8fbc8f
darkslateblue	72	61	139	#483d8b
darkslategray	47	79	79	#2f4f4f
darkturquoise	0	206	209	#00ced1
darkviolet	148	0	211	#9400d3
darkyellow	255	204	0	#ffcc00
deeppink	255	20	147	#ff1493
deepskyblue	0	191	255	#00bfff
dimgray	105	105	105	#696969
dodgerblue	30	144	255	#1e90ff
firebrick	178	34	34	#b22222
floralwhite	255	250	240	#fffaf0
forestgreen	34	139	34	#228b22
fuchsia	255	0	255	#ff00ff
gainsboro	220	220	220	#dcdcdc
ghostwhite	248	248	255	#f8f8ff
gold	255	215	0	#ffd700
goldenrod	218	165	32	#daa520
gray	136	136	136	#888888
green	0	136	0	#008800
greenyellow	173	255	47	#adff2f
honeydew	240	255	240	#f0fff0
hotpink	255	105	180	#ff69b4
indianred	205	92	92	#cd5c5c
indigo	75	0	130	#4b0082
ivory	255	255	240	#fffff0
khaki	240	230	140	#f0e68c
lavender	230	230	250	#e6e6fa
lavenderblush	255	240	245	#fff0f5
lawngreen	124	252	0	#7cfc00
lemonchiffon	255	250	205	#fffacd

lightblue	153	204	255	#99ccff
lightcoral	240	128	128	#f08080
lightcyan	224	255	255	#e0ffff
lightgoldenrodyellow	250	250	210	#fafad2
lightgray	221	221	221	#dddddd
lightgreen	153	255	153	#99ff99
lightpink	255	182	193	#ffb6c1
lightpurple	255	153	255	#ff99ff
lightred	255	153	153	#ff9999
lightsalmon	255	160	122	#ffa07a
lightseagreen	32	178	170	#20b2aa
lightskyblue	135	206	250	#87cefa
lightslategray	119	136	153	#778899
lightsteelblue	176	196	222	#b0c4de
lightyellow	255	255	153	#ffff99
lime	0	255	0	#00ff00
limegreen	50	205	50	#32cd32
linen	250	240	230	#faf0e6
magenta	255	0	255	#ff00ff
maroon	128	0	0	#800000
mediumaquamarine	102	205	170	#66cdaa
mediumblue	0	0	205	#0000cd
mediumorchid	186	85	211	#ba55d3
mediumpurple	147	112	219	#9370db
mediumseagreen	60	179	113	#3cb371
mediumslateblue	123	104	238	#7b68ee
mediumspringgreen	0	250	154	#00fa9a
mediumturquoise	72	209	204	#48d1cc
mediumvioletred	199	21	133	#c71585
midnightblue	25	25	112	#191970
mintcream	245	255	250	#f5fffa
mistyrose	255	228	225	#ffe4e1
moccasin	255	228	181	#ffe4b5
navajowhite	255	222	173	#ffdead
navy	0	0	128	#000080
oldlace	253	245	230	#fdf5e6
olive	128	128	0	#808000
olivedrab	107	142	35	#6b8e23
orange	255	165	0	#ffa500
orangered	255	69	0	#ff4500
orchid	218	112	214	#da70d6
palegoldenrod	238	232	170	#eee8aa

palegreen	152	251	152	#98fb98
paleturquoise	175	238	238	#afeeee
palevioletred	219	112	147	#db7093
papayawhip	255	239	213	#ffefd5
peachpuff	255	218	185	#ffdab9
peru	205	133	63	#cd853f
pink	255	192	203	#ffc0cb
plum	221	160	221	#dda0dd
powderblue	176	224	230	#b0e0e6
purple	255	0	153	#ff0099
red	255	51	0	#ff3300
rosybrown	188	143	143	#bc8f8f
royalblue	65	105	225	#4169e1
saddlebrown	139	69	19	#8b4513
salmon	250	128	114	#fa8072
sandybrown	244	164	96	#f4a460
seagreen	46	139	87	#2e8b57
seashell	255	245	238	#fff5ee
sienna	160	82	45	#a0522d
silver	192	192	192	#c0c0c0
skyblue	135	206	235	#87ceeb
slateblue	106	90	205	#6a5acd
slategray	112	128	144	#708090
snow	255	250	250	#fffafa
springgreen	0	255	127	#00ff7f
steelblue	70	130	180	#4682b4
tan	210	180	140	#d2b48c
teal	0	128	128	#008080
thistle	216	191	216	#d8bfd8
tomato	255	99	71	#ff6347
turquoise	64	224	208	#40e0d0
violet	238	130	238	#ee82ee
wheat	245	222	179	#f5deb3
white	255	255	255	#ffffff
whitesmoke	245	245	245	#f5f5f5
yellow	255	255	0	#ffff00
yellowgreen	154	205	50	#9acd32

