

MBS ChartDirector Plugin Documentation

Christian Schmitz

March 10, 2024

0.1 Introduction

This is the PDF version of the documentation for the Xojo Plug-in from Monkeybread Software Germany.
Plugin part: MBS ChartDirector Plugin

0.2 Content

- 1 List of all topics 3
- 2 List of all classes 57
- 3 All items in this plugin 61
- 4 List of Questions in the FAQ 845
- 5 The FAQ 855

Chapter 1

List of Topics

• 3 ChartDirector	61
– 3.1.1 class CDAngularAxisMBS	61
* 3.1.3 addLabel(pos as Double, label as string)	61
* 3.1.4 addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color)	62
* 3.1.5 addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1)	62
* 3.1.6 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as color, edgeColor as color)	63
* 3.1.7 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as Integer, edgeColor as Integer)	63
* 3.1.8 Constructor	64
* 3.1.9 getAxisImageMap(noOfSegments as Integer, mapWidth as Integer, url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string	64
* 3.1.10 getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string	64
* 3.1.11 setLabelGap(d as Integer)	65
* 3.1.12 setLabels(labels() as Double, formatString as string = "") as CDTextBoxMBS	66
* 3.1.13 setLabels(labels() as string) as CDTextBoxMBS	67
* 3.1.14 setLabelStyle(font as string = "bold", fontsize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS	67
* 3.1.15 setLinearScale(lowerLimit as Double, upperLimit as Double, labels() as string)	68
* 3.1.16 setLinearScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)	68
– 3.2.1 class CDAngularMeterMBS	70
* 3.2.3 addGlare	71
* 3.2.4 addGlare(radius as Double, span as Double = 135, rotate as Double = 0.0)	71

- * 3.2.5 addGlare(radius as Double, span as Double, rotate as Double, glareRadius as Double, intensity as Double = 0.13) 73
- * 3.2.6 addPointer(value as Double, fillColor as color, edgeColor as color = &cFFFFFFFF, pointerType as Integer = 6) as CDMeterPointerMBS 73
- * 3.2.7 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS 74
- * 3.2.8 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS 76
- * 3.2.9 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS 77
- * 3.2.10 addPointer(value as Double, fillColor as Integer, edgeColor as Integer = -1, pointerType as Integer = 6) as CDMeterPointerMBS 78
- * 3.2.11 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS 79
- * 3.2.12 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS 80
- * 3.2.13 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS 81
- * 3.2.14 addRing(startRadius as Integer, endRadius as Integer, fillColor as color, edgeColor as color) 82
- * 3.2.15 addRing(startRadius as Integer, endRadius as Integer, fillColor as Integer, edgeColor as Integer = -1) 83
- * 3.2.16 addRingSector(startRadius as Integer, endRadius as Integer, a1 as Double, a2 as Double, fillColor as color, edgeColor as color) 83
- * 3.2.17 addRingSector(startRadius as Integer, endRadius as Integer, a1 as Double, a2 as Double, fillColor as Integer, edgeColor as Integer = -1) 83
- * 3.2.18 addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer = 0, edgeColor as color = &cFFFFFFFF, scaleRadius as Integer = -2147483647) 84
- * 3.2.19 addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer, edgeColor as color, scaleRadius as Integer, startAngle as Double, endAngle as Double) 85
- * 3.2.20 addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer = 0, edgeColor as Integer = -1, scaleRadius as Integer = -2147483647) 86
- * 3.2.21 addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer, edgeColor as Integer, scaleRadius as Integer, startAngle as Double, endAngle as Double) 87
- * 3.2.22 addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color) 88
- * 3.2.23 addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1) 89
- * 3.2.24 addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as color, edgeColor as color) 89
- * 3.2.25 addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as Integer, edgeColor as Integer = -1) 90

* 3.2.26	Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)	90
* 3.2.27	Constructor(width as Integer, height as Integer, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)	90
* 3.2.28	relativeLinearGradient(gradient() as Double, angle as Double = 0.0, radius as Double = -1.0) as Integer	91
* 3.2.29	relativeRadialGradient(gradient() as Double, radius as Double = -1.0) as Integer	92
* 3.2.30	setCap(radius as Integer, fillColor as color, edgeColor as color)	93
* 3.2.31	setCap(radius as Integer, fillColor as Integer, edgeColor as Integer = &hfff0001)	93
* 3.2.32	setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888)	94
* 3.2.33	setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double)	95
* 3.2.34	setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double)	96
* 3.2.35	setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)	97
* 3.2.36	setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)	98
* 3.2.37	setCap2(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888)	100
* 3.2.38	setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)	101
* 3.2.39	setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double)	102
* 3.2.40	setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)	103
* 3.2.41	setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)	104
* 3.2.42	setMeter(cx as Integer, cy as Integer, radius as Integer, startAngle as Double, endAngle as Double)	106
– 3.3.1	class CDAreaLayerMBS	107
* 3.3.3	setGapColor(fillColor as color)	108
* 3.3.4	setGapColor(fillColor as Integer)	108
* 3.3.5	setMinLabelSize(s as Integer)	109
– 3.4.1	class CDArrayMBS	110
* 3.4.3	abs	110
* 3.4.4	acc	110
* 3.4.5	addArray(value as CDArrayMBS)	110
* 3.4.6	addArray(value() as Double)	111
* 3.4.7	addValue(value as Double)	111

* 3.4.8 aggregate(srcArray() as Double, aggregateMethod as Integer, param as Double = 50.0) as CCharArrayMBS	112
* 3.4.9 aggregateValues(srcArray() as Double, aggregateMethod as Integer, param as Double = 50.0) as Double()	113
* 3.4.10 avg as Double	114
* 3.4.11 Constructor	114
* 3.4.12 Constructor(a as CCharArrayMBS)	115
* 3.4.13 Constructor(data() as Double)	115
* 3.4.14 count as Integer	115
* 3.4.15 delta(offset as Integer = 1)	115
* 3.4.16 divArray(value as CCharArrayMBS)	116
* 3.4.17 divArray(value() as Double)	116
* 3.4.18 divValue(value as Double)	117
* 3.4.19 expAvg(smoothingFactor as Double)	117
* 3.4.20 financeDiv(values() as Double, zeroByZeroValue as Double)	118
* 3.4.21 getvalue(index as Integer) as Double	118
* 3.4.22 insert(value as Double, len as Integer, insertPoint as Integer = -1)	118
* 3.4.23 insert(value() as Double, insertPoint as Integer = -1)	119
* 3.4.24 lowess(smoothness as Double = 0.25, iteration as Integer = 0)	119
* 3.4.25 lowess(values() as Double, smoothness as Double = 0.25, iteration as Integer = 0)	121
* 3.4.26 max as Double	122
* 3.4.27 maxIndex as Integer	122
* 3.4.28 med as Double	122
* 3.4.29 min as Double	122
* 3.4.30 minIndex as Integer	123
* 3.4.31 movAvg(interval as Integer)	123
* 3.4.32 movCorr(interval as Integer, value() as Double)	123
* 3.4.33 movMax(interval as Integer)	124
* 3.4.34 movMed(interval as Integer)	124
* 3.4.35 movMin(interval as Integer)	124
* 3.4.36 movPercentile(interval as Integer, percentile as Double)	125
* 3.4.37 movStdDev(interval as Integer)	125
* 3.4.38 mulArray(value as CCharArrayMBS)	125
* 3.4.39 mulArray(value() as Double)	126
* 3.4.40 mulValue(value as Double)	126
* 3.4.41 percentile(p as Double) as double	127
* 3.4.42 rate(offset as Integer = 1)	127
* 3.4.43 replace(a as Double, b as Double)	128
* 3.4.44 result as memoryblock	128
* 3.4.45 selectEQZ	128
* 3.4.46 selectEQZ(decisionArray() as Double, fillValue as Double = 0)	128
* 3.4.47 selectGEZ	129

* 3.4.48	selectGEZ(decisionArray() as Double, fillValue as Double = 0)	129
* 3.4.49	selectGTZ	130
* 3.4.50	selectGTZ(decisionArray() as Double, fillValue as Double = 0)	130
* 3.4.51	selectLEZ	131
* 3.4.52	selectLEZ(decisionArray() as Double, fillValue as Double = 0)	131
* 3.4.53	selectLTZ	132
* 3.4.54	selectLTZ(decisionArray() as Double, fillValue as Double = 0)	132
* 3.4.55	selectNEZ	133
* 3.4.56	selectNEZ(decisionArray() as Double, fillValue as Double = 0)	133
* 3.4.57	selectRegularSpacing(majorTickStep as Integer, minorTickStep as Integer = 0, initialMargin as Integer = 0)	133
* 3.4.58	selectStartOfDay(majorTickStep as Integer = 1, initialMargin as Double = 10800.0)	134
* 3.4.59	selectStartOfHour(majorTickStep as Integer = 1, initialMargin as Double = 300.0)	135
* 3.4.60	selectStartOfMinute(majorTickStep as Integer = 1, initialMargin as Double = 5.0)	136
* 3.4.61	selectStartOfMonth(majorTickStep as Integer = 1, initialMargin as Double = 432000.0)	137
* 3.4.62	selectStartOfSecond(majorTickStep as Integer = 1, initialMargin as Double = 0.1)	138
* 3.4.63	selectStartOfWeek(majorTickStep as Integer = 1, initialMargin as Double = 172800.0)	139
* 3.4.64	selectStartOfYear(majorTickStep as Integer = 1, initialMargin as Double = 5184000.0)	140
* 3.4.65	shift(offset as Integer = 1)	141
* 3.4.66	shift(offset as Integer, fillValue as Double)	141
* 3.4.67	stdDev as Double	142
* 3.4.68	subArray(value as CDArrayMBS)	142
* 3.4.69	subArray(value() as Double)	142
* 3.4.70	subValue(value as Double)	143
* 3.4.71	sum as Double	144
* 3.4.72	trim(startIndex as Integer = 0, len as Integer = -1)	144
* 3.4.73	Values as Double()	144
– 3.5.1	class CDAxisMBS	145
* 3.5.3	addLabel(pos as Double, label as string)	145
* 3.5.4	addMark(value as Double, lineColor as color, text as string = "", font as string = "", fontsize as Double = 8) as CDMarkMBS	145
* 3.5.5	addMark(value as Double, lineColor as Integer, text as string = "", font as string = "", fontsize as Double = 8) as CDMarkMBS	146
* 3.5.6	addZone(startValue as Double, endValue as Double, colorvalue as color)	146
* 3.5.7	addZone(startValue as Double, endValue as Double, colorvalue as Integer)	147
* 3.5.8	Constructor	147
* 3.5.9	copyAxis(axis as CDAxisMBS)	147
* 3.5.10	getAlignment as Integer	148

* 3.5.11 getAxisImageMap(noOfSegments as Integer, mapWidth as Integer, url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string	148
* 3.5.12 getCoor(value as Double) as Integer	148
* 3.5.13 getFormattedLabel(v as Double, options as string = "") as string	148
* 3.5.14 getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string	149
* 3.5.15 getLabel(i as Double) as string	150
* 3.5.16 getLabelTable as CDMLTableMBS	151
* 3.5.17 getMaxValue as Double	151
* 3.5.18 getMinValue as Double	151
* 3.5.19 getThickness as Integer	152
* 3.5.20 getTicks as CDArrayMBS	152
* 3.5.21 getX as Integer	153
* 3.5.22 getY as Integer	153
* 3.5.23 makeLabelTable as CDMLTableMBS	153
* 3.5.24 setAngle(angle as Double)	154
* 3.5.25 setAutoScale(topExtension as Double = 0.1, bottomExtension as Double = 0.1, zeroAffinity as Double = 0.8)	154
* 3.5.26 setColors(axisColor as color, labelColor as color, titleColor as color, tickColor as color)	155
* 3.5.27 setColors(axisColor as Integer, labelColor as Integer = &hfff0002, titleColor as Integer = -1, tickColor as Integer = -1)	156
* 3.5.28 setDateScale(formatString as string = "")	156
* 3.5.29 setDateScale(lowerLimit as Double, upperLimit as Double, labels() as string)	157
* 3.5.30 setDateScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)	157
* 3.5.31 setFormatCondition(condition as string, operand as Double = 0)	158
* 3.5.32 setIndent(indent as boolean)	159
* 3.5.33 setLabelAlignment(alignment as Integer, minLabelSpace as Integer = 3)	160
* 3.5.34 setLabelFormat(formatString as string)	160
* 3.5.35 setLabelGap(d as Integer)	161
* 3.5.36 setLabelOffset(offset as Double)	161
* 3.5.37 setLabels(labels() as Double, formatString as string = "") as CDTextBoxMBS	162
* 3.5.38 setLabels(labels() as string) as CDTextBoxMBS	163
* 3.5.39 setLabelStep(majorTickStep as Integer, minorTickStep as Integer = 0, majorTickOffset as Integer = 0, minorTickOffset as Integer = -2147483647)	164
* 3.5.40 setLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS	164
* 3.5.41 setLabelStyle(font as string, fontsize as Double, fontcolor as color, fontAngle as Double = 0) as CDTextBoxMBS	165
* 3.5.42 setLength(length as Integer)	165
* 3.5.43 setLinearScale(formatString as string = "")	165

* 3.5.44	setLinearScale(lowerLimit as Double, upperLimit as Double, labels() as string)	166
* 3.5.45	setLinearScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)	166
* 3.5.46	setLogScale(formatString as string = "")	167
* 3.5.47	setLogScale(lowerLimit as Double, upperLimit as Double, labels() as string)	167
* 3.5.48	setLogScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)	168
* 3.5.49	setMargin(topMargin as Integer, bottomMargin as Integer = 0)	168
* 3.5.50	setMinTickInc(value as Double)	169
* 3.5.51	setMultiFormat(filter as Integer, format as string, labelSpan as Integer = 1, promoteFirst as boolean=true)	169
* 3.5.52	setMultiFormat(filter1 as Integer, format1 as string, filter2 as Integer, format2 as string, labelSpan as Integer = 1, promoteFirst as boolean=true)	170
* 3.5.53	setOffset(x as Integer, y as Integer)	171
* 3.5.54	setReverse(value as boolean=true)	172
* 3.5.55	setRounding(roundMin as boolean, roundMax as boolean)	172
* 3.5.56	setTickColor(majorTickColor as color, minorTickColor as color)	172
* 3.5.57	setTickColor(majorTickColor as Integer, minorTickColor as Integer = -1)	173
* 3.5.58	setTickDensity(majorTickSpacing as Integer, minorTickSpacing as Integer = -1)	173
* 3.5.59	setTickLength(majorTickLen as Integer)	173
* 3.5.60	setTickLength(majorTickLen as Integer, minorTickLen as Integer)	174
* 3.5.61	setTickOffset(offset as Double)	174
* 3.5.62	setTickWidth(majorTickWidth as Integer, minorTickWidth as Integer = -1)	175
* 3.5.63	setTitle(text as string, font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002) as CDTextBoxMBS	175
* 3.5.64	setTitle(text as string, font as string, fontsize as Double, fontcolor as color) as CD-TextBoxMBS	175
* 3.5.65	setTitlePos(alignment as Integer, titleGap as Integer = 3)	176
* 3.5.66	setWidth(width as Integer)	176
* 3.5.67	syncAxis(axis as CDAxisMBS, slope as Double = 1.0, intercept as Double = 0.0)	176
* 3.5.68	syncScale(axis as CDAxisMBS = nil, slope as double = 1.0, intercept as double = 0.0)	177
- 3.6.1	class CDBarLayerMBS	178
* 3.6.3	setBarGap(barGap as Double)	178
* 3.6.4	setBarGap(barGap as Double, subBarGap as Double)	179
* 3.6.5	setBarShape(shape as Integer, dataGroup as Integer = -1, dataItem as Integer = -1)	180
* 3.6.6	setBarShape(shape() as Integer, dataGroup as Integer = -1, dataItem as Integer = -1)	180
* 3.6.7	setBarWidth(barWidth as Integer, subBarWidth as Integer = -1)	181
* 3.6.8	setIconSize(height as Integer, width as Integer = -1)	181
* 3.6.9	setMinImageMapSize(s as Integer)	181
* 3.6.10	setMinLabelSize(s as Integer)	182

* 3.6.11	setOverlapRatio(overlapRatio as Double, firstOnTop as boolean=true)	182
* 3.6.12	setRoundedCorners	182
* 3.6.13	setRoundedCorners(r1 as Integer, r2 as Integer = -2147483647, r3 as Integer = -2147483647, r4 as Integer = -2147483647)	184
– 3.7.1	class CDBaseBoxLayerMBS	185
* 3.7.3	setDataGap(gap as Double)	185
* 3.7.4	setDataWidth(width as Integer)	185
* 3.7.5	setMinImageMapSize(size as Integer)	186
* 3.7.6	setRoundedCorners(r1 as Integer)	186
* 3.7.7	setRoundedCorners(r1 as Integer, r2 as Integer, r3 as Integer = -2147483647, r4 as Integer = -2147483647)	187
– 3.8.1	class CDBaseChartMBS	189
* 3.8.3	addExtraField(numbers() as Double)	190
* 3.8.4	addExtraField(paths() as folderitem)	190
* 3.8.5	addExtraField(texts() as string)	191
* 3.8.6	addLegend(x as Integer, y as Integer, noOfCols as Integer, font as string = "", fontsize as Double = 10) as CDLegendBoxMBS	191
* 3.8.7	addLegend(x as Integer, y as Integer, vertical as boolean=true, font as string = "", fontsize as Double = 10) as CDLegendBoxMBS	192
* 3.8.8	addLine(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, colorvalue as color, lineWidth as Integer = 1) as CDLineMBS	193
* 3.8.9	addLine(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, colorvalue as Integer = &hfff0001, lineWidth as Integer = 1) as CDLineMBS	193
* 3.8.10	addTable(x as Integer, y as Integer, alignment as Integer, col as Integer, row as Integer) as CDMLTableMBS	194
* 3.8.11	addText(x as Integer, y as Integer, text as string, font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002, alignment as Integer = 7, angle as Double = 0, vertical as boolean=false) as CDTextBoxMBS	194
* 3.8.12	addText(x as Integer, y as Integer, text as string, font as string, fontsize as Double, fontcolor as color, alignment as Integer = 7, angle as Double = 0, vertical as boolean=false) as CDTextBoxMBS	195
* 3.8.13	addTitle(alignment as Integer, text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS	195
* 3.8.14	addTitle(alignment as Integer, text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS	196
* 3.8.15	addTitle(text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS	197
* 3.8.16	addTitle(text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS	197
* 3.8.17	adjustBrightness(ColorValue as color, brightness as Double) as Integer	198
* 3.8.18	adjustBrightness(ColorValue as Integer, brightness as Double) as Integer	198