Examples:

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

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)
```

Date orders

<i>Constants</i>	<i>Value</i>	<i>Note</i>
ymd	1	Year Month Day
mdy	2	Month Day Year
dmy	3	Day Month Year

Examples:

```
DateTimeOptions(ymd)
```

```
DateTimeOptions(mdy;2)
```

DateTime scaling constants

<i>Constants</i>	<i>Value</i>
year	-1
quarter	-2
month	-3
week	-4
day	-5
hour-	-6
minute30	-7
minute20	-8
minute15	-9
minute10	-10
minute5	-11
minute	-12
second30	-13

second20	-14
second15	-15
second10	-16
second5	-17
second	-18

Examples:

```
Scaling(x;linear;;;year;month)
```

```
Scaling(y;linear;0:00;24:00;hour;minute15)
```

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)
SaveAsPNGFile("Graphs/Graph_A.png";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("Charts/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>
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 shape 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 are also be accessed by name. (Fig. 3)

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

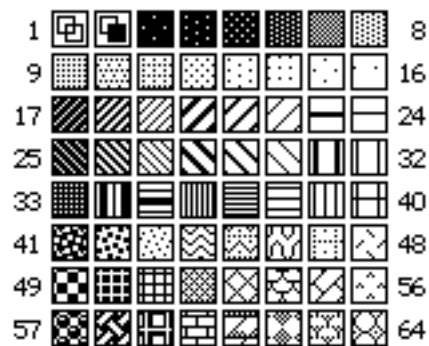


Fig. 3

Examples:

```
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)

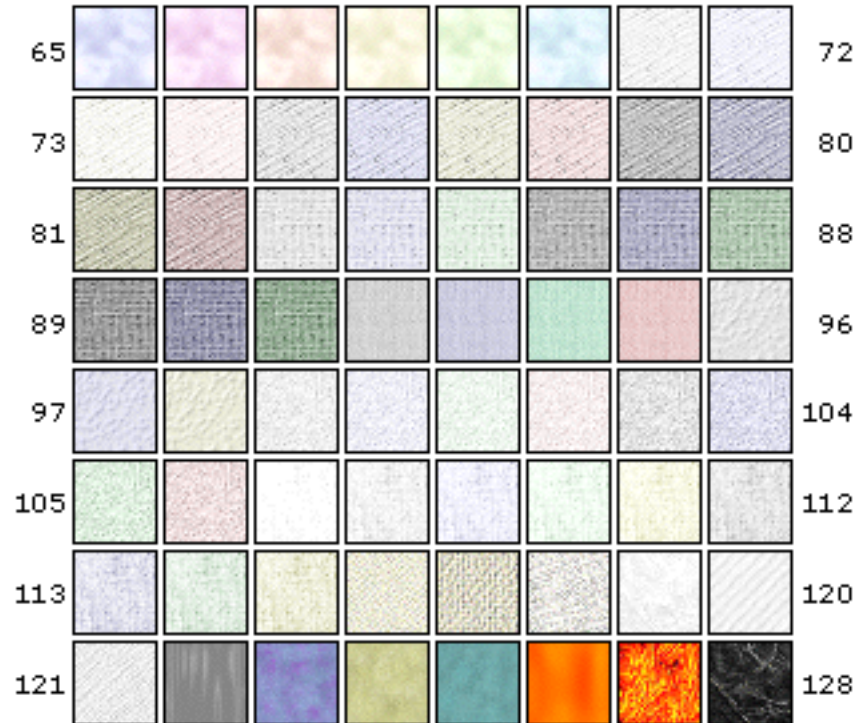


Fig. 4

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;;enlarge)
ChartBackgroundPict(xy;file;"Images/Background.jpg";;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")
```

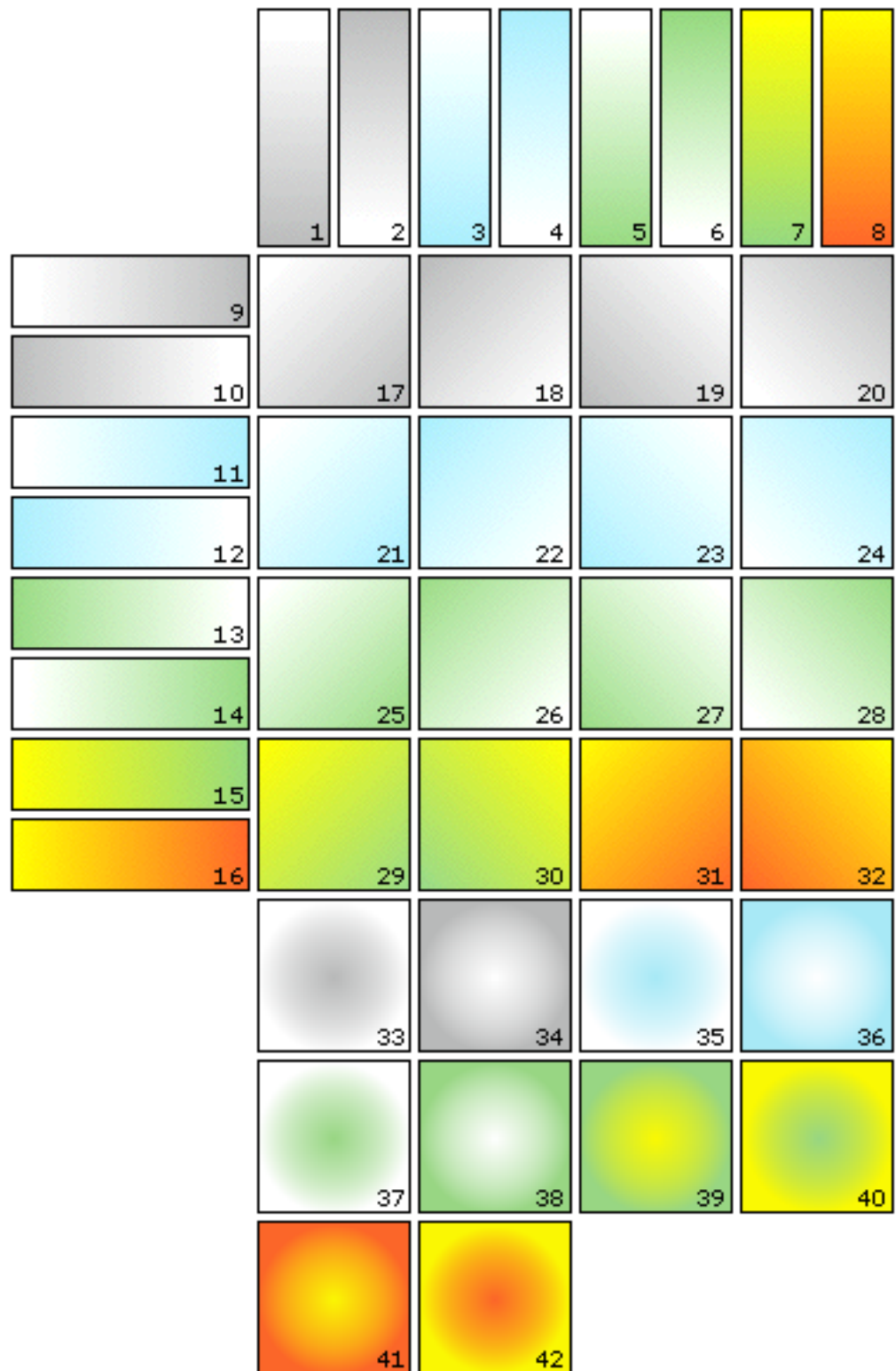


Fig. 5

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>
xyxy	1
xyxy	2

Examples:

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

Series indices

Constants	Value	
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.

Constants	Value
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

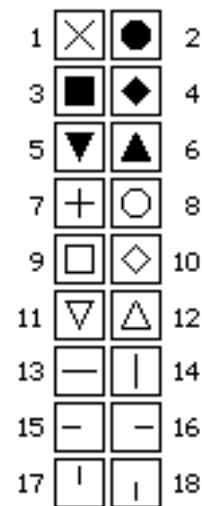


Fig. 6

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)
```

Vertical alignments

<i>Constants</i>	<i>Value</i>
top	1
center	2
bottom	3
baseline	4

Examples:

```
AddText(10;100;"Copyright";"Times";12:bold;;baseline)
```

```
AddText(10;100;"A\nB\nC";"Verdana";14:bold:center:center;90)
```

Format Specifiers

Number Format Specifiers

General structure

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

- Number 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 "	+1234.6
	" +-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)
0.00000012345678		
0.00012345678		

DateTime Format Specifiers

General structure

"[text] | specifiers | [text]"

DateTime format specifiers are to be placed in vertical bars "|" (pipe character). Leading and trailing texts can be attached as an option.