* 3.8.19 AllPassFilter as Integer	199
* 3.8.20 ArrowShape(angle as Double = 0.0, widthRatio as Double = 1, stemWidthRatio as Double = 0.5, stemLengthRatio as Double = 0.5) as Integer	199
* 3.8.21 barLighting(startBrightness as Double = 0.75, endBrightness as Double = 1.5) as Integer	199
* 3.8.22 blueMetalGradient as Integer()	200
* 3.8.23 brushedGoldColor(texture as Integer = 2, angle as Integer = 90) as Integer	200
* 3.8.24 brushedMetalColor(c as Integer, texture as Integer = 2, angle as Integer = 90) as Integer	201
* 3.8.25 brushedSilverColor(texture as Integer = 2, angle as Integer = 90) as Integer	202
* 3.8.26 bSearch(values() as Double, value as Double) as Double	202
* 3.8.27 chartTime(t as Integer) as Double	203
* 3.8.28 chartTime(year as Integer, month as Integer, day as Integer, hour as Integer = 0, minute as Integer = 0, second as Integer = 0) as Double	203
* 3.8.29 ClearTypeColor(gamma as Double = 0) as Integer	204
* 3.8.30 ClearTypeMono(gamma as Double = 0) as Integer	205
* 3.8.31 ColorToInteger(c as color, alpha as Integer = 0) as Integer	205
* 3.8.32 Constructor	205
* 3.8.33 Cross2Shape(width as Double = 0.5) as Integer	205
* 3.8.34 CrossShape(width as Double = 0.5) as Integer	206
* 3.8.35 cylinderEffect(orientation as Integer = 5, ambientIntensity as Double = 0.5, diffuseIntensity as Double = 0.5, specularIntensity as Double = 0.75, shininess as Integer = 8) as Integer	206
* 3.8.36 dashLineColor(colorvalue as color, patternCode as Integer = &h0505) as Integer	207
* 3.8.37 dashLineColor(colorvalue as Integer, patternCode as Integer = &h0505) as Integer	207
* 3.8.38 defaultPalette as Integer()	208
* 3.8.39 Destructor	208
* 3.8.40 enableVectorOutput	208
* 3.8.41 flatBorder(thickness as Integer) as Integer	208
* 3.8.42 formatValue(value as Double, formatstring as string) as string	209
* 3.8.43 getAbsOffsetX as Integer	209
* 3.8.44 getAbsOffsetY as Integer	210
* 3.8.45 getChartMetrics as string	210
* 3.8.46 getChartWeekDay(t as Double) as Integer	210
* 3.8.47 getChartYMD(t as Double) as Integer	210
* 3.8.48 getColor(index as Integer) as Integer	211
* 3.8.49 getCopyright as string	211
* 3.8.50 getDescription as string	211
* 3.8.51 getDrawArea as CDDrawAreaMBS	212
* 3.8.52 getHeight as Integer	212
* 3.8.53 getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string	212
* 3.8.54 getLegend as CDLegendBoxMBS	214

* 3.8.55	GetPath(path as folderitem) as string	215
* 3.8.56	getVersion as Integer	215
* 3.8.57	getWidth as Integer	215
* 3.8.58	glassEffect(glareSize as Integer = 3, glareDirection as Integer = 8, raisedEffect as Integer = 5) as Integer	215
* 3.8.59	goldColor(angle as Integer = 90) as Integer	216
* 3.8.60	goldGradient as Integer()	217
* 3.8.61	gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer	217
* 3.8.62	gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer	217
* 3.8.63	gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer	218
* 3.8.64	gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer	219
* 3.8.65	greenMetalGradient as Integer()	219
* 3.8.66	halfColor(c as Integer) as Integer	219
* 3.8.67	initDynamicLayer as CDDrawAreaMBS	220
* 3.8.68	kDataBound as Double	220
* 3.8.69	kLinearTick as Double	221
* 3.8.70	kLogTick as Double	221
* 3.8.71	kMicroTickOnly as Double	221
* 3.8.72	kMinorTickOnly as Double	221
* 3.8.73	kNoValue as Double	221
* 3.8.74	kTickInc as Double	222
* 3.8.75	kTouchBar as Double	222
* 3.8.76	layout	222
* 3.8.77	layoutLegend as CDLegendBoxMBS	222
* 3.8.78	linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer	223
* 3.8.79	linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer	223
* 3.8.80	linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer	224
* 3.8.81	linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer	225
* 3.8.82	makeChart as CDDrawAreaMBS	225
* 3.8.83	makeChart(format as Integer) as string	226
* 3.8.84	makeChart(path as folderitem) as boolean	227
* 3.8.85	makeChartPicture as picture	227
* 3.8.86	metalColor(c as Integer, angle as Integer = 90) as Integer	227
* 3.8.87	NonePassFilter as Integer	228

- * 3.8.88 patternColor(colorvalues() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer 228
- * 3.8.89 patternColor(colorvalues() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer 228
- * 3.8.90 patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer 229
- * 3.8.91 patternColor(path as string, startX as integer = 0, startY as integer = 0) as integer 230
- * 3.8.92 patternColor(pic as picture, startX as Integer = 0, startY as Integer = 0) as Integer 230
- * 3.8.93 phongLighting(ambientIntensity as Double = 0.5, diffuseIntensity as Double = 0.5, specularIntensity as Double = 0.75, shininess as Integer = 8) as Integer 231
- * 3.8.94 Polygon2Shape(slide as Integer) as Integer 231
- * 3.8.95 PolygonShape(slide as Integer) as Integer 232
- * 3.8.96 PolynomialRegression(n as Integer) as Integer 232
- * 3.8.97 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, data() as Integer, periodic as boolean=false) as Integer 232
- * 3.8.98 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer 233
- * 3.8.99 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer 234
- * 3.8.100 redMetalGradient as Integer() 234
- * 3.8.101 RegularSpacingFilter(labelStep as Integer = 1, initialMargin as Integer = 0) as Integer 234
- * 3.8.102 removeDynamicLayer 235
- * 3.8.103 RGB(r as Integer, g as Integer, b as Integer, a as Integer = 0) as Integer 236
- * 3.8.104 SelectItemFilter(item as Integer) as Integer 236
- * 3.8.105 setAMPMP(am as string, pm as string) 236
- * 3.8.106 setAntiAlias(shapeAntiAlias as Boolean, textAntiAlias as Integer) 236
- * 3.8.107 setBackground(colorvalue as color, edgeColor as color, raisedEffect as Integer = 0) 237
- * 3.8.108 setBackground(colorvalue as Integer, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0) 237
- * 3.8.109 setBgImage(img as string, align as Integer = 5) 238
- * 3.8.110 setBorder(colorvalue as color) 238
- * 3.8.111 setBorder(colorvalue as Integer) 238
- * 3.8.112 setColor(paletteEntry as Integer, colorvalue as color) 239
- * 3.8.113 setColor(paletteEntry as Integer, colorvalue as Integer) 239
- * 3.8.114 setColors(numbers() as color) 239
- * 3.8.115 setColors(numbers() as Integer) 239
- * 3.8.116 setColors(paletteEntry as Integer, numbers() as color) 240
- * 3.8.117 setColors(paletteEntry as Integer, numbers() as Integer) 240
- * 3.8.118 setDefaultColors(paletteEntry as Integer = 0) 241

* 3.8.119 setDefaultFonts(normal as string, bold as string, italic as string, boldItalic as string)	241
* 3.8.120 setDropShadow(ColorValue as color, OffsetX as Integer = 5, OffsetY as Integer = &h7ffffff, blurRadius as Integer = 5)	241
* 3.8.121 setDropShadow(ColorValue as Integer = &hAAAAAA, OffsetX as Integer = 5, OffsetY as Integer = &h7ffffff, blurRadius as Integer = 5)	242
* 3.8.122 SetFontSearchPath(path as folderitem)	242
* 3.8.123 SetFontSearchPath(path as string)	243
* 3.8.124 setFontTable(index as Integer, font as string)	244
* 3.8.125 setLicenseCode(n as string, enddate as Integer, v1 as Integer, v2 as Integer)	244
* 3.8.126 setMonthNames(names() as string)	244
* 3.8.127 setNumberFormat(thousandSeparator as string = "textasciitilde ", decimalPointChar as string = "'", signChar as string = "-")	245
* 3.8.128 setOutputOptions(options as string)	245
* 3.8.129 setResource(id as string, data as MemoryBlock)	246
* 3.8.130 setResource(id as string, data as string)	247
* 3.8.131 setResource(id as string, DataArea as CDDrawAreaMBS)	248
* 3.8.132 setResourceGlobal(id as string, data as MemoryBlock)	249
* 3.8.133 setResourceGlobal(id as string, data as string)	249
* 3.8.134 setResourceGlobal(id as string, DrawArea as CDDrawAreaMBS)	250
* 3.8.135 setRoundedFrame(extColor as color, r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)	250
* 3.8.136 setRoundedFrame(extColor as Integer = &hFFFFFF, r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)	250
* 3.8.137 setSearchPath(path as string)	251
* 3.8.138 setSize(width as Integer, height as Integer)	251
* 3.8.139 setThickFrame(thickness as Integer, frameColor as Integer = -1, outerEdgeColor as Integer = -1, innerEdgeColor as Integer = -1)	252
* 3.8.140 setTransparentColor(c as color)	252
* 3.8.141 setTransparentColor(c as Integer)	253
* 3.8.142 setTransparentColors(paletteEntry as Integer = 0)	254
* 3.8.143 setWallpaper(img as folderitem)	254
* 3.8.144 setWeekDayNames(names() as string)	254
* 3.8.145 setWhiteOnBlackColors(paletteEntry as Integer = 0)	254
* 3.8.146 silverColor(angle as Integer = 90) as Integer	254
* 3.8.147 silverGradient as Integer()	255
* 3.8.148 softLighting(direction as Integer = 8, raisedEffect as Integer = 4) as Integer	255
* 3.8.149 StarShape(slide as Integer) as Integer	256
* 3.8.150 StartOfDayFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer	256
* 3.8.151 StartOfHourFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer	257

* 3.8.152 StartOfMinuteFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer	258
* 3.8.153 StartOfMonthFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer	258
* 3.8.154 StartOfSecondFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer	259
* 3.8.155 StartOfWeekFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer	260
* 3.8.156 StartOfYearFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer	260
* 3.8.157 testFont(font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, byref buffer as string) as boolean	261
* 3.8.158 transparentPalette as Integer()	262
* 3.8.159 whiteOnBlackPalette as Integer()	262
* 3.8.160 xySize(x as Integer, y as Integer) as Integer	262
* 3.8.162 Handle as Integer	263
* 3.8.163 Resources as Dictionary	263
* 3.8.164 ScaleFactor as Double	263
– 3.9.1 class CDBaseMeterMBS	279
* 3.9.3 addColorScale(colorStops() as Double, startPos as Integer = -2147483647, startWidth as Integer = -2147483647, endPos as Integer = -2147483647, endWidth as Integer = -2147483647, edgeColor as Integer = -1)	279
* 3.9.4 addColorScale(colorStops() as Double, startPos as Integer, startWidth as Integer, endPos as Integer, endWidth as Integer, edgeColor as color)	280
* 3.9.5 addLabel(v as Double, label as string)	281
* 3.9.6 addPointer(value as Double, fillColor as color, edgeColor as color) as CDMeterPointerMBS	282
* 3.9.7 addPointer(value as Double, fillColor as Integer = &hfff0001, edgeColor as Integer = -1) as CDMeterPointerMBS	282
* 3.9.8 getCoor(v as Double) as Integer	282
* 3.9.9 getLabel(v as Double) as string	283
* 3.9.10 getTicks as CDArrayMBS	283
* 3.9.11 setLabelFormat(mainLabelFormat as string)	283
* 3.9.12 setLabelPos(labelInside as boolean, labelOffset as Integer = 0)	284
* 3.9.13 setLabelStyle(font as string = "", fontsize as Double = -1, fontcolor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS	285
* 3.9.14 setLabelStyle(font as string, fontsize as Double, fontcolor as color, fontAngle as Double = 0) as CDTextBoxMBS	285
* 3.9.15 setLineWidth(axisWidth as Integer, majorTickWidth as Integer = 1, minorTickWidth as Integer = 1, microTickWidth as Integer = 1)	286
* 3.9.16 setMeterColors(axisColor as color, labelColor as color, tickColor as color)	286
* 3.9.17 setMeterColors(axisColor as Integer, labelColor as Integer = -1, tickColor as Integer = -1)	286

* 3.9.18	setScale(lowerLimit as Double, upperLimit as Double, labels() as Double, formatstring as string = "")	287
* 3.9.19	setScale(lowerLimit as Double, upperLimit as Double, labels() as string)	287
* 3.9.20	setScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0, microTickInc as Double = 0)	288
* 3.9.21	setTickLength(majorLen as Integer)	288
* 3.9.22	setTickLength(majorLen as Integer, minorLen as Integer)	289
* 3.9.23	setTickLength(majorLen as Integer, minorLen as Integer, microLen as Integer)	289
– 3.10.1	class CDBoxMBS	291
* 3.10.3	Constructor	291
* 3.10.4	getHeight as Integer	291
* 3.10.5	getImageCoor(OffsetX as Integer = 0, OffsetY as Integer = 0) as string	291
* 3.10.6	getLeftX as Integer	292
* 3.10.7	getTopY as Integer	292
* 3.10.8	getWidth as Integer	293
* 3.10.9	setBackground(colorvalue as color, edgeColor as color, raisedEffect as Integer = 0)	293
* 3.10.10	setBackground(colorvalue as Integer, edgeColor as Integer = -1, raisedEffect as Integer = 0)	294
* 3.10.11	setPos(x as Integer, y as Integer)	294
* 3.10.12	setRoundedCorners(r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)	294
* 3.10.13	setSize(w as Integer, h as Integer)	295
– 3.11.1	class CDBoxWhiskerLayerMBS	296
* 3.11.3	setBoxColor(item as Integer, boxColor as color)	296
* 3.11.4	setBoxColor(item as Integer, boxColor as Integer)	296
* 3.11.5	setBoxColors(colors() as color)	296
* 3.11.6	setBoxColors(colors() as color, names() as string)	297
* 3.11.7	setBoxColors(colors() as Integer)	297
* 3.11.8	setBoxColors(colors() as Integer, names() as string)	298
* 3.11.9	setWhiskerBrightness(whiskerBrightness as Double)	298
– 3.12.1	class CDCandleStickLayerMBS	299
* 3.12.3	setColors(upFillColor as color, upLineColor as color, downFillColor as color, downLineColor as color)	299
* 3.12.4	setColors(upFillColor as Integer, upLineColor as Integer, downFillColor as Integer, downLineColor as Integer)	299
* 3.12.5	setExtraColors(upDownFillColor as color, upDownLineColor as color, downDownFillColor as color, downDownLineColor as color, leadValue as Double = -1.7E308)	300
* 3.12.6	setExtraColors(upDownFillColor as Integer, upDownLineColor as Integer, downDownFillColor as Integer, downDownLineColor as Integer, leadValue as Double = -1.7E308)	301
– 3.13.1	class CDColorAxisMBS	302
* 3.13.3	getBoxHeight as Integer	303
* 3.13.4	getBoxWidth as Integer	303

* 3.13.5 getColor(z as Double) as Integer	303
* 3.13.6 getColorScale as Double()	304
* 3.13.7 setAxisBorder(edgeColor as color, raisedEffect as Integer = 0)	304
* 3.13.8 setAxisBorder(edgeColor as Integer, raisedEffect as Integer = 0)	304
* 3.13.9 setAxisPos(x as Integer, y as Integer, alignment as Integer)	305
* 3.13.10 setBoundingBox(fillColor as color, edgeColor as color, raisedEffect as Integer = 0)	305
* 3.13.11 setBoundingBox(fillColor as Integer, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)	305
* 3.13.12 setBoxMargin(leftMargin as Integer, rightMargin as Integer, topMargin as Integer, bottomMargin as Integer)	305
* 3.13.13 setBoxMargin(m as Integer)	306
* 3.13.14 setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as color, overflowColor as color)	306
* 3.13.15 setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as Integer, overflowColor as Integer = -1)	307
* 3.13.16 setColorGradient(isContinuous as boolean, Colors() as Integer, underflowColor as Integer = -1, overflowColor as Integer = -1)	308
* 3.13.17 setColorGradient(isContinuous as boolean=true)	309
* 3.13.18 setColorScale(colorStops() as Double, underflowColor as Integer = -1, overflowColor as Integer = -1)	309
* 3.13.19 setCompactAxis(b as boolean=true)	310
* 3.13.20 setLevels(maxLevels as Integer)	311
* 3.13.21 setRoundedCorners(r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)	311
– 3.14.1 class CDCContourLayerMBS	312
* 3.14.3 addCustomContour(z as double, contourColor as Integer, contourWidth as Integer, contourLabel as string, font as string = "normal", fontsize as double = 12, fontColor as integer = &hfff0002) as CDMarkMBS	313
* 3.14.4 colorAxis as CDCColorAxisMBS	313
* 3.14.5 getCrossSection(x0 as Integer, y0 as Integer, x1 as Integer, y1 as Integer) as Double()	313
* 3.14.6 getZAtPixel(x as Integer, y as Integer) as Double	314
* 3.14.7 getZAtValue(x as double, y as double) as Double	314
* 3.14.8 setColorAxis(x as Integer, y as Integer, alignment as Integer, length as Integer, orientation as Integer) as CDCColorAxisMBS	315
* 3.14.9 setContourColor(contourColor as color, minorContourColor as color)	315
* 3.14.10 setContourColor(contourColor as Integer, minorContourColor as Integer = -1)	315
* 3.14.11 setContourLabelFormat(formatString as string)	316
* 3.14.12 setContourLabelSpacing(labelSpacing as Integer, minContourLen as Integer)	316
* 3.14.13 setContourLabelStyle(font as string = "normal", fontsize as double = 12, fontColor as integer = &hfff0002) as CDTextBoxMBS	316
* 3.14.14 setContourWidth(contourWidth as Integer, minorContourWidth as Integer = -1)	317

* 3.14.15	setExactContour(contour as boolean = true)	317
* 3.14.16	setExactContour(contour as boolean, markContour as boolean)	318
* 3.14.17	setSmoothInterpolation(b as boolean)	319
* 3.14.18	setZBounds(minZ as Double, maxZ as Double)	319
* 3.14.19	setZData(zData() as Double)	320
– 3.15.1	class CDDataAcceleratorMBS	321
* 3.15.3	addDataSeries(id as string, yData as CDDArrayMBS)	321
* 3.15.4	addDataSeries(id as string, yData() as double)	322
* 3.15.5	Constructor	323
* 3.15.6	Constructor(xData as CDDArrayMBS)	323
* 3.15.7	Constructor(xData() as double)	324
* 3.15.8	extendLength(len as Integer)	324
* 3.15.9	setSubsetRange(xStart as double, xEnd as double, resolution as Integer = 0)	325
– 3.16.1	class CDDDataSetMBS	326
* 3.16.3	Constructor	326
* 3.16.4	getDataColor as Integer	326
* 3.16.5	getDataName as string	326
* 3.16.6	getLegendIcon as string	327
* 3.16.7	getPosition(i as Integer) as Double	327
* 3.16.8	getUseYAxis as CDDAxisMBS	327
* 3.16.9	getValue(i as Integer) as Double	328
* 3.16.10	setData(data() as Double)	328
* 3.16.11	setDataColor(dataColor as color, edgeColor as color, shadowColor as color, shadowEdgeColor as color)	328
* 3.16.12	setDataColor(dataColor as Integer, edgeColor as Integer = -1, shadowColor as Integer = -1, shadowEdgeColor as Integer = -1)	328
* 3.16.13	setDataLabelFormat(formatString as string)	329
* 3.16.14	setDataLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = 0, fontangle as Double = 0) as CDDTextBoxMBS	329
* 3.16.15	setDataLabelStyle(font as string, fontsize as Double, fontcolor as color, fontangle as Double = 0) as CDDTextBoxMBS	330
* 3.16.16	setDataName(name as string)	330
* 3.16.17	setDataSymbol(drawobj as CDDDrawAreaMBS)	330
* 3.16.18	setDataSymbol(file as folderitem)	331
* 3.16.19	setDataSymbol(ImageFilePath as string)	332
* 3.16.20	setDataSymbol(pic as Picture)	333
* 3.16.21	setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1)	333
* 3.16.22	setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgeColor as color)	334
* 3.16.23	setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1)	335

* 3.16.24 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1)	335
* 3.16.25 setLineWidth(w as Integer)	336
* 3.16.26 setSymbolOffset(offsetX as Integer, offsetY as Integer)	336
* 3.16.27 setUseYAxis(axis as CDAxisMBS)	336
* 3.16.28 setUseYAxis(b as boolean=true)	337
– 3.17.1 class CDDiscreteHeatMapLayerMBS	338
* 3.17.3 colorAxis as CDCColorAxisMBS	338
* 3.17.4 setCellGap(gap as integer)	338
* 3.17.5 setColorAxis(x as integer, y as integer, alignment as integer, length as integer, orientation as integer) as CDCColorAxisMBS	339
* 3.17.6 setDirectColoring(b as boolean = true)	339
– 3.18.1 class CDDrawAreaMBS	340
* 3.18.3 adjustBrightness(c as Integer, brightness as Double) as Integer	340
* 3.18.4 affineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)	341
* 3.18.5 affineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)	341
* 3.18.6 angleGradientColor(cx as double, cy as double, a1 as double, a2 as double, r1 as double, r2 as double, c() as integer) as integer	342
* 3.18.7 arc(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, c as Integer)	343
* 3.18.8 circle(cx as Integer, cy as Integer, rx as Integer, ry as Integer, edgeColor as color, fillColor as color)	344
* 3.18.9 circle(cx as Integer, cy as Integer, rx as Integer, ry as Integer, edgeColor as Integer, fillColor as Integer)	344
* 3.18.10 clone(d as CDDrawAreaMBS, x as Integer, y as Integer, align as Integer, newWidth as Integer = -1, newHeight as Integer = -1, filter as Integer = 2, blur as Double = 1.0)	344
* 3.18.11 Constructor	344
* 3.18.12 cylinder(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, depthX as Integer, depthY as Integer, edgeColor as color, fillColor as color)	345
* 3.18.13 cylinder(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, depthX as Integer, depthY as Integer, edgeColor as Integer, fillColor as Integer)	345
* 3.18.14 dashLineColor(colorvalue as color, patternCode as Integer = &h0505) as Integer	346
* 3.18.15 dashLineColor(colorvalue as Integer, patternCode as Integer = &h0505) as Integer	346
* 3.18.16 enableVectorOutput	347
* 3.18.17 fill(x as Integer, y as Integer, colorvalue as color)	348
* 3.18.18 fill(x as Integer, y as Integer, colorvalue as color, borderColor as color)	348
* 3.18.19 fill(x as Integer, y as Integer, colorvalue as Integer)	348
* 3.18.20 fill(x as Integer, y as Integer, colorvalue as Integer, borderColor as Integer)	349
* 3.18.21 getARGBColor(c as Integer) as Integer	349
* 3.18.22 getHeight as Integer	349

* 3.18.23	getPixel(x as Integer, y as Integer) as Integer	350
* 3.18.24	getWidth as Integer	350
* 3.18.25	gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer	350
* 3.18.26	gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer	350
* 3.18.27	gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer	351
* 3.18.28	gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer	352
* 3.18.29	halfColor(c as Integer) as Integer	352
* 3.18.30	hCylinderTransform(yDiameter as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0)	353
* 3.18.31	hCylinderTransform(yDiameter as Integer, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)	353
* 3.18.32	hFlip	353
* 3.18.33	hline(x1 as Integer, x2 as Integer, y as Integer, c as Integer)	353
* 3.18.34	hTriangleTransform(tWidth as Integer = -1, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)	354
* 3.18.35	hTriangleTransform(tWidth as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0)	354
* 3.18.36	initDynamicLayer	354
* 3.18.37	line(x1 as Double, y1 as Double, x2 as Double, y2 as Double, colorValue as color, lineWidth as Integer = 1)	355
* 3.18.38	line(x1 as Double, y1 as Double, x2 as Double, y2 as Double, colorValue as Integer, lineWidth as Integer = 1)	355
* 3.18.39	linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer	356
* 3.18.40	linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer	356
* 3.18.41	linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer	358
* 3.18.42	linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer	358
* 3.18.43	load(path as string) as boolean	359
* 3.18.44	loadData(data as MemoryBlock, ImgType as Integer = -1) as boolean	359
* 3.18.45	loadData(data as string, ImgType as Integer = -1) as boolean	359
* 3.18.46	loadGIF(path as string) as boolean	359
* 3.18.47	loadJPG(path as string) as boolean	360
* 3.18.48	loadPNG(path as string) as boolean	360
* 3.18.49	loadWMP(path as string) as boolean	360
* 3.18.50	merge(d as CDDrawAreaMBS, x as Integer, y as Integer, align as Integer, transparency as Integer)	361

* 3.18.51 move(xOffset as Double, yOffset as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)	361
* 3.18.52 move(xOffset as Double, yOffset as Double, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)	362
* 3.18.53 out(file as folderitem) as boolean	362
* 3.18.54 outBMP as string	362
* 3.18.55 outBMP(file as folderitem) as boolean	363
* 3.18.56 outGIF as string	363
* 3.18.57 outGIF(file as folderitem) as boolean	363
* 3.18.58 outJPG(file as folderitem, quality as Integer = 80) as boolean	363
* 3.18.59 outJPG(quality as Integer = 80) as string	364
* 3.18.60 outPDF as string	364
* 3.18.61 outPDF(file as folderitem) as boolean	364
* 3.18.62 outPicture as picture	365
* 3.18.63 outPNG as string	365
* 3.18.64 outPNG(file as folderitem) as boolean	365
* 3.18.65 outSVG(file as folderitem, options as string = "") as boolean	365
* 3.18.66 outSVG(options as string = "") as string	366
* 3.18.67 outWMP as string	366
* 3.18.68 outWMP(file as folderitem) as boolean	367
* 3.18.69 patternColor(colors() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer	367
* 3.18.70 patternColor(colors() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer	367
* 3.18.71 patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer	368
* 3.18.72 Pixel(x as Integer, y as Integer, c as Integer)	369
* 3.18.73 polygon(x() as Double, y() as Double, edgeColor as color, fillColor as color)	369
* 3.18.74 polygon(x() as Double, y() as Double, edgeColor as Integer, fillColor as Integer)	369
* 3.18.75 polygon(x() as Integer, y() as Integer, edgeColor as color, fillColor as color)	370
* 3.18.76 polygon(x() as Integer, y() as Integer, edgeColor as Integer, fillColor as Integer)	370
* 3.18.77 polyShape(xy() as integer, edgeColor as color, fillColor as color)	370
* 3.18.78 polyShape(xy() as integer, edgeColor as Integer, fillColor as Integer)	371
* 3.18.79 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, data() as Integer, periodic as boolean=false) as Integer	372
* 3.18.80 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer	372
* 3.18.81 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer	373
* 3.18.82 rAffineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)	373
* 3.18.83 rAffineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)	374

* 3.18.84	rect(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, edgeColor as color, fillColor as color, raisedEffect as Integer = 0)	375
* 3.18.85	rect(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, edgeColor as Integer, fillColor as Integer, raisedEffect as Integer = 0)	375
* 3.18.86	reduceColors(colorCount as Integer, blackAndWhite as boolean=false) as Integer	375
* 3.18.87	removeDynamicLayer(keepOriginal as boolean = false)	376
* 3.18.88	resize(newWidth as Integer, newHeight as Integer, filter as Integer = 1, blur as Double = 1.0)	376
* 3.18.89	ring(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, edgeColor as color, fillColor as color)	376
* 3.18.90	ring(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, edgeColor as Integer, fillColor as Integer)	377
* 3.18.91	ringSector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, a1 as Double, a2 as Double, edgeColor as color, fillColor as color)	377
* 3.18.92	ringSector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, a1 as Double, a2 as Double, edgeColor as Integer, fillColor as Integer)	377
* 3.18.93	rotate(angle as Double, bgColor as color, cx as Double = -1, cy as Double = -1, filter as Integer = 2, blur as Double = 1.0)	378
* 3.18.94	rotate(angle as Double, bgColor as Integer = &hFFFFFF, cx as Double = -1, cy as Double = -1, filter as Integer = 2, blur as Double = 1.0)	378
* 3.18.95	sector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, edgeColor as color, fillColor as color)	378
* 3.18.96	sector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, edgeColor as Integer, fillColor as Integer)	379
* 3.18.97	setAntiAlias(shapeAntiAlias as boolean=true, textAntiAlias as Integer = 2)	379
* 3.18.98	setAntiAliasText(value as Integer)	380
* 3.18.99	setBgColor(c as color)	380
* 3.18.100	setBgColor(c as Integer)	380
* 3.18.101	setClipRect(left as Integer, top as Integer, right as Integer, bottom as Integer)	381
* 3.18.102	setColorTable(colors() as color, offset as Integer)	381
* 3.18.103	setColorTable(colors() as Integer, offset as Integer)	382
* 3.18.104	setDefaultFonts(normal as string, bold as string, italic as string, boldItalic as string)	382
* 3.18.105	setDitherMethod(value as Integer)	382
* 3.18.106	setFontTable(index as Integer, font as string)	383
* 3.18.107	setInterlace(value as boolean)	384
* 3.18.108	setOutputOptions(options as string)	384
* 3.18.109	setPaletteMode(value as Integer)	384
* 3.18.110	setResource(id as string, data as MemoryBlock)	385
* 3.18.111	setResource(id as string, data as string)	386
* 3.18.112	setResource(id as string, drawArea as CDDrawAreaMBS)	387
* 3.18.113	setSearchPath(path as string)	387
* 3.18.114	setSize(width as Integer, height as Integer, bgColor as color)	388

- * 3.18.115 setSize(width as Integer, height as Integer, bgColor as Integer = &hFFFFFF) 388
- * 3.18.116 setTransparentColor(value as color) 388
- * 3.18.117 setTransparentColor(value as Integer) 389
- * 3.18.118 shearTransform(xShear as Double, yShear as Double = 0, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 390
- * 3.18.119 shearTransform(xShear as Double, yShear as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0) 390
- * 3.18.120 sphereTransform(xDiameter as Integer, yDiameter as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0) 391
- * 3.18.121 sphereTransform(xDiameter as Integer, yDiameter as Integer, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 391
- * 3.18.122 surface(cx1 as Double, y1 as Double, x2 as Double, y2 as Double, depthX as Integer, depthY as Integer, edgeColor as color, fillColor as color) 391
- * 3.18.123 surface(cx1 as Double, y1 as Double, x2 as Double, y2 as Double, depthX as Integer, depthY as Integer, edgeColor as Integer, fillColor as Integer) 392
- * 3.18.124 text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean) as CDTTFTTextMBS 392
- * 3.18.125 text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7) 393
- * 3.18.126 text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7) 394
- * 3.18.127 text(str as string, font as string, fontsize as Double) as CDTTFTTextMBS 395
- * 3.18.128 text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as color) 396
- * 3.18.129 text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as Integer) 396
- * 3.18.130 tile(d as CDDrawAreaMBS, transparency as Integer) 397
- * 3.18.131 vCylinderTransform(xDiameter as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0) 397
- * 3.18.132 vCylinderTransform(xDiameter as Integer, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 397
- * 3.18.133 vFlip 398
- * 3.18.134 vline(y1 as Integer, y2 as Integer, x as Integer, c as Integer) 398
- * 3.18.135 vTriangleTransform(tHeight as Integer = -1, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 398
- * 3.18.136 vTriangleTransform(tHeight as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0) 399
- * 3.18.137 waveTransform(period as Integer, amplitude as Double, direction as Double = 0, startAngle as Double = 0, longitudinal as boolean=false, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 399
- * 3.18.138 waveTransform(period as Integer, amplitude as Double, direction as Double, startAngle as Double, longitudinal as boolean, bgColor as color, filter as Integer = 2, blur as Double = 1.0) 400

* 3.18.140 Handle as Integer	400
* 3.18.141 Resources as Dictionary	400
– 3.19.1 class CDDrawObjMBS	401
* 3.19.3 Constructor	401
* 3.19.4 paint(d as CDDrawAreaMBS)	401
* 3.19.5 setZOrder(z as Integer)	401
– 3.20.1 class CDFinanceChartMBS	402
* 3.20.3 addAccDist(height as Integer, ColorValue as color) as CDXYChartMBS	404
* 3.20.4 addAccDist(height as Integer, ColorValue as Integer) as CDXYChartMBS	404
* 3.20.5 addADX(height as Integer, period as Integer, posColor as color, negColor as color, ColorValue as color) as CDXYChartMBS	404
* 3.20.6 addADX(height as Integer, period as Integer, posColor as Integer, negColor as Integer, ColorValue as Integer) as CDXYChartMBS	405
* 3.20.7 addAroon(height as Integer, period as Integer, upColor as color, downColor as color) as CDXYChartMBS	405
* 3.20.8 addAroon(height as Integer, period as Integer, upColor as Integer, downColor as Integer) as CDXYChartMBS	405
* 3.20.9 addAroonOsc(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS	406
* 3.20.10 addAroonOsc(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS	406
* 3.20.11 addATR(height as Integer, period as Integer, color1 as color, color2 as color) as CDXYChartMBS	407
* 3.20.12 addATR(height as Integer, period as Integer, color1 as Integer, color2 as Integer) as CDXYChartMBS	407
* 3.20.13 addBand(upperLine() as Double, lowerLine() as Double, LineColor as color, FillColor as color, Name as string) as CDInterLineLayerMBS	407
* 3.20.14 addBand(upperLine() as Double, lowerLine() as Double, LineColor as Integer, FillColor as Integer, Name as string) as CDInterLineLayerMBS	408
* 3.20.15 addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as color, name as string) as CDBarLayerMBS	408
* 3.20.16 addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as Integer, name as string) as CDBarLayerMBS	409
* 3.20.17 addBarIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS	409
* 3.20.18 addBarIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS	410
* 3.20.19 addBollingerBand(period as Integer, bandWidth as Double, lineColor as color, FillColor as color) as CDInterLineLayerMBS	410
* 3.20.20 addBollingerBand(period as Integer, bandWidth as Double, lineColor as Integer, FillColor as Integer) as CDInterLineLayerMBS	410
* 3.20.21 addBollingerWidth(height as Integer, period as Integer, width as Double, colorValue as color) as CDXYChartMBS	411

- * 3.20.22 addBollingerWidth(height as Integer, period as Integer, width as Double, colorValue as Integer) as CDXYChartMBS 411
- * 3.20.23 addCandleStick(upColor as color, downColor as color) as CDCandleStickLayerMBS 412
- * 3.20.24 addCandleStick(upColor as Integer, downColor as Integer) as CDCandleStickLayerMBS 412
- * 3.20.25 addCCI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 412
- * 3.20.26 addCCI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 413
- * 3.20.27 addChaikinMoneyFlow(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS 413
- * 3.20.28 addChaikinMoneyFlow(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 413
- * 3.20.29 addChaikinOscillator(height as Integer, ColorValue as color) as CDXYChartMBS 414
- * 3.20.30 addChaikinOscillator(height as Integer, ColorValue as Integer) as CDXYChartMBS 414
- * 3.20.31 addChaikinVolatility(height as Integer, period1 as Integer, period2 as Integer, ColorValue as color) as CDXYChartMBS 415
- * 3.20.32 addChaikinVolatility(height as Integer, period1 as Integer, period2 as Integer, ColorValue as Integer) as CDXYChartMBS 415
- * 3.20.33 addCloseLine(ColorValue as color) as CDLineLayerMBS 415
- * 3.20.34 addCloseLine(ColorValue as Integer) as CDLineLayerMBS 416
- * 3.20.35 addCLV(height as Integer, ColorValue as color) as CDXYChartMBS 416
- * 3.20.36 addCLV(height as Integer, ColorValue as Integer) as CDXYChartMBS 416
- * 3.20.37 addComparison(data() as Double, ColorValue as color, Name as string) as CDLineLayerMBS 417
- * 3.20.38 addComparison(data() as Double, ColorValue as Integer, Name as string) as CDLineLayerMBS 417
- * 3.20.39 addDonchianChannel(period as Integer, lineColor as color, FillColor as color) as CDInterLineLayerMBS 417
- * 3.20.40 addDonchianChannel(period as Integer, lineColor as Integer, FillColor as Integer) as CDInterLineLayerMBS 418
- * 3.20.41 addDonchianWidth(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS 418
- * 3.20.42 addDonchianWidth(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 418
- * 3.20.43 addDPO(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS 419
- * 3.20.44 addDPO(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 419
- * 3.20.45 addEaseOfMovement(height as Integer, period as Integer, ColorValue1 as color, ColorValue2 as color) as CDXYChartMBS 420
- * 3.20.46 addEaseOfMovement(height as Integer, period as Integer, ColorValue1 as Integer, ColorValue2 as Integer) as CDXYChartMBS 420

- * 3.20.47 addEnvelop(period as Integer, range as Double, lineColor as color, FillColor as color) as CDInterLineLayerMBS 420
- * 3.20.48 addEnvelop(period as Integer, range as Double, lineColor as Integer, FillColor as Integer) as CDInterLineLayerMBS 421
- * 3.20.49 addExpMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS 421
- * 3.20.50 addExpMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS 421
- * 3.20.51 addFastStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as color, ColorValue2 as color) as CDXYChartMBS 422
- * 3.20.52 addFastStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as Integer, ColorValue2 as Integer) as CDXYChartMBS 422
- * 3.20.53 addHLOC(upColor as color, downColor as color) as CDHLOCLayerMBS 423
- * 3.20.54 addHLOC(upColor as Integer, downColor as Integer) as CDHLOCLayerMBS 423
- * 3.20.55 addIndicator(height as Integer) as CDXYChartMBS 423
- * 3.20.56 addLineIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as color, name as string) as CDLineLayerMBS 424
- * 3.20.57 addLineIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as Integer, name as string) as CDLineLayerMBS 424
- * 3.20.58 addLineIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS 425
- * 3.20.59 addLineIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS 425
- * 3.20.60 addMACD(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, signalColor as color, divColor as color) as CDXYChartMBS 426
- * 3.20.61 addMACD(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, signalColor as Integer, divColor as Integer) as CDXYChartMBS 426
- * 3.20.62 addMainChart(height as Integer) as CDXYChartMBS 427
- * 3.20.63 addMassIndex(height as Integer, ColorValue as color, upColor as color, downColor as color) as CDXYChartMBS 427
- * 3.20.64 addMassIndex(height as Integer, ColorValue as Integer, upColor as Integer, downColor as Integer) as CDXYChartMBS 427
- * 3.20.65 addMedianPrice(ColorValue as color) as CDLineLayerMBS 428
- * 3.20.66 addMedianPrice(ColorValue as Integer) as CDLineLayerMBS 428
- * 3.20.67 addMFI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 428
- * 3.20.68 addMFI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 429
- * 3.20.69 addMomentum(height as Integer, period as Integer, ColorValue as color) as CDXY-ChartMBS 429
- * 3.20.70 addMomentum(height as Integer, period as Integer, ColorValue as Integer) as CDXY-ChartMBS 429
- * 3.20.71 addNVI(height as Integer, period as Integer, ColorValue as color, signalColor as color) as CDXYChartMBS 430
- * 3.20.72 addNVI(height as Integer, period as Integer, ColorValue as Integer, signalColor as Integer) as CDXYChartMBS 430