Year specifiers:

YY year without century
YYYY year with 4 digits

Quarter specifier:

Q quarter of year (1..4)
Qx quarter + English ordinal suffix, not localized
QX quarter + English ordinal suffix, not localized, uppercase

Month specifiers:

M month of year (1..12)
MM month of year with leading 0 (01..12)
Mo 1st letter of localized month name, uppercase
Mon abbr. month name, localized
MON abbr. month name, localized, uppercase
Month month name, localized
MONTH month name, localized, uppercase

Week specifiers:

W weekday index (Sunday=1, Monday=2,..., Saturday=7)
WY week of year (1..53), The starting day is defined by
 DateTimeOptions()
WWY week of year with leading 0 (01..53)
WD1 1st letter of localized weekday name, uppercase
Wd2 2-letter abbr. of localized weekday name
WD2 2-letter abbr. of localized weekday name, uppercase
Wd3 abbr. weekday name, localized
WD3 abbr. weekday name, localized, uppercase
Weekday weekday name, localized
WEEKDAY weekday name, localized, uppercase

Day specifiers:

D	day of month (1..31)
DD	day of month with leading 0 (01..31)
DY	day of year (1..366)
DDY	day of year with leading 0 (001..366)
Dx	English ordinal suffix, not localized
DX	English ordinal suffix, not localized, uppercase

Time specifiers:

h	hour in 24-hour format (0..23)
hh	hour in 24-hour format with leading 0 (00..23)
h12	hour in 12-hour format (1..12)
hh12	hour in 12-hour format with leading 0 (01..12)
m	minute (0..59)
mm	minute with leading 0 (00..59)
s	second (0..59)
ss	second with leading 0 (00..59)
ampm	am or pm (in combination with h12 or hh12)
AMPM	AM or PM (in combination with h12 or hh12)

Examples

<i>dateTime</i>	<i>format specifier</i>	<i>result</i>
7/31/2009	" M/D/YY "	7/31/09
	" D.M.YYYY "	31.7.2009
	" YYYYMMDD "	20090731
1:58:04	" h:mm:ss "	1:58:04
	" hh:mm "	01:58
	"Time: h:mm:ss "	Time: 1:58:04
14:05:32	" h12:mm ampm "	2:05 pm

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_GetErrorMessage(flags)` – `flags` is of type string!

		range	default
<code>flags[1]:</code>	<code>doPlayAlertSound</code>	0..1	"1"
<code>flags[2]:</code>	<code>doShowErrorCode</code>	0..1	"0"
<code>flags[3]:</code>	<code>descriptionFlag</code>	0..5	"3"
<code>flags[4..]:</code>	<code>messagePrefix</code>		" "

`descriptionFlag:` "0"...no description
 "1"...basic info
 "2"...basic info + location
 "3"...basic info + location + erroneous script part
 "4"...reserved
 "5"...reserved

Examples:

```
xmCH_GetErrorMessage("103")    // default.
xmCH_GetErrorMessage("110")    // play sound and
                                // show only error code.
xmCH_GetErrorMessage("110$$")  // play sound and show
                                // error code, e.g. $$1180
```

If no error occurs, the function `xmCH_GetErrorMessage()` 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: `xmCH_SetLanguage(0)`
German: `xmCH_SetLanguage(1)`

Error codes:

1000: Syntax error.
1001: Invalid comment.
1010: Invalid function name.
1020: Invalid font name.
1100: Too few arguments.
1101: Too many arguments.
1110: Too few elements.
1111: Too many elements.
1112: Invalid number of elements.
1120: Value out of range.
1130: Invalid String.
1131: String too short.
1132: String too long.
1140: Value error.
1150: Clipboard contains invalid data.
1160: Invalid constant.
1170: Not enough data.
1180: Can't find resource.
1200: Function `OpenDrawing(...)` missing.
1201: Multiple call of function: `OpenDrawing()`
1210: Function `CloseDrawing(...)` missing.
1211: Multiple call of function: `CloseDrawing()`
1220: Can't open new view.
1230: Function `OpenView(...)` missing.
1240: Nested chart definition.
1241: Function `OpenChart(...)` missing.
1242: No chart function defined.
1300: Invalid axis index.
2000: Invalid format.
2001: Invalid number format.
2002: Invalid date/time format.
2010: `ChartData(): Invalid value.`
2020: Invalid image dimension.
2100: Invalid scaling values."

2200: Frame too small to draw chart.
2300: Curve fitting: Not enough data.
2350: Moving Average: Not enough data.
2400: Path error.
4000: File not found.
4001: File exist error.
4002: File creation error.
4003: File open error.
4004: File read error.
4005: File write error.
4006: File close error.
4007: File exchange error.
4008: File delete error.
4009: File search error.
4010: File error. No such volume.
4011: File name error.
4012: File size error.
4013: Unknown file format.
4014: Directory not found.
5000: Out of memory.
5010: Division by zero.
5011: Invalid logarithmic value.
5012: Index out of range.

Index

//	11
\"	11
\\	11
\n	11
\r	11
\t	11
\uXXXX	11
	70
A	
AddArc	12
AddArrow	13
AddClipOval	13
AddClipPolygon	13
AddClipRect	14
AddClipReset	14
AddClipRoundRect	14
AddClipSlice	15
AddClipSmoothPolygon	15
addCounter	79
AddEllipse	15
AddFrame	16
AddLine	16
AddOval	16
AddPath	17
AddPicture	17
AddPolygon	18
AddPolyline	18
AddRect	18
AddRoundFrame	19
AddRoundRect	19
AddSlice	20
AddSmoothPolygon	20
AddSmoothPolyline	20
AddSymbol	21

AddText	21
aliceblue	73
all	71, 79, 89-90
AMPM	95
ampm	95
antiquewhite	73
Appearance constants	70
aqua	73
aquamarine	73
AreaChart	21
AreaChart2D	22
AreaChartOptions	22
areaProp	72
Arguments	8
Arrow head locations	71
ArrowStyle	22
automatic	83
average	84
Axis and grid locations	72
Axis indices	71
AxisLabelBackground	23
AxisLabelOptions	23
AxisLabelStyle	23
AxisLabelText	24
AxisLine	24
AxisMajorTickLabelBackground	24
AxisMajorTickLabelOptions	25
AxisMajorTickLabelStyle	25
AxisMajorTickLabelTexts	25
AxisMajorTicks	26
AxisMinorTickLabelBackground	26
AxisMinorTickLabelOptions	27
AxisMinorTickLabelStyle	27
AxisMinorTickLabelTexts	27
AxisMinorTicks	28
AxisOptions	28
azure	73
B	
back	72
Background	28
BackgroundPict	29
backslash	11
backward	85
BarChart	29

BarChart2D	30
BarChartOptions	30
baseline	91
begin	71, 82
beige	73
bisque	73
black	73, 85
Black&White Patterns	85
blanchedalmond	73
blue	73
blueviolet	73
bold	91
BorderStyle	30
both	78
bottom	91
bottomBar	90
bottomCenter	82, 84
bottomLeft	82, 84
bottomRight	82, 84
BoxPlot	31
BoxPlotOptions	31
brown	73
Bubble types	72
BubbleChart	31
BubbleChart2D	32
BubbleChartOptions	32
bullet	90
burlywood	73

C

cadetblue	73
CandlestickChart	32
CandlestickChart2D	33
carriage return	11
center	81-82, 91
centerCenter	82, 84
centeredBackward	85
centeredForward	85
centerLeft	82, 84
centerRight	82, 84
ChartBackground	33
ChartBackgroundPict	33
ChartData	8, 34
ChartDataLowerLimits	34
ChartDataOptions	35