- * 3.20.73 addOBV(height as Integer, ColorValue as color) as CDXYChartMBS 431
- * 3.20.74 addOBV(height as Integer, ColorValue as Integer) as CDXYChartMBS 431
- * 3.20.75 addParabolicSAR(accInitial as Double, accIncrement as Double, accMaximum as Double, symbolType as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDLineLayerMBS 431
- * 3.20.76 addParabolicSAR(accInitial as Double, accIncrement as Double, accMaximum as Double, symbolType as Integer, symbolSize as Integer, fillColor as Integer, edgeColor as Integer) as CDLineLayerMBS 432
- * 3.20.77 addPerformance(height as Integer, ColorValue as color) as CDXYChartMBS 432
- * 3.20.78 addPerformance(height as Integer, ColorValue as Integer) as CDXYChartMBS 432
- * 3.20.79 addPlotAreaTitle(alignment as Integer, text as string) as CDTextBoxMBS 433
- * 3.20.80 addPPO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, signalColor as color, divColor as color) as CDXYChartMBS 433
- * 3.20.81 addPPO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, signalColor as Integer, divColor as Integer) as CDXYChartMBS 434
- * 3.20.82 addPVI(height as Integer, period as Integer, ColorValue as color, signalColor as color) as CDXYChartMBS 434
- * 3.20.83 addPVI(height as Integer, period as Integer, ColorValue as Integer, signalColor as Integer) as CDXYChartMBS 434
- * 3.20.84 addPVO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, signalColor as color, divColor as color) as CDXYChartMBS 435
- * 3.20.85 addPVO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, signalColor as Integer, divColor as Integer) as CDXYChartMBS 435
- * 3.20.86 addPVT(height as Integer, ColorValue as color) as CDXYChartMBS 436
- * 3.20.87 addPVT(height as Integer, ColorValue as Integer) as CDXYChartMBS 436
- * 3.20.88 addROC(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS 436
- * 3.20.89 addROC(height as Integer, period as Integer, ColorValue as Integer) as CDXY-ChartMBS 437
- * 3.20.90 addRSI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 437
- * 3.20.91 addRSI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 437
- * 3.20.92 addSimpleMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS 438
- * 3.20.93 addSimpleMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS 438
- * 3.20.94 addSlowStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as color, ColorValue2 as color) as CDXYChartMBS 439
- * 3.20.95 addSlowStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as Integer, ColorValue2 as Integer) as CDXYChartMBS 439
- * 3.20.96 addStdDev(height as Integer, period as Integer, ColorValue as color) as CDXY-ChartMBS 439
- * 3.20.97 addStdDev(height as Integer, period as Integer, ColorValue as Integer) as CDXY-ChartMBS 440

- * 3.20.98 addStochRSI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 440
- * 3.20.99 addStochRSI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 440
- * 3.20.100 addThreshold(chart as CDXYChartMBS, layer as CDLineLayerMBS, topRange as Double, topColor as color, bottomRange as Double, bottomColor as color) 441
- * 3.20.101 addThreshold(chart as CDXYChartMBS, layer as CDLineLayerMBS, topRange as Double, topColor as Integer, bottomRange as Double, bottomColor as Integer) 441
- * 3.20.102 addTriMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS 442
- * 3.20.103 addTriMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS 443
- * 3.20.104 addTRIX(height as Integer, Period as Integer, ColorValue as color) as CDXY-ChartMBS 443
- * 3.20.105 addTRIX(height as Integer, Period as Integer, ColorValue as Integer) as CDXY-ChartMBS 443
- * 3.20.106 addTypicalPrice(ColorValue as color) as CDLineLayerMBS 444
- * 3.20.107 addTypicalPrice(ColorValue as Integer) as CDLineLayerMBS 444
- * 3.20.108 addUltimateOscillator(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 444
- * 3.20.109 addUltimateOscillator(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 445
- * 3.20.110 addVolBars(height as Integer, upColor as color, downColor as color, flatColor as color) as CDBarLayerMBS 445
- * 3.20.111 addVolBars(height as Integer, upColor as Integer, downColor as Integer, flatColor as Integer) as CDBarLayerMBS 446
- * 3.20.112 addVolIndicator(height as Integer, upColor as color, downColor as color, flatColor as color) as CDXYChartMBS 446
- * 3.20.113 addVolIndicator(height as Integer, upColor as Integer, downColor as Integer, flatColor as Integer) as CDXYChartMBS 446
- * 3.20.114 addWeightedClose(ColorValue as color) as CDLineLayerMBS 447
- * 3.20.115 addWeightedClose(ColorValue as Integer) as CDLineLayerMBS 447
- * 3.20.116 addWeightedMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS 448
- * 3.20.117 addWeightedMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS 448
- * 3.20.118 addWilliamR(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 448
- * 3.20.119 addWilliamR(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 449
- * 3.20.120 Constructor(width as Integer) 449
- * 3.20.121 currentChart as CDXYChartMBS 449
- * 3.20.122 enableAntiAlias(antiAlias as boolean) 450

* 3.20.123	getToolTipDateFormat as string	450
* 3.20.124	mainChart as CDXYChartMBS	450
* 3.20.125	setAxisOnRight(b as Boolean)	450
* 3.20.126	setData(timeStamps() as Double, highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, volData() as Double, extraPoints as Integer)	451
* 3.20.127	setDateLabelFormat(yearFormat as string, firstMonthFormat as string, otherMonthFormat as string, firstDayFormat as string, otherDayFormat as string, firstHourFormat as string, otherHourFormat as string)	451
* 3.20.128	setDateLabelSpacing(labelSpacing as Integer)	452
* 3.20.129	setLegendStyle(font as string, fontSize as Double, fontColor as color, bgColor as color)	452
* 3.20.130	setLegendStyle(font as string, fontSize as Double, fontColor as Integer, bgColor as Integer)	453
* 3.20.131	setLogScale(b as Boolean)	453
* 3.20.132	setMargins(leftMargin as Integer, topMargin as Integer, rightMargin as Integer, bottomMargin as Integer)	453
* 3.20.133	setNumberLabelFormat(formatString as string)	454
* 3.20.134	setPercentageAxis as CDAxisMBS	454
* 3.20.135	setPlotAreaBorder(borderColor as color, borderGap as Integer)	454
* 3.20.136	setPlotAreaBorder(borderColor as Integer, borderGap as Integer)	455
* 3.20.137	setPlotAreaStyle(bgColor as color, majorHGridColor as color, majorVGridColor as color, minorHGridColor as color, minorVGridColor as color)	455
* 3.20.138	setPlotAreaStyle(bgColor as Integer, majorHGridColor as Integer, majorVGridColor as Integer, minorHGridColor as Integer, minorVGridColor as Integer)	455
* 3.20.139	setToolTipDateFormat(monthFormat as string, dayFormat as string, hourFormat as string)	456
* 3.20.140	setXAxisStyle(font as string, fontSize as Double, fontColor as color, fontAngle as Double)	456
* 3.20.141	setXAxisStyle(font as string, fontSize as Double, fontColor as Integer, fontAngle as Double)	457
* 3.20.142	setYAxisStyle(font as string, fontSize as Double, fontColor as color, bgColor as color)	457
* 3.20.143	setYAxisStyle(font as string, fontSize as Double, fontColor as Integer, bgColor as Integer)	457
– 3.21.1	class CDFinanceSimulatorMBS	459
* 3.21.3	Constructor(seed as Integer, startTime as Double, endTime as Double, resolution as Integer)	459
* 3.21.4	Constructor(seed as string, startTime as Double, endTime as Double, resolution as Integer)	459
* 3.21.5	getCloseData as CDArrayMBS	460
* 3.21.6	getHighData as CDArrayMBS	460
* 3.21.7	getLowData as CDArrayMBS	460
* 3.21.8	getOpenData as CDArrayMBS	460

* 3.21.9	getTimeStamps as CDArrayMBS	460
* 3.21.10	getVolData as CDArrayMBS	461
– 3.22.1	class CDHLOCLayerMBS	462
* 3.22.3	setColorMethod(colorMethod as Integer, riseColor as color, fallColor as color, leadValue as Double = -1.7E308)	462
* 3.22.4	setColorMethod(colorMethod as Integer, riseColor as Integer, fallColor as Integer = -1, leadValue as Double = -1.7E308)	462
– 3.23.1	class CDImageMapHandlerMBS	464
* 3.23.3	Constructor(ImageMap as string)	464
* 3.23.4	getHotSpot(xCoordinate as double, yCoordinate as double) as integer	465
* 3.23.5	getKey(i as Integer) as string	465
* 3.23.6	getValue(i as Integer) as string	466
* 3.23.7	getValue(key as string) as string	466
– 3.24.1	class CDInterLineLayerMBS	467
* 3.24.3	setGapColor(gapColor12 as Color, gapColor21 as Color)	467
* 3.24.4	setGapColor(gapColor12 as Integer, gapColor21 as Integer = -1)	467
– 3.25.1	class CDLayerMBS	469
* 3.25.3	addCustomAggregateLabel(dataItem as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS	469
* 3.25.4	addCustomAggregateLabel(dataItem as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS	470
* 3.25.5	addCustomDataLabel(dataSet as Integer, dataItem as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS	470
* 3.25.6	addCustomDataLabel(dataSet as Integer, dataItem as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS	471
* 3.25.7	addCustomGroupLabel(dataGroup as Integer, dataItem as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS	471
* 3.25.8	addCustomGroupLabel(dataGroup as Integer, dataItem as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS	472
* 3.25.9	addDataGroup(name as string)	472
* 3.25.10	addDataSet(data as CDArrayMBS, colorvalue as color, name as string = "") as CDDataSetMBS	473
* 3.25.11	addDataSet(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "") as CDDDataSetMBS	473
* 3.25.12	addDataSet(data() as Double, colorvalue as color, name as string = "") as CDDataSetMBS	474
* 3.25.13	addDataSet(data() as Double, colorvalue as Integer = -1, name as string = "") as CDDDataSetMBS	474

* 3.25.14 addExtraField(numbers() as Double)	475
* 3.25.15 addExtraField(texts() as string)	475
* 3.25.16 alignLayer(layer as CDLayerMBS, dataSet as Integer)	476
* 3.25.17 Constructor	476
* 3.25.18 getDataSet(dataSet as Integer) as CDDatasetMBS	476
* 3.25.19 getDataSetByZ(z as Integer) as CDDatasetMBS	477
* 3.25.20 getDataSetCount as Integer	477
* 3.25.21 getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string	478
* 3.25.22 getImageCoor(dataSet as Integer, dataItem as Integer = &h8000001, offsetX as Integer = 0, offsetY as Integer = 0) as string	478
* 3.25.23 getImageCoor2(dataItem as Integer, offsetX as Integer = 0, offsetY as Integer = 0) as string	479
* 3.25.24 getLegendIcon(dataSetNo as Integer) as string	480
* 3.25.25 getNearestXValue(target as Double) as Double	480
* 3.25.26 getXCoor(value as Double) as Integer	481
* 3.25.27 getXIndexOf(xValue as Double, tolerance as Double = 0) as Integer	481
* 3.25.28 getXPosition(i as Integer) as Double	482
* 3.25.29 getYCoor(value as Double, axis as boolean=true) as Integer	482
* 3.25.30 getYCoor(value as Double, axis as CDAxisMBS) as Integer	482
* 3.25.31 moveBack(layer as CDLayerMBS=nil)	483
* 3.25.32 moveFront(layer as CDLayerMBS=nil)	483
* 3.25.33 set3D(d as Integer = -1, zGap as Integer = 0)	484
* 3.25.34 setAggregateLabelFormat(formatString as string)	484
* 3.25.35 setAggregateLabelStyle(font as string = "", fontSize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS	484
* 3.25.36 setAggregateLabelStyle(font as string, fontSize as Double, fontcolor as color, fontAngle as Double = 0) as CDTextBoxMBS	485
* 3.25.37 setBaseLine(BaseLine as Double)	485
* 3.25.38 setBorderColor(colorvalue as color, lightingEffect as Integer = 0)	486
* 3.25.39 setBorderColor(colorvalue as Integer, lightingEffect as Integer = 0)	486
* 3.25.40 setDataCombineMethod(m as Integer)	487
* 3.25.41 setDataLabelFormat(formatString as string)	487
* 3.25.42 setDataLabelStyle(font as string = "", fontSize as Double = 8, fontcolor as color, fontAngle as Double = 0) as CDTextBoxMBS	488
* 3.25.43 setDataLabelStyle(font as string = "", fontSize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS	488
* 3.25.44 setHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "")	489
* 3.25.45 setLegend(m as Integer)	490
* 3.25.46 setLegendOrder(dataSetOrder as Integer, layerOrder as Integer = -1)	490
* 3.25.47 setLineWidth(w as Integer)	491
* 3.25.48 setUseYAxis(axis as CDAxisMBS)	492

* 3.25.49	setUseYAxis2(b as boolean=true)	492
* 3.25.50	setXData(data as CDArrayMBS)	492
* 3.25.51	setXData(data() as Double)	493
* 3.25.52	setXData(dates() as date)	494
* 3.25.53	setXData(dates() as dateTime)	495
* 3.25.54	setXData(minValue as Double, maxValue as Double)	495
* 3.25.55	xZoneColor(threshold as Double, belowColor as color, aboveColor as color) as Integer	496
* 3.25.56	xZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer) as Integer	496
* 3.25.57	yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as boolean=true) as Integer	497
* 3.25.58	yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as CDAxisMBS) as Integer	498
* 3.25.59	yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as boolean=true) as Integer	498
* 3.25.60	yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as CDAxisMBS) as Integer	499
– 3.26.1	class CDLegendBoxMBS	500
* 3.26.3	addKey(pos as Integer, text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)	500
* 3.26.4	addKey(pos as Integer, text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)	501
* 3.26.5	addKey(text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)	501
* 3.26.6	addKey(text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)	502
* 3.26.7	addText(pos as Integer, text as string)	502
* 3.26.8	addText(text as string)	502
* 3.26.9	getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string	503
* 3.26.10	getImageCoor(dataItem as Integer, offsetX as Integer = 0, offsetY as Integer = 0) as string	503
* 3.26.11	setCols(noOfCols as Integer)	504
* 3.26.12	setKeyBorder(edgeColor as color, raisedEffect as Integer = 0)	504
* 3.26.13	setKeyBorder(edgeColor as Integer, raisedEffect as Integer = 0)	505
* 3.26.14	setKeySize(width as Integer = -1, height as Integer = -1, gap as Integer = -1)	505
* 3.26.15	setKeySpacing(keySpacing as Integer, lineSpacing as Integer = -1)	505
* 3.26.16	setLineStyleKey(b as boolean=true)	506
* 3.26.17	setReverse(b as boolean=true)	507
– 3.27.1	class CDLinearMeterMBS	508
* 3.27.3	addBar(startValue as Double, endValue as Double, colorvalue as color, effect as Integer = 0, roundedCorner as Integer = 0) as CDTextBoxMBS	508

* 3.27.4	addBar(startValue as Double, endValue as Double, colorvalue as Integer, effect as Integer = 0, roundedCorner as Integer = 0) as CDTextBoxMBS	509
* 3.27.5	addZone(startValue as Double, endValue as Double, colorvalue as color, label as string = "") as CDTextBoxMBS	509
* 3.27.6	addZone(startValue as Double, endValue as Double, colorvalue as Integer, label as string = "") as CDTextBoxMBS	509
* 3.27.7	Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)	510
* 3.27.8	Constructor(width as Integer, height as Integer, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)	510
* 3.27.9	setMeter(leftX as Integer, topY as Integer, width as Integer, height as Integer, axisPos as Integer = 4, isReversed as boolean=false)	511
* 3.27.10	setRail(railColor as color, railWidth as Integer = 2, railOffset as Integer = 6)	511
* 3.27.11	setRail(railColor as Integer, railWidth as Integer = 2, railOffset as Integer = 6)	511
– 3.28.1	class CDLineLayerMBS	513
* 3.28.3	getLine(dataSet as Integer = 0) as CDLineObjMBS	513
* 3.28.4	setFastLineMode(b as boolean = true)	513
* 3.28.5	setGapColor(lineColor as color, lineWidth as Integer = -1)	514
* 3.28.6	setGapColor(lineColor as Integer, lineWidth as Integer = -1)	514
* 3.28.7	setImageMapWidth(Width as Integer)	515
* 3.28.8	setSymbolScale(zDataX() as Double, scaleTypeX as Integer = 0)	515
* 3.28.9	setSymbolScale(zDataX() as Double, scaleTypeX as Integer, zDataY() as Double, scaleTypeY as Integer = 0)	516
– 3.29.1	class CDLineMBS	518
* 3.29.3	setColor(c as color)	518
* 3.29.4	setColor(c as Integer)	518
* 3.29.5	setPos(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer)	518
* 3.29.6	setWidth(w as Integer)	519
– 3.30.1	class CDLineObjMBS	520
* 3.30.3	Constructor	520
– 3.31.1	class CDMarkMBS	521
* 3.31.3	getLine as CDLineObjMBS	521
* 3.31.4	setDrawOnTop(b as boolean)	521
* 3.31.5	setLineWidth(width as Integer)	521
* 3.31.6	setMarkColor(lineColor as color, textColor as color, tickColor as color)	522
* 3.31.7	setMarkColor(lineColor as Integer, textColor as Integer = -1, tickColor as Integer = -1)	522
* 3.31.8	setValue(value as Double)	522
– 3.32.1	class CDMeterPointerMBS	523
* 3.32.3	setColor(fillColor as color, edgeColor as color)	523
* 3.32.4	setColor(fillColor as Integer, edgeColor as Integer = -1)	523

* 3.32.5	setPos(value as Double)	524
* 3.32.6	setShape(pointerCoor() as Integer)	524
* 3.32.7	setShape(pointerCoor() as Integer, lengthRatio as Double)	525
* 3.32.8	setShape(pointerCoor() as Integer, lengthRatio as Double, widthRatio as Double)	526
* 3.32.9	setShape(pointerType as Integer)	527
* 3.32.10	setShape(pointerType as Integer, lengthRatio as Double)	528
* 3.32.11	setShape(pointerType as Integer, lengthRatio as Double, widthRatio as Double)	529
* 3.32.12	setShapeAndOffset(pointerCoor() as Integer)	530
* 3.32.13	setShapeAndOffset(pointerCoor() as Integer, startOffset as Double)	531
* 3.32.14	setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double)	532
* 3.32.15	setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)	533
* 3.32.16	setShapeAndOffset(pointerType as Integer)	534
* 3.32.17	setShapeAndOffset(pointerType as Integer, startOffset as Double)	535
* 3.32.18	setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double)	536
* 3.32.19	setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)	537
– 3.33.1	class CDMLTableMBS	539
* 3.33.3	appendCol as CDTextBoxMBS	539
* 3.33.4	appendRow as CDTextBoxMBS	539
* 3.33.5	getCell(col as Integer, row as Integer) as CDTextBoxMBS	539
* 3.33.6	getColCount as Integer	539
* 3.33.7	getColStyle(col as Integer) as CDTextBoxMBS	540
* 3.33.8	getColWidth(col as Integer) as Integer	540
* 3.33.9	getHeight as Integer	540
* 3.33.10	getRowCount as Integer	540
* 3.33.11	getRowHeight(row as Integer) as Integer	540
* 3.33.12	getRowStyle(row as Integer) as CDTextBoxMBS	540
* 3.33.13	getStyle as CDTextBoxMBS	541
* 3.33.14	getWidth as Integer	541
* 3.33.15	insertCol(col as Integer) as CDTextBoxMBS	541
* 3.33.16	insertRow(row as Integer) as CDTextBoxMBS	541
* 3.33.17	layout	541
* 3.33.18	setCell(col as Integer, row as Integer, width as Integer, height as Integer, text as string) as CDTextBoxMBS	541
* 3.33.19	setPos(x as Integer, y as Integer, alignment as Integer = 7)	542
* 3.33.20	setText(col as Integer, row as Integer, text as string) as CDTextBoxMBS	542
– 3.34.1	class CDMultiChartMBS	543
* 3.34.3	addChart(x as Integer, y as Integer, c as CDBaseChartMBS)	543