ChartDataRead	35
ChartDataUpperLimits	35
ChartDataWrite	36
chartreuse	73
chocolate	73
circle	90
clipboard	9, 87
Clipping constants	72
CloseChart	36
CloseDrawing	36
CloseView	36
Color constants	73
Color Patterns	86
Comments	11
constant	79
coral	73
cornflowerblue	73
cornsilk	73
crimson	73
crop	87
cross	90
Curve fitting constants	77
CurveFitting	37
CurveFittingLineStyle	37
CurveFittingOptions	37
cyan	73
D	
D	95
darkblue	73
darkcyan	73
darkgoldenrod	74
darkGray	85
darkgray	74
darkgreen	74
darkkhaki	74
darkmagenta	74
darkolivegreen	74
darkorange	74
darkorchid	74
darkpurple	74
darkred	74
darksalmon	74
darkseagreen	74
darkslateblue	74

darkslategray	74
darkturquoise	74
darkviolet	74
darkyellow	74
Date orders	77
DateTime scaling constants	77
DateTimeOptions	38
day	77
Day specifiers	95
DD	95
DDY	95
deeppink	74
deepskyblue	74
default	71
Default format	92
diameterProp	72
diamond	90
diff	72
dimgray	74
dmy	77
dodgerblue	74
Double quotes	8
downTriangle	90
DropLineReferenceLine	38
DropLineReferencePoint	38
DropLineReferenceSeries	39
DropLineStyle	39
DX	95
Dx	95
DY	95
E	
e	84
edge	82
end	71, 82
Engineering notation	92
enlarge	87
Error bar directions	78
Error bar shapes	78
Error bar types	79
ErrorBarData	39
ErrorBars	40
ErrorBarStyle	40
ErrorBarStyle2D	41
exp	77

Explode constants	79
exponential	84
F	
file	9, 87
File flags	79
FillStyle	41
firebrick	74
Flag constants	80
Floating point format	92
floralwhite	74
forestgreen	74
forward	85
frequency	80
Frequency line constants	80
front	72
fuchsia	74
Functions	7
G	
gainsboro	74
GanttChart	41
ghostwhite	74
gold	74
goldenrod	74
Gradients	9
gray	74, 85
green	74
greenyellow	74
Grid shapes	80
GridFrame	42
GridLocation	42
H	
h	95
h12	95
hBar	90
hexadecimal	11
hh	95
hh12	95
high	81
High-Low chart constants	81
highLow	81
HighLowChart	42
HighLowChart2D	43

highLowClose	81
highLowCloseOpen	81
Histogram	43
HistogramOptions	44
HistogramRange	44
hollowDiamond	90
hollowDownTriangle	90
hollowSquare	90
hollowUpTriangle	90
honeydew	74
horizontal	71
Horizontal alignments	81
hotpink	74
hour	77
I	
in	91
indianred	74
indigo	74
Integer format	92
italic	91
ivory	74
J	
JPEG compression constants	81
jump	83
K	
khaki	74
L	
label	71
Label locations	82
LabelBackground	44
LabelOptions	45
LabelStyle	45
LabelTexts	46
lavender	74
lavenderblush	74
lawngreen	74
left	81
leftBar	90
Legend marker types	83
LegendBackground	46
LegendOptions	46

LegendStyle	47
LegendTexts	47
lemonchiffon	74
lightblue	75
lightcoral	75
lightcyan	75
lightgoldenrodyellow	75
lightGray	85
lightgray	75
lightgreen	75
lightpink	75
lightpurple	75
lightred	75
lightsalmon	75
lightseagreen	75
lightskyblue	75
lightslategray	75
lightsteelblue	75
lightyellow	75
lime	75
limegreen	75
line	83
Line shape constants	83
linear	77, 89
Linear gradients	9
LineChart	48
LineChart2D	48
linen	75
LineStyle	48
Location constants	84
log	77, 89
low	81

M

M	94
m	95
magenta	75
MajorGridLineColors	49
MajorGridLinePatterns	49
MajorGridLineWidths	49
MajorGridStripeColors	49
MajorGridStripePatterns	50
maroon	75
Mathematical constants	84
max	79, 81

mdy	77
median	84
mediumaquamarine	75
mediumblue	75
mediumorchid	75
mediumpurple	75
mediumseagreen	75
mediumslateblue	75
mediumspringgreen	75
mediumturquoise	75
mediumvioletred	75
midnightblue	75
min	79, 81
MinorGridLineColors	50
MinorGridLinePatterns	50
MinorGridLineWidths	51
MinorGridStripeColors	51
MinorGridStripePatterns	51
mintcream	75
minus	78
minute	77
minute10	77
minute15	77
minute20	77
minute30	77
minute5	77
mistyrose	75
MM	94
mm	95
Mo	94
moccasin	75
MON	94
Mon	94
MONTH	94
Month	94
month	77
Month specifiers	94
Mouse coordinates	11
Moving average alignments	85
Moving average calculations	84
MovingAverage	52
MovingAverageLineStyle	52
MovingAverageOptions	53