* 3.34.4	Constructor(width as Integer = 640, height as Integer = 480, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)	543
* 3.34.5	Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)	544
* 3.34.6	getChart(index as Integer) as CDBaseChartMBS	544
* 3.34.7	getChartCount as Integer	544
* 3.34.8	setMainChart(c as CDBaseChartMBS)	545
– 3.35.1	class CDMultiPagePDFMBS	546
* 3.35.3	addPage(chart as CDBaseChartMBS)	546
* 3.35.4	addPage(DrawArea as CDDrawAreaMBS)	546
* 3.35.5	Constructor	547
* 3.35.6	outPDF as String	547
* 3.35.7	outPDF(file as FolderItem) as Boolean	547
* 3.35.8	outPDF(path as String) as Boolean	548
– 3.37.1	class CDPieChartMBS	550
* 3.37.3	Constructor(width as Integer = 640, height as Integer = 480, bgcolor as color, edgeColor as color, raisedEffect as Integer = 0)	550
* 3.37.4	Constructor(width as Integer = 640, height as Integer = 480, bgcolor as Integer = &hFFFF0000, edgeColor as Integer = &hFF000000, raisedEffect as Integer = 0)	550
* 3.37.5	sector(sectorNo as Integer) as CDSectorMBS	551
* 3.37.6	set3D(depth as Integer = -1, angle as Double = -1, shadowMode as boolean=false)	551
* 3.37.7	set3D(depths() as Double, angle as Double = 45, shadowMode as boolean=false)	552
* 3.37.8	setData(data() as Double)	552
* 3.37.9	setData(data() as Double, label() as string)	553
* 3.37.10	setDonutSize(x as Integer, y as Integer, r as Integer, r2 as Integer)	553
* 3.37.11	setExplode(sectorNo as Integer, distance as Integer = -1)	553
* 3.37.12	setExplodeGroup(startSector as Integer, endSector as Integer, distance as Integer = -1)	554
* 3.37.13	setJoinLine(joinLineColor as color, joinLineWidth as Integer = -1)	554
* 3.37.14	setJoinLine(joinLineColor as Integer, joinLineWidth as Integer = -1)	554
* 3.37.15	setLabelFormat(formatString as string)	555
* 3.37.16	setLabelLayout(layoutMethod as Integer, pos as Integer = -1, topBound as Integer = -1, bottomBound as Integer = -1)	556
* 3.37.17	setLabelPos(pos as Integer, joinLineColor as color)	557
* 3.37.18	setLabelPos(pos as Integer, joinLineColor as Integer = -1)	558
* 3.37.19	setLabelStyle(font as string = "", fontsize as Double = 8, fontColor as Integer = &hfff0002) as CDTextBoxMBS	558
* 3.37.20	setLabelStyle(font as string, fontsize as Double, fontColor as color) as CDTextBoxMBS	559
* 3.37.21	setLineColor(edgeColor as color, joinLineColor as color)	559
* 3.37.22	setLineColor(edgeColor as Integer, joinLineColor as Integer = -1)	559
* 3.37.23	setPieSize(x as Integer, y as Integer, r as Integer)	560

* 3.37.24	setSectorStyle(shadingMethod as Integer, edgeColor as color, edgeWidth as Integer = -1)	560
* 3.37.25	setSectorStyle(shadingMethod as Integer, edgeColor as Integer = -1, edgeWidth as Integer = -1)	560
* 3.37.26	setStartAngle(startAngle as Double, clockWise as boolean=true)	561
– 3.38.1	class CDPlotAreaMBS	562
* 3.38.3	Constructor	562
* 3.38.4	getBottomY as Integer	562
* 3.38.5	getHeight as Integer	562
* 3.38.6	getLeftX as Integer	563
* 3.38.7	getRightX as Integer	563
* 3.38.8	getTopY as Integer	563
* 3.38.9	getWidth as Integer	563
* 3.38.10	moveGridBefore(layer as CDLayerMBS=nil)	563
* 3.38.11	set4QBgColor(Q1Color as color, Q2Color as color, Q3Color as color, Q4Color as color, edgeColor as color)	564
* 3.38.12	set4QBgColor(Q1Color as Integer, Q2Color as Integer, Q3Color as Integer, Q4Color as Integer, edgeColor as Integer = -1)	564
* 3.38.13	setAltBgColor(horizontal as Boolean, color1 as color, color2 as color, edgeColor as color)	565
* 3.38.14	setAltBgColor(horizontal as Boolean, color1 as Integer, color2 as Integer, edgeColor as Integer = -1)	565
* 3.38.15	setBackground(colorvalue as color, altBgColor as color, edgeColor as color)	565
* 3.38.16	setBackground(colorvalue as Integer, altBgColor as Integer = -1, edgeColor as Integer = -1)	566
* 3.38.17	setBackground(file as folderitem, align as Integer = 5)	566
* 3.38.18	setGridAxis(xGridAxis as CDAxisMBS, yGridAxis as CDAxisMBS)	567
* 3.38.19	setGridColor(hGridColor as color, vGridColor as color, minorHGridColor as color, minorVGridColor as color)	567
* 3.38.20	setGridColor(hGridColor as Integer, vGridColor as Integer = &hff000000, minorHGridColor as Integer = -1, minorVGridColor as Integer = -1)	567
* 3.38.21	setGridWidth(hGridWidth as Integer, vGridWidth as Integer = -1, minorHGridWidth as Integer = -1, minorVGridWidth as Integer = -1)	568
– 3.40.1	class CDPolarChartMBS	570
* 3.40.3	addAreaLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS	570
* 3.40.4	addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS	571
* 3.40.5	addAreaLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS	571
* 3.40.6	addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS	572
* 3.40.7	addAreaLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS	572

- * 3.40.8 addAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarAreaLayerMBS 573
- * 3.40.9 addLineLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarLineLayerMBS 573
- * 3.40.10 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarLineLayerMBS 574
- * 3.40.11 addLineLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarLineLayerMBS 575
- * 3.40.12 addLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarLineLayerMBS 575
- * 3.40.13 addLineLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarLineLayerMBS 576
- * 3.40.14 addLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarLineLayerMBS 576
- * 3.40.15 addSplineAreaLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarSplineAreaLayerMBS 577
- * 3.40.16 addSplineAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarSplineAreaLayerMBS 577
- * 3.40.17 addSplineAreaLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarSplineAreaLayerMBS 578
- * 3.40.18 addSplineAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarSplineAreaLayerMBS 578
- * 3.40.19 addSplineAreaLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarSplineAreaLayerMBS 579
- * 3.40.20 addSplineAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarSplineAreaLayerMBS 580
- * 3.40.21 addSplineLineLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarSplineLineLayerMBS 580
- * 3.40.22 addSplineLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarSplineLineLayerMBS 581
- * 3.40.23 addSplineLineLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarSplineLineLayerMBS 581
- * 3.40.24 addSplineLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarSplineLineLayerMBS 582
- * 3.40.25 addSplineLineLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarSplineLineLayerMBS 582
- * 3.40.26 addSplineLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarSplineLineLayerMBS 583
- * 3.40.27 addVectorLayer(rdata() as Double, adata() as Double, lengths() as Double, directions() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "") as CDPolarVectorLayerMBS 583
- * 3.40.28 addVectorLayer(rdata() as Double, adata() as Double, lengths() as Double, directions() as Double, lengthScale as Integer, colorvalue as color, name as string = "") as CDPolarVectorLayerMBS 585

- * 3.40.29 Constructor(width as Integer = 640, height as Integer = 480, bgColor as color, edgeColor as color, raisedEffect as Integer = 0) 585
- * 3.40.30 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0) 585
- * 3.40.31 getXCoord(r as Double, a as Double) as Integer 586
- * 3.40.32 getYCoord(r as Double, a as Double) as Integer 586
- * 3.40.33 setGridColor(rGridColor as color, rGridWidth as Integer, aGridColor as color, aGridWidth as Integer = 1) 587
- * 3.40.34 setGridColor(rGridColor as Integer = &h80000000, rGridWidth as Integer = 1, aGridColor as Integer = &h80000000, aGridWidth as Integer = 1) 587
- * 3.40.35 setGridStyle(polygonGrid as boolean, gridOnTop as boolean=true) 587
- * 3.40.36 setPlotArea(x as Integer, y as Integer, r as Integer, bgColor as color, edgeColor as color, edgeWidth as Integer = 1) 588
- * 3.40.37 setPlotArea(x as Integer, y as Integer, r as Integer, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000, edgeWidth as Integer = 1) 588
- * 3.40.38 setPlotAreaBg(bgColor1 as color, bgColor2 as color, altRings as boolean = true) 589
- * 3.40.39 setPlotAreaBg(bgColor1 as Integer, bgColor2 as Integer, altRings as boolean = true) 589
- * 3.40.40 setStartAngle(startAngle as Double, clockwise as boolean=true) 589
- * 3.40.42 angularAxis as CDAngularAxisMBS 590
- * 3.40.43 radialAxis as CDRadialAxisMBS 590
- 3.41.1 class CDPolarLayerMBS 591
 - * 3.41.3 addCustomDataLabel(i as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS 591
 - * 3.41.4 addCustomDataLabel(i as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS 592
 - * 3.41.5 Constructor 592
 - * 3.41.6 getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string 592
 - * 3.41.7 getImageCoord(dataItem as Integer, offsetX as Integer = 0, offsetY as Integer = 0) as string 593
 - * 3.41.8 setAngles(data() as Double) 594
 - * 3.41.9 setBorderColor(edgeColor as color) 594
 - * 3.41.10 setBorderColor(edgeColor as Integer) 594
 - * 3.41.11 setData(data() as Double, colorvalue as color, name as string = "") 595
 - * 3.41.12 setData(data() as Double, colorvalue as Integer = -1, name as string = "") 595
 - * 3.41.13 setDataLabelFormat(formatString as string) 595
 - * 3.41.14 setDataLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = 0, fontangle as Double = 0) as CDTextBoxMBS 595
 - * 3.41.15 setDataLabelStyle(font as string, fontsize as Double, fontcolor as color, fontangle as Double = 0) as CDTextBoxMBS 596
 - * 3.41.16 setDataSymbol(area as CDDrawAreaMBS) 596

* 3.41.17 setDataSymbol(image as folderitem)	597
* 3.41.18 setDataSymbol(pic as Picture)	597
* 3.41.19 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgecolor as Integer = -1)	598
* 3.41.20 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgecolor as Integer = -1)	599
* 3.41.21 setDataSymbol(symbol as Integer, size as Integer = 7, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1)	599
* 3.41.22 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1)	600
* 3.41.23 setHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "")	600
* 3.41.24 setImageMapWidth(width as Integer)	601
* 3.41.25 setLineWidth(w as Integer)	602
* 3.41.26 setSymbolOffset(offsetX as Integer, offsetY as Integer)	602
* 3.41.27 setSymbolScale(data() as Double, scaleType as Integer = 0)	602
– 3.42.1 class CDPolarLineLayerMBS	604
* 3.42.3 setCloseLoop(b as boolean)	604
* 3.42.4 setGapColor(lineColor as color, lineWidth as Integer)	604
* 3.42.5 setGapColor(lineColor as Integer, lineWidth as Integer)	604
– 3.43.1 class CDPolarSplineAreaLayerMBS	606
* 3.43.3 setTension(tension as Double)	606
– 3.44.1 class CDPolarSplineLineLayerMBS	607
* 3.44.3 setTension(tension as Double)	607
– 3.45.1 class CDPolarVectorLayerMBS	608
* 3.45.3 setArrowAlignment(alignment as Integer)	608
* 3.45.4 setArrowHead(polygon() as Integer)	608
* 3.45.5 setArrowHead(width as Integer, height as Integer)	609
* 3.45.6 setArrowStem(polygon() as Integer)	609
* 3.45.7 setIconSize(height as Integer, width as Integer = 0)	610
* 3.45.8 setVector(lengths() as Double, directions() as Double, lengthScale as Integer = 0)	610
* 3.45.9 setVectorMargin(startMargin as Double)	611
* 3.45.10 setVectorMargin(startMargin as Double, endMargin as Double)	611
– 3.46.1 class CDPyramidChartMBS	612
* 3.46.3 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)	612
* 3.46.4 Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)	613
* 3.46.5 getLayer(layerNo as Integer) as CDPyramidLayerMBS	613
* 3.46.6 setCenterLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS	613

* 3.46.7	setCenterLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS	614
* 3.46.8	setConeSize(cx as Integer, cy as Integer, radius as Integer, height as Integer)	614
* 3.46.9	setData(data() as Double)	614
* 3.46.10	setData(data() as Double, labels() as string)	615
* 3.46.11	setFunnelSize(cx as Integer, cy as Integer, radius as Integer, height as Integer, tubeRadius as Double = 0.2, tubeHeight as Double = 0.3)	615
* 3.46.12	setGradientShading(startBrightness as Double, endBrightness as Double)	616
* 3.46.13	setJoinLine(ColorValue as color, width as Integer = -1)	617
* 3.46.14	setJoinLine(ColorValue as Integer, width as Integer = -1)	617
* 3.46.15	setJoinLineGap(pyramidGap as Integer)	617
* 3.46.16	setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer)	618
* 3.46.17	setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer, textGap as Integer)	619
* 3.46.18	setLayerBorder(ColorValue as color, width as Integer = -1)	619
* 3.46.19	setLayerBorder(ColorValue as Integer, width as Integer = -1)	620
* 3.46.20	setLayerGap(layerGap as Double)	620
* 3.46.21	setLeftLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS	621
* 3.46.22	setLeftLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS	621
* 3.46.23	setLighting(ambientIntensity as Double = 0.5, diffuseIntensity as Double = 0.5, specularIntensity as Double = 1, shininess as Double = 8)	622
* 3.46.24	setPyramidSides(noOfSides as Integer)	622
* 3.46.25	setPyramidSize(cx as Integer, cy as Integer, radius as Integer, height as Integer)	622
* 3.46.26	setRightLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS	623
* 3.46.27	setRightLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS	624
* 3.46.28	setViewAngle(elevation as Double, rotation as Double = 0, twist as Double = 0)	624
- 3.47.1	class CDPyramidLayerMBS	626
* 3.47.3	Constructor	626
* 3.47.4	setCenterLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS	626
* 3.47.5	setCenterLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS	627
* 3.47.6	setColor(ColorValue as color)	627
* 3.47.7	setColor(ColorValue as Integer)	627
* 3.47.8	setJoinLine(ColorValue as color, width as Integer = -1)	627
* 3.47.9	setJoinLine(ColorValue as Integer, width as Integer = -1)	628
* 3.47.10	setJoinLineGap(pyramidGap as Integer)	628
* 3.47.11	setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer)	629

* 3.47.12	setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer, textGap as Integer)	629
* 3.47.13	setLayerBorder(ColorValue as color, width as Integer = -1)	630
* 3.47.14	setLayerBorder(ColorValue as Integer, width as Integer = -1)	630
* 3.47.15	setLayerGap(layerGap as Double)	631
* 3.47.16	setLeftLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS	631
* 3.47.17	setLeftLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS	632
* 3.47.18	setRightLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS	632
* 3.47.19	setRightLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS	633
- 3.49.1	class CDRanSeriesMBS	635
* 3.49.3	Constructor(seed as Integer)	635
* 3.49.4	create(seed as Integer) as CDRanSeriesMBS	635
* 3.49.5	fillDateSeries(Values() as Double, startTime as double, tickInc as double, weekDayOnly as Boolean = false)	636
* 3.49.6	fillSeries(Values() as Double, minValue as double, maxValue as double)	636
* 3.49.7	fillSeries(Values() as Double, startValue as double, minDelta as double, maxDelta as double, lowerLimit as double = -1E+308, upperLimit as double = 1E+308)	636
* 3.49.8	get2DSeries(xLen as Integer, yLen as Integer, minValue as Double, maxValue as Double) as Double()	637
* 3.49.9	getDateSeries(len as Integer, startTime as Double, tickInc as Double, weekDayOnly as boolean = false) as Double()	637
* 3.49.10	getSeries(len as Integer, minValue as Double, maxValue as Double) as Double()	638
* 3.49.11	getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double) as Double()	638
* 3.49.12	getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double) as Double()	639
* 3.49.13	getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double) as Double()	640
- 3.50.1	class CDRanTableMBS	641
* 3.50.3	Constructor(seed as Integer, noOfCols as Integer, noOfRows as Integer)	641
* 3.50.4	getCol(colNo as Integer) as CDArrayMBS	641
* 3.50.5	selectDate(colNo as Integer, minDate as Double, maxDate as Double) as Integer	642
* 3.50.6	setCol(colNo as Integer, minValue as Double, maxValue as Double)	642
* 3.50.7	setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double)	642
* 3.50.8	setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double)	643
* 3.50.9	setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double)	644

* 3.50.10 setDateCol(i as Integer, startTime as Double, tickInc as Double, weekDayOnly as boolean=false)	644
* 3.50.11 setHLOCCols(i as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double = 0.0)	645
* 3.50.12 setHLOCCols(i as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double)	645
– 3.52.1 class CDSectorMBS	648
* 3.52.3 Constructor	648
* 3.52.4 getImageCoor(offsetX as Integer = 0, offsetY as Integer = 0) as string	648
* 3.52.5 getLabelCoor(offsetX as Integer = 0, offsetY as Integer = 0) as string	649
* 3.52.6 setColor(colorvalue as color, edgeColor as color, joinLineColor as color)	649
* 3.52.7 setColor(colorvalue as Integer, edgeColor as Integer = -1, joinLineColor as Integer = -1)	649
* 3.52.8 setExplode(distance as Integer = -1)	650
* 3.52.9 setJoinLine(joinLineColor as color, joinLineWidth as Integer = 1)	650
* 3.52.10 setJoinLine(joinLineColor as Integer, joinLineWidth as Integer = 1)	650
* 3.52.11 setLabelFormat(formatString as string = "")	651
* 3.52.12 setLabelLayout(layoutMethod as Integer, pos as Integer = -1)	652
* 3.52.13 setLabelPos(pos as Integer, joinLineColor as color)	652
* 3.52.14 setLabelPos(pos as Integer, joinLineColor as Integer = -1)	652
* 3.52.15 setLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002) as CDTextBoxMBS	653
* 3.52.16 setLabelStyle(font as string, fontsize as Double, fontcolor as color) as CDTextBoxMBS	653
* 3.52.17 setStyle(shadingMethod as Integer, edgeColor as color, edgeWidth as Integer = -1)	654
* 3.52.18 setStyle(shadingMethod as Integer, edgeColor as Integer = -1, edgeWidth as Integer = -1)	654
– 3.53.1 class CDSplineLayerMBS	655
* 3.53.3 setMonotonicity(m as Integer)	655
* 3.53.4 setTension(tension as Double)	655
– 3.54.1 class CDStepLineLayerMBS	657
* 3.54.3 setAlignment(a as Integer)	657
– 3.55.1 class CDSurfaceChartMBS	658
* 3.55.3 addSurfaceLine(x() as double, y() as double, colorValue as Integer, lineWidth as Integer = -1, side as Integer = 0)	658
* 3.55.4 addSurfaceLine(x1 as double, y1 as double, x2 as double, y2 as double, colorValue as Integer, lineWidth as Integer = -1, side as Integer = 0)	658
* 3.55.5 addSurfaceZone(x1 as double, y1 as double, x2 as double, y2 as double, fillColor as Integer, edgeColor as Integer = &hff000000, edgeWidth as Integer = 1)	659
* 3.55.6 addXYProjection(offset as Integer = 0)	659
* 3.55.7 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)	660

* 3.55.8	Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)	660
* 3.55.9	getValuesAtPixel(x as Integer, y as Integer) as Double()	661
* 3.55.10	setBackSideBrightness(brightness as Double)	661
* 3.55.11	setBackSideColor(ColorValue as color)	661
* 3.55.12	setBackSideColor(ColorValue as Integer)	662
* 3.55.13	setBackSideLighting(ambientLight as Double, diffuseLight as Double, specularLight as Double, shininess as Double)	662
* 3.55.14	setContourColor(contourColor as color, minorContourColor as color)	663
* 3.55.15	setContourColor(contourColor as Integer, minorContourColor as Integer = -1)	663
* 3.55.16	setData(xData() as Double, yData() as Double, zData() as Double)	663
* 3.55.17	setData(xData() as double, yData() as double, zData() as double, wData() as double)	664
* 3.55.18	setInterpolation(xSamples as Integer, ySamples as Integer = -1, isSmooth as Boolean = true)	664
* 3.55.19	setInterpolation(xSamples as Integer, ySamples as Integer, isSmooth as Boolean, isColorSmooth as Boolean)	665
* 3.55.20	setLighting(ambientIntensity as Double, diffuseIntensity as Double, specularIntensity as Double, shininess as Double)	666
* 3.55.21	setShadingMode(shadingMode as Integer, wireWidth as Integer = 1)	666
* 3.55.22	setSurfaceAxisGrid(majorXGridColor as color, majorYGridColor as color, minorXGridColor as color, minorYGridColor as color)	667
* 3.55.23	setSurfaceAxisGrid(majorXGridColor as Integer, majorYGridColor as Integer = -1, minorXGridColor as Integer = -1, minorYGridColor as Integer = -1)	668
* 3.55.24	setSurfaceDataGrid(xGridColor as color, yGridColor as color)	668
* 3.55.25	setSurfaceDataGrid(xGridColor as Integer, yGridColor as Integer = -1)	668
* 3.55.26	setSurfaceTexture(patternColor as color)	669
* 3.55.27	setSurfaceTexture(patternColor as integer)	669
* 3.55.28	setWContourColor(wContourColor as color, wMinorContourColor as color)	670
* 3.55.29	setWContourColor(wContourColor as integer, wMinorContourColor as Integer = -1)	670
– 3.56.1	class CDTextBoxMBS	672
* 3.56.3	setAlignment(a as Integer)	672
* 3.56.4	setFontAngle(angle as Double, vertical as boolean=false)	672
* 3.56.5	setFontColor(colorvalue as color)	673
* 3.56.6	setFontColor(colorvalue as Integer)	673
* 3.56.7	setFontSize(fontHeight as Double, fontWidth as Double = 0)	673
* 3.56.8	setFontStyle(font as string, fontIndex as Integer = 0)	674
* 3.56.9	setHeight(height as Integer)	674
* 3.56.10	setMargin(leftMargin as Integer, rightMargin as Integer, topMargin as Integer, bottomMargin as Integer)	674
* 3.56.11	setMargin(m as Integer)	675
* 3.56.12	setMaxWidth(width as Integer)	675

* 3.56.13	setText(text as string)	676
* 3.56.14	setTruncate(maxWidth as Integer, maxLines as Integer = 1)	676
* 3.56.15	setWidth(width as Integer)	676
– 3.57.1	class CDThreeDChartMBS	678
* 3.57.3	colorAxis as CDColorAxisMBS	678
* 3.57.4	Constructor	678
* 3.57.5	getElevationAngle as double	678
* 3.57.6	getPlotRegionDepth as integer	679
* 3.57.7	getPlotRegionHeight as integer	679
* 3.57.8	getPlotRegionWidth as integer	679
* 3.57.9	getRotationAngle as double	679
* 3.57.10	getXCoord(xValue as Double, yValue as Double, zValue as Double) as Integer	679
* 3.57.11	getYCoord(xValue as Double, yValue as Double, zValue as Double) as Integer	680
* 3.57.12	setColorAxis(x as Integer, y as Integer, alignment as Integer, length as Integer, orientation as Integer) as CDColorAxisMBS	680
* 3.57.13	setPerspective(perspective as Double)	680
* 3.57.14	setPlotRegion(cx as Integer, cy as Integer, xWidth as Integer, yDepth as Integer, zHeight as Integer)	681
* 3.57.15	setViewAngle(elevation as Double, rotation as Double = 0, twist as Double = 0)	681
* 3.57.16	setWallColor(xyColor as color, yzColor as color, zxColor as color, borderColor as color)	682
* 3.57.17	setWallColor(xyColor as Integer, yzColor as Integer = -1, zxColor as Integer = -1, borderColor as Integer = -1)	683
* 3.57.18	setWallGrid(majorXGridColor as color, majorYGridColor as color, majorZGridColor as color, minorXGridColor as color, minorYGridColor as color, minorZGridColor as color)	683
* 3.57.19	setWallGrid(majorXGridColor as Integer, majorYGridColor as Integer = -1, majorZGridColor as Integer = -1, minorXGridColor as Integer = -1, minorYGridColor as Integer = -1, minorZGridColor as Integer = -1)	684
* 3.57.20	setWallThickness(xyThickness as Integer, yzThickness as Integer = -1, zxThickness as Integer = -1)	685
* 3.57.21	setWallVisibility(xyVisible as boolean, yzVisible as boolean, zxVisible as boolean)	685
* 3.57.22	setZAxisPos(pos as Integer)	685
* 3.57.24	xAxis as CDAxisMBS	686
* 3.57.25	yAxis as CDAxisMBS	686
* 3.57.26	zAxis as CDAxisMBS	686
– 3.58.1	class CDThreeDScatterChartMBS	687
* 3.58.3	addScatterGroup(xData() as Double, yData() as Double, zData() as Double, name as string = "", symbol as Integer = 7, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDThreeDScatterGroupMBS	687
* 3.58.4	addScatterGroup(xData() as Double, yData() as Double, zData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDThreeDScatterGroupMBS	688

* 3.58.5	Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)	688
* 3.58.6	Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)	689
– 3.59.1	class CDThreeDScatterGroupMBS	690
* 3.59.3	Constructor	690
* 3.59.4	setDataSymbol(DrawArea as CDDrawAreaMBS)	690
* 3.59.5	setDataSymbol(file as folderitem)	691
* 3.59.6	setDataSymbol(ImageFilePath as string)	692
* 3.59.7	setDataSymbol(pic as Picture)	692
* 3.59.8	setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color)	693
* 3.59.9	setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1)	694
* 3.59.10	setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1)	695
* 3.59.11	setDataSymbol(symbol as Integer, size as Integer, fillColor as color)	695
* 3.59.12	setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1)	696
* 3.59.13	setDropLine	697
* 3.59.14	setDropLine(dropLineColor as color, dropLineWidth as Integer = 1)	697
* 3.59.15	setDropLine(dropLineColor as Integer, dropLineWidth as Integer = 1)	698
* 3.59.16	setLegendIcon(width as Integer, height as Integer = -1, color as Integer = -1)	698
* 3.59.17	setLegendIcon(width as Integer, height as Integer, color as color)	698
* 3.59.18	setSymbolOffset(offsetX as Integer, offsetY as Integer)	699
– 3.60.1	class CDTreeMapChartMBS	700
* 3.60.3	Constructor(width as integer = 640, height as integer = 480, bgColor as integer = &hfff0000, edgeColor as integer = &hff000000, raisedEffect as integer = 0)	700
* 3.60.4	Constructor(width as integer, height as integer, bgColor as color, edgeColor as color, raisedEffect as integer = 0)	700
* 3.60.5	getLevelPrototype(index as integer) as CDTreeMapNodeMBS	701
* 3.60.6	setMapLevel(n as integer)	701
* 3.60.7	setPlotArea(x as integer, y as integer, width as integer, height as integer)	702
* 3.60.9	RootNode as CDTreeMapNodeMBS	702
– 3.61.1	class CDTreeMapNodeMBS	703
* 3.61.3	addExtraField(numbers() as double)	703
* 3.61.4	addExtraField(texts() as string)	704
* 3.61.5	Constructor	704
* 3.61.6	Node(Index as Integer) as CDTreeMapNodeMBS	704
* 3.61.7	setColors(fillColor as color, edgeColor as color, raisedEffect as integer = -2147483647)	705
* 3.61.8	setColors(fillColor as integer, edgeColor as integer = -1, raisedEffect as integer = -2147483647)	705

* 3.61.9 setData(data() as double)	705
* 3.61.10 setData(data() as double, labels() as string)	706
* 3.61.11 setData(data() as double, labels() as string, colors() as Color)	706
* 3.61.12 setData(data() as double, labels() as string, colors() as Integer)	707
* 3.61.13 setLabelFormat(formatString as string = " { label } ", font as string = "normal", fontSize as Integer = 10, fontColor as Integer = &hfff0002, alignment as Integer = 7)	707
* 3.61.14 setLabelFormat(formatString as string, font as string, fontSize as Integer, fontColor as Color, alignment as Integer = 7)	708
* 3.61.15 setLayoutAspectMultiplier(multiplier as Double)	708
* 3.61.16 setLayoutAspectRatio(ratio as Double)	709
* 3.61.17 setLayoutMethod(layoutMethod as Integer, layoutDirection as Integer = -1)	709
* 3.61.18 setLayoutMethod(layoutMethod as Integer, layoutDirection as Integer, swapXY as Boolean)	709
* 3.61.19 setPos(x as Integer, y as Integer, w as Integer, h as Integer)	710
* 3.61.20 setSorting(Mode as Integer)	710
* 3.61.22 BottomY as Integer	710
* 3.61.23 Handle as Integer	711
* 3.61.24 Height as Integer	711
* 3.61.25 Label as String	711
* 3.61.26 LeftX as Integer	711
* 3.61.27 NodeCount as Integer	712
* 3.61.28 RightX as Integer	712
* 3.61.29 TopY as Integer	712
* 3.61.30 Value as Double	712
* 3.61.31 Width as Integer	712
– 3.62.1 class CDTrendLayerMBS	713
* 3.62.3 addConfidenceBand(confidence as Double, upperFillColor as color, upperEdgeColor as color, upperLineWidth as Integer, lowerFillColor as color, lowerEdgeColor as color, lower- LineWidth as Integer = -1)	714
* 3.62.4 addConfidenceBand(confidence as Double, upperFillColor as Integer, upperEdgeColor as Integer = &hFF000000, upperLineWidth as Integer = -1, lowerFillColor as Integer = -1, lowerEdgeColor as Integer = -1, lowerLineWidth as Integer = -1)	714
* 3.62.5 addPredictionBand(confidence as Double, upperFillColor as color, upperEdgeColor as color, upperLineWidth as Integer, lowerFillColor as color, lowerEdgeColor as color, lower- LineWidth as Integer = -1)	715
* 3.62.6 addPredictionBand(confidence as Double, upperFillColor as Integer, upperEdgeColor as Integer = &hFF000000, upperLineWidth as Integer = -1, lowerFillColor as Integer = -1, lowerEdgeColor as Integer = -1, lowerLineWidth as Integer = -1)	715
* 3.62.7 getCoefficient(index as Integer) as Double	716
* 3.62.8 getCorrelation as Double	716
* 3.62.9 getIntercept as Double	716
* 3.62.10 getLine as CDLineObjMBS	716
* 3.62.11 getSlope as Double	716

	47
* 3.62.12 getStdError as Double	716
* 3.62.13 setImageMapWidth(w as Integer)	717
* 3.62.14 setRegressionType(regressionType as Integer)	717
– 3.63.1 class CDTTFTTextMBS	718
* 3.63.3 Constructor	718
* 3.63.4 destroy	718
* 3.63.5 draw(x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7)	718
* 3.63.6 draw(x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7)	719
* 3.63.7 getHeight as Integer	719
* 3.63.8 getLineDistance as Integer	719
* 3.63.9 getLineHeight as Integer	719
* 3.63.10 getWidth as Integer	719
– 3.64.1 class CDVectorLayerMBS	721
* 3.64.3 setArrowAlignment(alignment as Integer)	721
* 3.64.4 setArrowHead(polygon() as Integer)	721
* 3.64.5 setArrowHead(width as Integer, height as Integer = 0)	722
* 3.64.6 setArrowStem(polygon() as Integer)	722
* 3.64.7 setIconSize(height as Integer, width as Integer = 0)	723
* 3.64.8 setVector(lengths() as Double, directions() as Double, lengthScale as Integer = 0)	723
* 3.64.9 setVectorMargin(startMargin as Double)	724
* 3.64.10 setVectorMargin(startMargin as Double, endMargin as Double)	724
– 3.65.1 class CDViewPortControlBaseMBS	726
* 3.65.3 Constructor	726
* 3.65.4 getProperty(attr as integer) as integer	727
* 3.65.5 handleMouseDown(x as double, y as double)	727
* 3.65.6 handleMouseMove(x as double, y as double, isDragging as boolean)	727
* 3.65.7 handleMouseUp(x as double, y as double)	727
* 3.65.8 isOnPlotArea(x as double, y as double) as boolean	728
* 3.65.9 paintViewPort	728
* 3.65.10 setMouseMargin(mouseMargin as Integer, cornerMargin as Integer)	728
* 3.65.11 setProperty(attr as integer, value as integer)	729
* 3.65.12 setZoomScrollDirection(zoomDirection as integer, scrollDirection as integer)	729
* 3.65.14 Chart as CDBaseChartMBS	729
* 3.65.15 ClickToCenter as Boolean	729
* 3.65.16 CornerMargin as Integer	730
* 3.65.17 Cursor as Integer	730
* 3.65.18 DragBorderToResize as Boolean	730
* 3.65.19 DragInsideToMove as Boolean	730
* 3.65.20 DragOutsideToSelect as Boolean	730
* 3.65.21 MouseMargin as Integer	731
* 3.65.22 needUpdateChart as Boolean	731

* 3.65.23	needUpdateDisplay as Boolean	731
* 3.65.24	needUpdateImageMap as Boolean	731
* 3.65.25	SelectionBorderColor as Color	731
* 3.65.26	SelectionBorderWidth as Integer	732
* 3.65.27	ViewPortBorderColor as Color	732
* 3.65.28	ViewPortBorderWidth as Integer	732
* 3.65.29	ViewPortExternalColor as Color	733
* 3.65.30	ViewPortFillColor as Color	733
* 3.65.31	ViewPortManager as CDViewPortManagerMBS	733
– 3.66.1	class CDViewPortManagerMBS	734
* 3.66.3	canZoomIn(zoomDirection as Integer) as boolean	734
* 3.66.4	canZoomOut(zoomDirection as Integer) as boolean	735
* 3.66.5	clearAllRanges	735
* 3.66.6	commitPendingSyncAxis(baseChart as CDBaseChartMBS)	735
* 3.66.7	Constructor	736
* 3.66.8	dragTo(scrollDirection as integer, x as double, y as double) as boolean	736
* 3.66.9	getPlotAreaHeight as Integer	736
* 3.66.10	getPlotAreaLeft as Integer	736
* 3.66.11	getPlotAreaTop as Integer	737
* 3.66.12	getPlotAreaWidth as Integer	737
* 3.66.13	getValueAtViewPort(id as string, ratio as Double, isLogScale as boolean = false) as Double	737
* 3.66.14	getViewPortAtValue(id as string, ratio as Double, isLogScale as boolean = false) as Double	737
* 3.66.15	getViewPortHeight as Double	738
* 3.66.16	getViewPortLeft as Double	738
* 3.66.17	getViewPortTop as Double	738
* 3.66.18	getViewPortWidth as Double	738
* 3.66.19	getZoomInHeightLimit as Double	739
* 3.66.20	getZoomInWidthLimit as Double	739
* 3.66.21	getZoomOutHeightLimit as Double	739
* 3.66.22	getZoomOutWidthLimit as Double	739
* 3.66.23	getZoomXYRatio as double	739
* 3.66.24	inExtendedPlotArea(x as Integer, y as Integer) as boolean	739
* 3.66.25	inPlotArea(x as double, y as double) as boolean	740
* 3.66.26	setChartMetrics(metrics as string)	740
* 3.66.27	setFullRange(ID as string, minValue as Double, maxValue as Double)	740
* 3.66.28	setKeepAspectRatio(value as boolean)	741
* 3.66.29	setPlotAreaMouseMargin(leftMargin as Integer, rightMargin as Integer, topMargin as Integer, bottomMargin as Integer)	741
* 3.66.30	setViewPortHeight(value as Double)	741
* 3.66.31	setViewPortLeft(value as Double)	742

* 3.66.32 setViewPortTop(value as Double)	742
* 3.66.33 setViewPortWidth(value as Double)	742
* 3.66.34 setZoomInHeightLimit(value as Double)	742
* 3.66.35 setZoomInWidthLimit(value as Double)	743
* 3.66.36 setZoomOutHeightLimit(value as Double)	743
* 3.66.37 setZoomOutWidthLimit(value as Double)	744
* 3.66.38 startDrag	744
* 3.66.39 syncDateAxisWithViewPort(id as string, axis as CDAxisMBS)	744
* 3.66.40 syncLinearAxisWithViewPort(id as string, axis as CDAxisMBS)	745
* 3.66.41 syncLogAxisWithViewPort(id as string, axis as CDAxisMBS)	745
* 3.66.42 updateFullRangeH(id as string, minValue as Double, maxValue as Double, updateType as Integer) as boolean	746
* 3.66.43 updateFullRangeV(id as string, minValue as Double, maxValue as Double, updateType as Integer) as boolean	746
* 3.66.44 validateViewPort	746
* 3.66.45 zoomAround(x as double, y as double, xZoomRatio as double, yZoomRatio as double) as boolean	747
* 3.66.46 zoomAt(zoomDirection as integer, x as double, y as double, zoomRatio as double) as boolean	747
* 3.66.47 zoomTo(zoomDirection as integer, x1 as double, y1 as double, x2 as double, y2 as double) as boolean	747
– 3.67.1 class CDXYChartMBS	749
* 3.67.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS	749
* 3.67.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS	750
* 3.67.5 addAreaLayer(dataCombineMethod as Integer = 1, depth as Integer = 0) as CDAreaLayerMBS	751
* 3.67.6 addAreaLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS	752
* 3.67.7 addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS	752
* 3.67.8 addAreaLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDAreaLayerMBS	753
* 3.67.9 addAreaLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDAreaLayerMBS	754
* 3.67.10 addAxis(align as Integer, offset as Integer) as CDAxisMBS	754
* 3.67.11 addBarLayer(data() as Double, colors() as color, depth as Integer = 0) as CDBarLayerMBS	755
* 3.67.12 addBarLayer(data() as Double, colors() as color, names() as string, depth as Integer = 0) as CDBarLayerMBS	755
* 3.67.13 addBarLayer(data() as Double, colors() as Integer, depth as Integer = 0) as CDBarLayerMBS	756

- * 3.67.14 addBarLayer(data() as Double, colors() as Integer, names() as string, depth as Integer = 0) as CDBarLayerMBS 757
- * 3.67.15 addBarLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDBarLayerMBS 758
- * 3.67.16 addBarLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDBarLayerMBS 759
- * 3.67.17 addBarLayer(dataCombineMethod as Integer = 3, depth as Integer = 0) as CDBarLayerMBS 759
- * 3.67.18 addBoxLayer(boxTop() as Double, boxBottom() as Double, colorvalue as color, name as string = "") as CDBoxWhiskerLayerMBS 761
- * 3.67.19 addBoxLayer(boxTop() as Double, boxBottom() as Double, colorvalue as Integer = -1, name as string = "") as CDBoxWhiskerLayerMBS 761
- * 3.67.20 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS 761
- * 3.67.21 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS 763
- * 3.67.22 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 765
- * 3.67.23 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as color, whiskerColor as color, edgeColor as color) as CDBoxWhiskerLayerMBS 767
- * 3.67.24 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as Integer = -1, whiskerColor as Integer = &hfff0001, edgeColor as Integer = -1) as CDBoxWhiskerLayerMBS 767
- * 3.67.25 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS 769
- * 3.67.26 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS 770
- * 3.67.27 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 771
- * 3.67.28 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double) as CDBoxWhiskerLayerMBS 773
- * 3.67.29 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS 774
- * 3.67.30 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS 775
- * 3.67.31 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS 775
- * 3.67.32 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS 777