N

navajowhite	75
navy	75
newline	11
none	71-72, 77-79, 80, 83-84, 90
normal	81

O

off	80
ogive	80
oldlace	75
olive	75
olivedrab	75
on	80
OpenChart	53
OpenDrawing	54
OpenView	54
orange	75
orangered	75
orchid	75
out	82, 91
oval	71, 78, 80

P

palegoldenrod	75
palegreen	76
paleturquoise	76
palevioletred	76
papayawhip	76
Pattern constants	85
peachpuff	76
percent	79, 89
peru	76
pi	84
Picture adjustment constants	87
Picture source constants	87
PictureStyle	55
PieChart	55
PieChartAuxLines	55
PieChartCenterLabelBackground	56
PieChartCenterLabelStyle	56
PieChartCenterLabelText	57
PieChartExplodeDepths	57
PieChartExplodes	57
PieChartInnerLabelBackground	58

PieChartInnerLabelStyle	58
PieChartInnerLabelTexts	59
PieChartLabelOptions	59
pink	76
plain	91
Plane indices	89
plum	76
plus	78, 90
PolarChart	59
PolarChartOptions	60
poly	80, 83
pow	77
powderblue	76
proportional	71
purple	76
Q	
Q	94
quarter	77
Quarter specifier	94
quote	11
quotes	9
QX	94
Qx	94
R	
RadarChart	60
RadarChartOptions	60
Radial gradients	10
RBG components	73
rect	78, 80, 83
red	76
reduce	87
reduceOrEnlarge	87
replace	79
resource	9, 87
resource ID	9
reverseOgive	80
RGB colors	9
right	81
rightBar	90
rosybrown	76
royalblue	76
runningTotalsEdge	82
runningTotalsOut	82

S

s	95
saddlebrown	76
salmon	76
sandybrown	76
SaveAsBMPFile	61
SaveAsEMFFile	61
SaveAsGIFFile	61
SaveAsJPGFile	62
SaveAsPDFFile	62
SaveAsPICTFile	62
SaveAsPNGFile	63
SaveAsSVGFile	63
SaveAsTIFFFile	63
Scaling	64
Scaling constants	89
ScalingOptions	64
Scan directions	89
ScatterChart	64
ScatterChart2D	65
Scientific notation	92
seagreen	76
seashell	76
second	78
second10	78
second15	78
second20	78
second30	77
second5	78
sect	72
SendToClipboard	65
Series indices	90
SetDecimalPoint	65
SetThousandsSep	65
shadow	71
ShadowStyle	66
sienna	76
silver	76
skyblue	76
slateblue	76
slategray	76
smartBegin	82
smartCenter	82
smartEnd	82

smartOut	82
smooth	83
snow	76
springgreen	76
square	90
ss	95
stacked	71, 90
stdDev	79
stdError	79
steelblue	76
step	83
symbol	71, 83
Symbols	90
SymbolStyle	66

T

tab character	10
tan	76
teal	76
Text styles	91
thistle	76
throwError	79
Tick mark locations	91
tile	87
Time specifiers	95
TitleBackground	67
TitleOptions	67
TitleStyle	68
TitleSubStyle	68
TitleText	69
tomato	76
top	91
topBar	90
topCenter	82, 84
topLeft	82, 84
topRight	82, 84
totalsEdge	82
totalsOut	82
transparent	85
turquoise	76
Typographical quotes	8

U

underline	91
Unicode character	11

union	72
upTriangle	90
V	
valueList	79
vBar	90
Vertical alignments	91
violet	76
W	
W	94
WD1	94
WD2	94
Wd2	94
WD3	94
Wd3	94
week	77
Week specifiers	94
WEEKDAY	94
Weekday	94
wheat	76
white	76
whitesmoke	76
WWY	94
WY	94
X	
x	71
xmCH_GetErrorMessage	96
xmCH_GetMouse	11
xmCH_GetVersion	11
xmCH_SetLanguage	97
xyyy	89
xy	89
xyxy	89
xz	89
Y	
y	71
year	77
Year specifiers	94
yellow	76
yellowgreen	76
ymd	77

YY	94
YYYY	94
yz	89
Z	
z	71