- * 3.67.33 addCandleStickLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, riseColor as color, fallColor as color, edgeColor as color) as CDCandleStickLayerMBS 778
- * 3.67.34 addCandleStickLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, riseColor as Integer = -1, fallColor as Integer = 0, edgeColor as Integer = &hfff0001) as CDCandleStickLayerMBS 778
- * 3.67.35 addContourLayer(xData() as Double, yData() as Double, zData() as Double) as CDContourLayerMBS 779
- * 3.67.36 addDiscreteHeatMapLayer(xGrid() as double, yGrid() as Double, zData() as Double) as CDDiscreteHeatMapLayerMBS 780
- * 3.67.37 addDiscreteHeatMapLayer(zData() as Double, xCount as Integer) as CDDiscreteHeatMapLayerMBS 780
- * 3.67.38 addHLOCLayer as CDHLOCLayerMBS 781
- * 3.67.39 addHLOCLayer(highData() as Double, lowData() as Double) as CDHLOCLayerMBS 782
- * 3.67.40 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double) as CDHLOCLayerMBS 783
- * 3.67.41 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as color) as CDHLOCLayerMBS 785
- * 3.67.42 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS 785
- * 3.67.43 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer = -1) as CDHLOCLayerMBS 787
- * 3.67.44 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 787
- * 3.67.45 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer = -1) as CDHLOCLayerMBS 788
- * 3.67.46 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 789
- * 3.67.47 addInterLineLayer(line1 as CDLineObjMBS, line2 as CDLineObjMBS, color12 as color, color21 as color) as CDInterLineLayerMBS 791
- * 3.67.48 addInterLineLayer(line1 as CDLineObjMBS, line2 as CDLineObjMBS, color12 as Integer, color21 as Integer = -1) as CDInterLineLayerMBS 791
- * 3.67.49 addLineLayer(data as CDArrayMBS, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- * 3.67.50 addLineLayer(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- * 3.67.51 addLineLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793
- * 3.67.52 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793

- * 3.67.53 addLineLayer(dataCombineMethod as Integer = 0, depth as Integer = 0) as CDLineLayerMBS 794
- * 3.67.54 addLineLayer(fastDB as CDDataAcceleratorMBS, seriesId as string, color as integer = -1, name as string = "") as CDLineLayerMBS 795
- * 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796
- * 3.67.56 addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 797
- * 3.67.57 addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 798
- * 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- * 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800
- * 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- * 3.67.61 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 803
- * 3.67.62 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 804
- * 3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 804
- * 3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 805
- * 3.67.65 addScatterLayer(xDate() as Double, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 806
- * 3.67.66 addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 808
- * 3.67.67 addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 809
- * 3.67.68 addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 810
- * 3.67.69 addSplineLayer as CDSplineLayerMBS 811

- * 3.67.70 addSplineLayer(data() as Double, colorvalue as color, name as string = "") as CDSplineLayerMBS 811
- * 3.67.71 addSplineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDSplineLayerMBS 811
- * 3.67.72 addStepLineLayer as CDStepLineLayerMBS 812
- * 3.67.73 addStepLineLayer(data() as Double, colorvalue as color, name as string = "") as CDStepLineLayerMBS 813
- * 3.67.74 addStepLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDStepLineLayerMBS 813
- * 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813
- * 3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 814
- * 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815
- * 3.67.78 addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 816
- * 3.67.79 addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 817
- * 3.67.80 addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 818
- * 3.67.81 addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 819
- * 3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 820
- * 3.67.83 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- * 3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- * 3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 822
- * 3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 823
- * 3.67.87 addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "") as CDVectorLayerMBS 824
- * 3.67.88 addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "") as CDVectorLayerMBS 825
- * 3.67.89 addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer = 0, colorvalue as integer = -1, name as string = "") as CDVectorLayerMBS 826
- * 3.67.90 addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer, colorvalue as color, name as string = "") as CDVectorLayerMBS 826

- * 3.67.91 addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "") as CDVectorLayerMBS 827
- * 3.67.92 addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "") as CDVectorLayerMBS 828
- * 3.67.93 Constructor(width as Integer = 640, height as Integer = 480, bgcolor as Integer = &hFFFF0000, edgeColor as Integer = &hFF000000, raisedEffect as Integer = 0) 829
- * 3.67.94 Constructor(width as Integer, height as Integer, bgcolor as color, edgeColor as color, raisedEffect as Integer = 0) 829
- * 3.67.95 getLayer(i as Integer) as CDLayerMBS 830
- * 3.67.96 getLayerByZ(i as Integer) as CDLayerMBS 830
- * 3.67.97 getLayerCount as Integer 830
- * 3.67.98 getNearestXValue(xCoor as Double) as Double 830
- * 3.67.99 getPlotArea as CDPlotAreaMBS 831
- * 3.67.100 getXCoor(value as Double) as Integer 831
- * 3.67.101 getXValue(xCoor as Integer) as Double 831
- * 3.67.102 getYCoor(value as Double, yAxis as CDAxisMBS=nil) as Integer 832
- * 3.67.103 getYValue(yCoor as Integer, axis as CDAxisMBS = nil) as Double 833
- * 3.67.104 layoutAxes 833
- * 3.67.105 packPlotArea(leftX as Integer, topY as Integer, rightX as Integer, bottomY as Integer, minWidth as Integer = 0, minHeight as Integer = 0) 833
- * 3.67.106 setAxisAtOrigin(originMode as Integer = 3, symmetryMode as Integer = 0) 834
- * 3.67.107 setClipping(margin as Integer = 0) 835
- * 3.67.108 setPlotArea(x as Integer, y as Integer, width as Integer, height as Integer, bgColor as color, altBgColor as color, edgeColor as color, hGridColor as color, vGridColor as color) as CDPlotAreaMBS 836
- * 3.67.109 setPlotArea(x as Integer, y as Integer, width as Integer, height as Integer, bgColor as Integer = &hff000000, altBgColor as Integer = -1, edgeColor as Integer = -1, hGridColor as Integer = &hc0c0c0, vGridColor as Integer = &hff000000) as CDPlotAreaMBS 836
- * 3.67.110 setTrimData(startPos as Integer, len as Integer = &h7ffffff) 837
- * 3.67.111 setXAxisOnTop(value as boolean=true) 837
- * 3.67.112 setYAxisOnRight(value as boolean=true) 838
- * 3.67.113 swapXY(value as boolean=true) 838
- * 3.67.114 syncYAxis(slope as Double = 1, intercept as Double = 0) 838
- * 3.67.115 xScaleColor(scale() as double) as Integer 839
- * 3.67.116 xZoneColor(threshold as Double, belowColor as color, aboveColor as color) as Integer 839
- * 3.67.117 xZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer) as Integer 839
- * 3.67.118 yScaleColor(scale() as double, yAxis as CDAxisMBS = nil) as Integer 840
- * 3.67.119 yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as CDAxisMBS=nil) as Integer 841

- * 3.67.120 yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as CDAxisMBS=nil) as Integer 841
- * 3.67.122 xAxis as CDAxisMBS 842
- * 3.67.123 xAxis2 as CDAxisMBS 842
- * 3.67.124 yAxis as CDAxisMBS 842
- * 3.67.125 yAxis2 as CDAxisMBS 843

Chapter 2

List of all classes

• CDAngularAxisMBS	61
• CDAngularMeterMBS	70
• CDAreaLayerMBS	107
• CDArrayMBS	110
• CDAxisMBS	145
• CDBarLayerMBS	178
• CDBaseBoxLayerMBS	185
• CDBaseChartMBS	189
• CDBaseMeterMBS	279
• CDBoxMBS	291
• CDBoxWhiskerLayerMBS	296
• CDCandleStickLayerMBS	299
• CDColorAxisMBS	302
• CDContourLayerMBS	312
• CDDataAcceleratorMBS	321
• CDDataSetMBS	326
• CDDiscreteHeatMapLayerMBS	338
• CDDrawAreaMBS	340
• CDDrawObjMBS	401

• CDFinanceChartMBS	402
• CDFinanceSimulatorMBS	459
• CDHLOCLayerMBS	462
• CDImageMapHandlerMBS	464
• CDInterLineLayerMBS	467
• CDLayerMBS	469
• CDLegendBoxMBS	500
• CDLinearMeterMBS	508
• CDLineLayerMBS	513
• CDLineMBS	518
• CDLineObjMBS	520
• CDMarkMBS	521
• CDMeterPointerMBS	523
• CDMLTableMBS	539
• CDMultiChartMBS	543
• CDMultiPagePDFMBS	546
• CDNotInitializedExceptionMBS	549
• CDPieChartMBS	550
• CDPlotAreaMBS	562
• CDPolarAreaLayerMBS	569
• CDPolarChartMBS	570
• CDPolarLayerMBS	591
• CDPolarLineLayerMBS	604
• CDPolarSplineAreaLayerMBS	606
• CDPolarSplineLineLayerMBS	607
• CDPolarVectorLayerMBS	608
• CDPyramidChartMBS	612
• CDPyramidLayerMBS	626
• CDRadialAxisMBS	634

	59
• CDRanSeriesMBS	635
• CDRanTableMBS	641
• CDScatterLayerMBS	647
• CDSectorMBS	648
• CDSplineLayerMBS	655
• CDStepLineLayerMBS	657
• CDSurfaceChartMBS	658
• CDTextBoxMBS	672
• CDThreeDChartMBS	678
• CDThreeDScatterChartMBS	687
• CDThreeDScatterGroupMBS	690
• CDTreeMapChartMBS	700
• CDTreeMapNodeMBS	703
• CDTrendLayerMBS	713
• CDTTFFTextMBS	718
• CDVectorLayerMBS	721
• CDViewPortControlBaseMBS	726
• CDViewPortManagerMBS	734
• CDXYChartMBS	749

Chapter 3

ChartDirector

3.1 class CDAngularAxisMBS

3.1.1 class CDAngularAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The AngularAxis class represents angular axes in polar charts. The angular axis is the axis representing the angular component of a polar coordinate.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.1.2 Methods

3.1.3 addLabel(pos as Double, label as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an extra label on the axis.

Notes:

Parameter	Default	Description
pos	(Mandatory)	The position on the axis to add the label.
label	(Mandatory)	The text label to add.

3.1.4 addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addZone method, but uses color instead of integer data type for passing color values.

See also:

- 3.1.5 addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1) 62
- 3.1.6 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as color, edgeColor as color) 63
- 3.1.7 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as Integer, edgeColor as Integer) 63

3.1.5 addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an angular zone to the polar chart.

Notes: This method is just a short cut to `AngularAxis.addZone`, in which the starting radius is always 0, and ending radius is the radius of the polar plot region. In other words, the angular zone is a sector on the polar chart.

Parameters:

startValue	(Mandatory)	The data value that marks the start angular position of the zone.
endValue	(Mandatory)	The data value that marks the end angular position of the zone.
fillColor	(Mandatory)	The fill color of the zone.
edgeColor	-1	The edge color of the zone. The default is the same as the fill color.

See also:

- 3.1.4 addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color) 62
- 3.1.6 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as color, edgeColor as color) 63
- 3.1.7 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as Integer, edgeColor as Integer) 63

3.1.6 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addZone method, but uses color instead of integer data type for passing color values.

See also:

- 3.1.4 addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color) 62
- 3.1.5 addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1) 62
- 3.1.7 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as Integer, edgeColor as Integer) 63

3.1.7 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as Integer, edgeColor as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an angular zone to the polar chart.

Notes: An angular zone marks an angular region that spans from a start angle to an end angle, where the angles are specified as data values on the angular axis scale. The addZone method allows you to specify a starting and ending radius in pixels for drawing the zone.

Parameter	Default	Description
startValue	(Mandatory)	The data value that marks the start angular position of the zone.
endValue	(Mandatory)	The data value that marks the end angular position of the zone.
startRadius	(Mandatory)	The starting radius of the zone in pixels.
endRadius	(Mandatory)	The ending radius of the zone in pixels.
fillColor	(Mandatory)	The fill color of the zone.
edgeColor	-1	The edge color of the zone. The default is the same as the fill color.

See also:

- 3.1.4 addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color) 62
- 3.1.5 addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1) 62
- 3.1.6 addZone(startValue as Double, endValue as Double, startRadius as Double, endRadius as Double, fillColor as color, edgeColor as color) 63

3.1.8 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.1.9 `getAxisImageMap(noOfSegments as Integer, mapWidth as Integer, url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates an HTML image map for the axis itself.

Notes: This method is similar to `AngularAxis.getHTMLImageMap`. The difference is instead of generating an image map for the labels, it generates an image map for the axis itself. The axis will be divided into a number of segments, with an image map entry created for each segment.

<code>noOfSegments</code>	(Mandatory)	The number of segments to divide the axis into.
<code>mapWidth</code>	(Mandatory)	The width of the axis used for the purpose of generating the image map.
<code>url</code>	(Mandatory)	The URL to be used in the "href" attribute of the image map. Parameter Substitution and Formatting is supported. Use an empty string if no href attribute is needed.
<code>queryFormat</code>	""	A text string representing the template of the query parameters to be appended to the URL. Parameter Substitution and Formatting is supported. The special keyword "{ default }" represents the default query parameters. This is useful for specifying appending to the default. Note that an empty string means to use the default query parameters. To specify no query parameter, use a space character.
<code>extraAttr</code>	""	A text string to specify additional attributes to add to the <area>tag. Parameter Substitution and Formatting is supported.
<code>offsetX</code>	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
<code>offsetY</code>	0	An offset to be added to all y coordinates in the image map. See <code>offsetX</code> above for description.

Return Value

A text string containing the image map generated.

3.1.10 `getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates an HTML image map for the axis labels.

Notes: This method should be called only after creating the chart image (eg. using `BaseChart.makeChart`). The image map cannot be determined without creating the chart image first.

This method accepts a URL as its argument. When generating an image map, it appends query parameters to the URL to indicate which legend entry the user has clicked.

The following is an example image map generated for an axis with 3 labels.

```
<area shape="rect" coords="30,220,70,239" href="handler.asp?value=0&label=John">
<area shape="rect" coords="70,220,110,239" href="handler.asp?value=1&label=Mary">
<area shape="rect" coords="110,220,150,239" href="handler.asp?value=2&label=Peter">
```

The image map consists of multiple `<area>` tags, one for each label. In the "href" attributes, query parameters are appended to the URL to provide information on the label clicked.

The image map produced by `ChartDirector` does not include the `<map>` and `</map>` tag. This is intentional so that you can add additional custom `<area>` tags to the image map, or append multiple image maps together.

The format of the appended URL parameters is determined using the `queryFormat` argument, which by default is:

```
value= { value } &label= { label }
```

The texts in curly brackets (i.e. `{ value }`, `{ label }`) will be replaced by the actual values when generating the image map. For example, `{ label }` will be replaced by the label text.

Please refer to `Parameter Substitution and Formatting` on all available parameters and how to format them. In addition to customizing the query parameters, `ChartDirector` supports additional HTML attributes in the `<area>` tags using the `extraAttr` argument.

For example, the following `extraAttr` will add a "title" HTML attribute to every `<area>` tag. The "title" attribute will be displayed as "tool tip" when the mouse moves over the image map.

```
title='Click me for details on { label } '
```

Another common usage of the `extraAttr` argument is to add "onmouseover" and "onmouseout" HTML attributes to handle user interaction using Javascript on the browser.

Return Value

A text string containing the image map generated.

3.1.11 setLabelGap(d as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

url	(Mandatory)	The URL to be used in the "href" attribute of the image map. Parameter Substitution and Formatting is supported. Use an empty string if no href attribute is needed.
queryFormat	""	A text string representing the template of the query parameters to be appended to the URL. Parameter Substitution and Formatting is supported. The special keyword " { default } " represents the default query parameters. This is useful for specifying appending to the default. Note that an empty string means to use the default query query parameters. To specify no query parameter, use a space character.
extraAttr	""	A text string to specify additional attributes to add to the <area>tag. Parameter Substitution and Formatting is supported.
offsetX	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
offsetY	0	An offset to be added to all y coordinates in the image map. See offsetX above for description.

Function: Sets the distance between the axis labels and the axis.

Notes: Parameters:

d (Mandatory) The distance between the axis label and axis in pixels.

3.1.12 setLabels(labels() as Double, formatString as string = "") as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the numeric labels to be used on the axis.

Notes: This method is typically used to set the angular axis to enumerated scale. For more details on what is enumerated axis scale, please refer to `AngularAxis.setLabels`. In some cases, it may be desirable to skip some labels. If you want to leave a label position empty, use `kNoValue` as the axis label.

Parameter	Default	Description
labels	(Mandatory)	An array of numbers to be used as the axis labels.
formatString	""	A format string to specified how to format the labels into human readable form. Please refer to <code>Axis.setLabelFormat</code> for the syntax of the format string. An empty string means the format will be automatically determined.

Return Value

A `TextBox` object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See also:

- 3.1.13 `setLabels(labels() as string) as CDTextBoxMBS`

3.1.13 setLabels(labels() as string) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the text labels to be used on the axis.

Notes:

Parameter	Default	Description
lowerLimit	(Mandatory)	The lower bound of the axis.
upperLimit	(Mandatory)	The upper bound of the axis.
labels	(Mandatory)	An array of text strings to be used as the labels on the axis. ChartDirector will distribute the labels evenly on the axis.

This method is typically used in radar charts to set the angular axis to enumerated scale.

In enumerated scale, the labels are laid out evenly on the perimeter of the polar plot area. It is like the labels are at on the vertices of an n-side polygon, where n is the number of labels. Radial grid lines are drawn from the center to the vertices.

The angular coordinates of the data points are determined based on their position in the data array. The first data point will be at the same angular coordinate as the first vertex, the second data point at the second vertex, and so on.

Internally, ChartDirector will assign a value of 0 to the first axis label, 1 to the second axis label, and so on. These values are not visible. Only the axis labels are visible. However, these values may be useful for some ChartDirector features that need to reference the axis position by value, such as adding extra labels using AngularAxis.addLabel.

Parameter	Default	Description
text	(Mandatory)	An array of strings containing the text of the labels.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See also:

- 3.1.12 setLabels(labels() as Double, formatString as string = "") as CDTextBoxMBS 66

3.1.14 setLabelStyle(font as string = "bold", fontsize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the font style used to for the axis labels.

Notes: See Font Specification for details on various font attributes.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of

Parameter	Default	Description
font	"bold"	The font used to draw the labels.
fontSize	10	The font size used to draw the labels in points.
fontColor	TextColor	The color used to draw the labels.
fontAngle	0	The rotation angle of the labels.

the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

3.1.15 setLinearScale(lowerLimit as Double, upperLimit as Double, labels() as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use the given linear scale and the given labels.

Notes:

Parameter	Default	Description
lowerLimit	(Mandatory)	The lower bound of the axis.
upperLimit	(Mandatory)	The upper bound of the axis.
labels	(Mandatory)	An array of text strings to be used as the labels on the axis. ChartDirector will distribute the labels evenly on the axis.

See also:

- 3.1.16 setLinearScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0) 68

3.1.16 setLinearScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use the given linear scale.

Notes:

See also:

- 3.1.15 setLinearScale(lowerLimit as Double, upperLimit as Double, labels() as string) 68

Parameter	Default	Description
lowerLimit	(Mandatory)	The lower bound of the axis.
upperLimit	(Mandatory)	The upper bound of the axis.
majorTickInc	0	Adds major ticks to the axis, where the major ticks are separated by majorTickInc in value. Each major tick will have an associated text label for the value if the axis at the tick. The default value of 0 means no major tick is used.
minorTickInc	0	Adds minor ticks to the axis, where the minor ticks are separated by minorTickInc in value. The default value of 0 means no minor tick is used.

3.2 class CDAngularMeterMBS

3.2.1 class CDAngularMeterMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The AngularMeter class represents angular meters.

Example:

```
// The value to display on the meter
const value = 27.48

// Create an AngularMeter object of size 200 x 115 pixels, with silver background
// color, black border, 2 pixel 3D border border and rounded corners
dim m as new CDAngularMeterMBS(200, 115, CDBaseChartMBS.silverColor, &h000000, 2)
m.setRoundedFrame

// Set the meter center at (100, 100), with radius 85 pixels, and span from -90
// to +90 degrees (semi-circle)
m.setMeter(100, 100, 85, -90, 90)

// Meter scale is 0 - 100, with major tick every 20 units, minor tick every 10
// units, and micro tick every 5 units
m.setScale(0, 100, 20, 10, 5)

// Set 0 - 60 as green (66FF66) zone
m.addZone(0, 60, 0, 85, &h66ff66)

// Set 60 - 80 as yellow (FFFF33) zone
m.addZone(60, 80, 0, 85, &hffff33)

// Set 80 - 100 as red (FF6666) zone
m.addZone(80, 100, 0, 85, &hff6666)

// Add a text label centered at (100, 60) with 12 pts Arial Bold font
call m.addText(100, 60, "PSI", "arialbd.ttf", 12, CDBaseChartMBS.kTextColor, CDBaseChartMBS.kCenter)

// Add a text box at the top right corner of the meter showing the value
// formatted to 2 decimal places, using white text on a black background, and
// with 1 pixel 3D depressed border
m.addText(156, 8, m.formatValue(value, "2"), "arial.ttf", 8, &hffffff).setBackground(&h000000, 0, -1)

// Add a semi-transparent blue (40666699) pointer with black border at the
// specified value
call m.addPointer(value, &h40666699, &h000000)

// Output the chart
```

Backdrop=m.makeChartPicture

Notes: Subclass of the CDBaseMeterMBS class.

Blog Entries

- [ChartDirector Meters and Gauges](#)
- [MBS Xojo Plugins, version 20.4pr2](#)
- [Chart Diagrams with Xojo](#)
- [MBS Xojo / Real Studio Plugins, version 15.2pr4](#)
- [Dock Meter Example](#)

Xojo Developer Magazine

- [7.4, page 35: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [21.6, pages 81 to 82: From 0 to 100, Creating Gauges with MBS and ChartDirector by Stefanie Juchmes](#)

3.2.2 Methods

3.2.3 addGlare

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds glare effect to the meter.

Notes: The glare effect is created by brightening part of the meter face.

The face is divided by an arc, of which one part is brightened. The glare effect works best for meters with dark background.

All parameters are optional.

See also:

- [3.2.4 addGlare\(radius as Double, span as Double = 135, rotate as Double = 0.0\)](#) 71
- [3.2.5 addGlare\(radius as Double, span as Double, rotate as Double, glareRadius as Double, intensity as Double = 0.13\)](#) 73

3.2.4 addGlare(radius as Double, span as Double = 135, rotate as Double = 0.0)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Argument	Default	Description
radius	NoValue	The radius of the face of the meter. The default is the meter scale radius as set by <code>AngularMeter.setMeter</code> .
span	135	The angular span of the arc that divides the face into two parts. The dividing arc intersects with the face border at two points. The angular span is defined as center angle of these two points (that is, the angle subtended at the meter center by these two points).
rotate	0	By default, the glare will be symmetrically with respect to the vertical axis. This argument can be used to rotate the glare clockwise in degrees.
glareRadius	NoValue	The radius of the arc that divides the face into two parts. A positive value means the arc will result in a convex glare, while a negative means concave glare. The default is equal to 1.5 times of the face radius.
intensity	0.13	The intensity of the glare, which must be from 0 to 1. A value of 0 means no glare, while a value of 1 means the glare will be 100% white.

Function: Adds glare effect to the meter.

Notes: The glare effect is created by brightening part of the meter face.

The face is divided by an arc, of which one part is brightened. The glare effect works best for meters with dark background.

All parameters are optional.

Argument	Default	Description
radius	NoValue	The radius of the face of the meter. The default is the meter scale radius as set by <code>AngularMeter.setMeter</code> .
span	135	The angular span of the arc that divides the face into two parts. The dividing arc intersects with the face border at two points. The angular span is defined as center angle of these two points (that is, the angle subtended at the meter center by these two points).
rotate	0	By default, the glare will be symmetrically with respect to the vertical axis. This argument can be used to rotate the glare clockwise in degrees.
glareRadius	NoValue	The radius of the arc that divides the face into two parts. A positive value means the arc will result in a convex glare, while a negative means concave glare. The default is equal to 1.5 times of the face radius.
intensity	0.13	The intensity of the glare, which must be from 0 to 1. A value of 0 means no glare, while a value of 1 means the glare will be 100% white.

See also:

- 3.2.3 `addGlare` 71
- 3.2.5 `addGlare(radius as Double, span as Double, rotate as Double, glareRadius as Double, intensity as Double = 0.13)` 73

3.2.5 addGlare(radius as Double, span as Double, rotate as Double, glareRadius as Double, intensity as Double = 0.13)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds glare effect to the meter.

Notes: The glare effect is created by brightening part of the meter face.

The face is divided by an arc, of which one part is brightened. The glare effect works best for meters with dark background.

All parameters are optional.

Argument	Default	Description
radius	NoValue	The radius of the face of the meter. The default is the meter scale radius as set by <code>AngularMeter.setMeter</code> .
span	135	The angular span of the arc that divides the face into two parts. The dividing arc intersects with the face border at two points. The angular span is defined as center angle of these two points (that is, the angle subtended at the meter center by these two points).
rotate	0	By default, the glare will be symmetrically with respect to the vertical axis. This argument can be used to rotate the glare clockwise in degrees.
glareRadius	NoValue	The radius of the arc that divides the face into two parts. A positive value means the arc will result in a convex glare, while a negative means concave glare. The default is equal to 1.5 times of the face radius.
intensity	0.13	The intensity of the glare, which must be from 0 to 1. A value of 0 means no glare, while a value of 1 means the glare will be 100% white.

See also:

- 3.2.3 addGlare 71
- 3.2.4 addGlare(radius as Double, span as Double = 135, rotate as Double = 0.0) 71

3.2.6 addPointer(value as Double, fillColor as color, edgeColor as color = &cFFFFFFFFF, pointerType as Integer = 6) as CDMeterPointerMBS

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a new style pointer to the meter.

Notes: When compared to classical pointers (added by `CDBaseMeterMBS.addPointer`), new style pointers are thinner. They are designed to be used with the new style meter cap (see `CDAngularMeterMBS.setCap2`).

They are also designed to have their base and tip movable in the radial direction, which means they can be detached from the center, and can even points from the outside inwards.

Parameters are optional and colors can be passed as Integer or color values.

Return a `MeterPointer` object representing the pointer. You may use this object to fine-tune the appearance

Argument	Default	Description
value	(Mandatory)	The value that the pointer points to.
fillColor	(Mandatory)	The fill color of the pointer.
edgeColor	-1	The edge color of the pointer. -1 means the edge color is the same as the fill color.
pointerType	TriangularPointer2	The pointer shape. Please refer to <code>MeterPointer.setShapeAndOffset</code> for the available shapes.
startOffset	NoValue	The distance between the starting point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
endOffset	NoValue	The distance between the ending point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

of the pointer.

See also:

- 3.2.7 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS` 74
- 3.2.8 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS` 76
- 3.2.9 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS` 77
- 3.2.10 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer = -1, pointerType as Integer = 6) as CDMeterPointerMBS` 78
- 3.2.11 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS` 79
- 3.2.12 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS` 80
- 3.2.13 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS` 81

3.2.7 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a new style pointer to the meter.

Notes: When compared to classical pointers (added by `CDBaseMeterMBS.addPointer`), new style pointers

are thinner. They are designed to be used with the new style meter cap (see `CDAngularMeterMBS.setCap2`). They are also designed to have their base and tip movable in the radial direction, which means they can be detached from the center, and can even points from the outside inwards.

Parameters are optional and colors can be passed as Integer or color values.

Argument	Default	Description
value	(Mandatory)	The value that the pointer points to.
fillColor	(Mandatory)	The fill color of the pointer.
edgeColor	-1	The edge color of the pointer. -1 means the edge color is the same as the fill color.
pointerType	TriangularPointer2	The pointer shape. Please refer to <code>MeterPointer.setShapeAndOffset</code> for the available shapes.
startOffset	NoValue	The distance between the starting point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
endOffset	NoValue	The distance between the ending point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

Return a `MeterPointer` object representing the pointer. You may use this object to fine-tune the appearance of the pointer.

See also:

- 3.2.6 `addPointer(value as Double, fillColor as color, edgeColor as color = &cFFFFFFFF, pointerType as Integer = 6) as CDMeterPointerMBS` 73
- 3.2.8 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS` 76
- 3.2.9 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS` 77
- 3.2.10 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer = -1, pointerType as Integer = 6) as CDMeterPointerMBS` 78
- 3.2.11 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS` 79
- 3.2.12 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS` 80
- 3.2.13 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS` 81

3.2.8 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a new style pointer to the meter.

Notes: When compared to classical pointers (added by `CDBaseMeterMBS.addPointer`), new style pointers are thinner. They are designed to be used with the new style meter cap (see `CDAngularMeterMBS.setCap2`). They are also designed to have their base and tip movable in the radial direction, which means they can be detached from the center, and can even point from the outside inwards.

Parameters are optional and colors can be passed as Integer or color values.

Argument	Default	Description
value	(Mandatory)	The value that the pointer points to.
fillColor	(Mandatory)	The fill color of the pointer.
edgeColor	-1	The edge color of the pointer. -1 means the edge color is the same as the fill color.
pointerType	TriangularPointer2	The pointer shape. Please refer to <code>MeterPointer.setShapeAndOffset</code> for the available shapes.
startOffset	NoValue	The distance between the starting point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
endOffset	NoValue	The distance between the ending point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

Return a `MeterPointer` object representing the pointer. You may use this object to fine-tune the appearance of the pointer.

See also:

- 3.2.6 `addPointer(value as Double, fillColor as color, edgeColor as color = &cFFFFFFFF, pointerType as Integer = 6) as CDMeterPointerMBS` 73
- 3.2.7 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS` 74
- 3.2.9 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS` 77
- 3.2.10 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer = -1, pointerType as Integer = 6) as CDMeterPointerMBS` 78
- 3.2.11 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS` 79

- 3.2.12 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double)` as `CDMeterPointerMBS` 80
- 3.2.13 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` as `CDMeterPointerMBS` 81

3.2.9 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` as `CDMeterPointerMBS`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a new style pointer to the meter.

Notes: When compared to classical pointers (added by `CDBaseMeterMBS.addPointer`), new style pointers are thinner. They are designed to be used with the new style meter cap (see `CDAngularMeterMBS.setCap2`). They are also designed to have their base and tip movable in the radial direction, which means they can be detached from the center, and can even point from the outside inwards.

Parameters are optional and colors can be passed as Integer or color values.

Argument	Default	Description
<code>value</code>	(Mandatory)	The value that the pointer points to.
<code>fillColor</code>	(Mandatory)	The fill color of the pointer.
<code>edgeColor</code>	-1	The edge color of the pointer. -1 means the edge color is the same as the fill color.
<code>pointerType</code>	<code>TriangularPointer2</code>	The pointer shape. Please refer to <code>MeterPointer.setShapeAndOffset</code> for the available shapes.
<code>startOffset</code>	<code>NoValue</code>	The distance between the starting point of the pointer with the meter center, expressed as a ratio to the scale radius. <code>NoValue</code> means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
<code>endOffset</code>	<code>NoValue</code>	The distance between the ending point of the pointer with the meter center, expressed as a ratio to the scale radius. <code>NoValue</code> means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
<code>widthRatio</code>	<code>NoValue</code>	The width of the pointer relative to the default width. <code>NoValue</code> means the width is automatically determined.

Return a `MeterPointer` object representing the pointer. You may use this object to fine-tune the appearance of the pointer.

See also:

- 3.2.6 `addPointer(value as Double, fillColor as color, edgeColor as color = &cFFFFFFFF, pointerType as Integer = 6)` as `CDMeterPointerMBS` 73
- 3.2.7 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double)` as `CDMeterPointerMBS` 74

- 3.2.8 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double)` as `CDMeterPointerMBS` 76
- 3.2.10 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer = -1, pointerType as Integer = 6)` as `CDMeterPointerMBS` 78
- 3.2.11 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double)` as `CDMeterPointerMBS` 79
- 3.2.12 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double)` as `CDMeterPointerMBS` 80
- 3.2.13 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` as `CDMeterPointerMBS` 81

3.2.10 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer = -1, pointerType as Integer = 6)` as `CDMeterPointerMBS`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a new style pointer to the meter.

Notes: When compared to classical pointers (added by `CDBaseMeterMBS.addPointer`), new style pointers are thinner. They are designed to be used with the new style meter cap (see `CDAngularMeterMBS.setCap2`). They are also designed to have their base and tip movable in the radial direction, which means they can be detached from the center, and can even point from the outside inwards.

Parameters are optional and colors can be passed as `Integer` or `color` values.

Argument	Default	Description
<code>value</code>	(Mandatory)	The value that the pointer points to.
<code>fillColor</code>	(Mandatory)	The fill color of the pointer.
<code>edgeColor</code>	-1	The edge color of the pointer. -1 means the edge color is the same as the fill color.
<code>pointerType</code>	<code>TriangularPointer2</code>	The pointer shape. Please refer to <code>MeterPointer.setShapeAndOffset</code> for the available shapes.
<code>startOffset</code>	<code>NoValue</code>	The distance between the starting point of the pointer with the meter center, expressed as a ratio to the scale radius. <code>NoValue</code> means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
<code>endOffset</code>	<code>NoValue</code>	The distance between the ending point of the pointer with the meter center, expressed as a ratio to the scale radius. <code>NoValue</code> means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
<code>widthRatio</code>	<code>NoValue</code>	The width of the pointer relative to the default width. <code>NoValue</code> means the width is automatically determined.

Return a `MeterPointer` object representing the pointer. You may use this object to fine-tune the appearance of the pointer.

See also:

3.2. CLASS CDANGULARMETERMBS	79
• 3.2.6 addPointer(value as Double, fillColor as color, edgeColor as color = &cFFFFFFFF, pointerType as Integer = 6) as CDMeterPointerMBS	73
• 3.2.7 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS	74
• 3.2.8 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS	76
• 3.2.9 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS	77
• 3.2.11 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS	79
• 3.2.12 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS	80
• 3.2.13 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS	81

3.2.11 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a new style pointer to the meter.

Notes: When compared to classical pointers (added by CDBaseMeterMBS.addPointer), new style pointers are thinner. They are designed to be used with the new style meter cap (see CDAngularMeterMBS.setCap2). They are also designed to have their base and tip movable in the radial direction, which means they can be detached from the center, and can even point from the outside inwards.

Parameters are optional and colors can be passed as Integer or color values.

Return a MeterPointer object representing the pointer. You may use this object to fine-tune the appearance of the pointer.

See also:

• 3.2.6 addPointer(value as Double, fillColor as color, edgeColor as color = &cFFFFFFFF, pointerType as Integer = 6) as CDMeterPointerMBS	73
• 3.2.7 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS	74
• 3.2.8 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS	76
• 3.2.9 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS	77

Argument	Default	Description
value	(Mandatory)	The value that the pointer points to.
fillColor	(Mandatory)	The fill color of the pointer.
edgeColor	-1	The edge color of the pointer. -1 means the edge color is the same as the fill color.
pointerType	TriangularPointer2	The pointer shape. Please refer to <code>MeterPointer.setShapeAndOffset</code> for the available shapes.
startOffset	NoValue	The distance between the starting point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
endOffset	NoValue	The distance between the ending point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

- 3.2.10 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer = -1, pointerType as Integer = 6)` as `CDMeterPointerMBS` 78
- 3.2.12 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double)` as `CDMeterPointerMBS` 80
- 3.2.13 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` as `CDMeterPointerMBS` 81

3.2.12 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double)` as `CDMeterPointerMBS`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a new style pointer to the meter.

Notes: When compared to classical pointers (added by `CDBaseMeterMBS.addPointer`), new style pointers are thinner. They are designed to be used with the new style meter cap (see `CDAngularMeterMBS.setCap2`). They are also designed to have their base and tip movable in the radial direction, which means they can be detached from the center, and can even point from the outside inwards.

Parameters are optional and colors can be passed as Integer or color values.

Return a `MeterPointer` object representing the pointer. You may use this object to fine-tune the appearance of the pointer.

See also:

- 3.2.6 `addPointer(value as Double, fillColor as color, edgeColor as color = &cFFFFFFFF, pointerType as Integer = 6)` as `CDMeterPointerMBS` 73

Argument	Default	Description
value	(Mandatory)	The value that the pointer points to.
fillColor	(Mandatory)	The fill color of the pointer.
edgeColor	-1	The edge color of the pointer. -1 means the edge color is the same as the fill color.
pointerType	TriangularPointer2	The pointer shape. Please refer to MeterPointer.setShapeAndOffset for the available shapes.
startOffset	NoValue	The distance between the starting point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to MeterPointer.setShapeAndOffset for more information about this argument.
endOffset	NoValue	The distance between the ending point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to MeterPointer.setShapeAndOffset for more information about this argument.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

- 3.2.7 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS 74
- 3.2.8 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS 76
- 3.2.9 addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS 77
- 3.2.10 addPointer(value as Double, fillColor as Integer, edgeColor as Integer = -1, pointerType as Integer = 6) as CDMeterPointerMBS 78
- 3.2.11 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS 79
- 3.2.13 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS 81

3.2.13 addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a new style pointer to the meter.

Notes: When compared to classical pointers (added by CDBaseMeterMBS.addPointer), new style pointers are thinner. They are designed to be used with the new style meter cap (see CDAngularMeterMBS.setCap2). They are also designed to have their base and tip movable in the radial direction, which means they can be detached from the center, and can even point from the outside inwards.

Parameters are optional and colors can be passed as Integer or color values.

Argument	Default	Description
value	(Mandatory)	The value that the pointer points to.
fillColor	(Mandatory)	The fill color of the pointer.
edgeColor	-1	The edge color of the pointer. -1 means the edge color is the same as the fill color.
pointerType	TriangularPointer2	The pointer shape. Please refer to <code>MeterPointer.setShapeAndOffset</code> for the available shapes.
startOffset	NoValue	The distance between the starting point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
endOffset	NoValue	The distance between the ending point of the pointer with the meter center, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined. Please refer to <code>MeterPointer.setShapeAndOffset</code> for more information about this argument.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

Return a `MeterPointer` object representing the pointer. You may use this object to fine-tune the appearance of the pointer.

See also:

- 3.2.6 `addPointer(value as Double, fillColor as color, edgeColor as color = &cFFFFFFFF, pointerType as Integer = 6) as CDMeterPointerMBS` 73
- 3.2.7 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS` 74
- 3.2.8 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS` 76
- 3.2.9 `addPointer(value as Double, fillColor as color, edgeColor as color, pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) as CDMeterPointerMBS` 77
- 3.2.10 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer = -1, pointerType as Integer = 6) as CDMeterPointerMBS` 78
- 3.2.11 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double) as CDMeterPointerMBS` 79
- 3.2.12 `addPointer(value as Double, fillColor as Integer, edgeColor as Integer, pointerType as Integer, startOffset as Double, endOffset as Double) as CDMeterPointerMBS` 80

3.2.14 `addRing(startRadius as Integer, endRadius as Integer, fillColor as color, edgeColor as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addRing method, but uses color instead of integer data type for passing color values.

See also:

- 3.2.15 addRing(startRadius as Integer, endRadius as Integer, fillColor as Integer, edgeColor as Integer = -1) 83

3.2.15 addRing(startRadius as Integer, endRadius as Integer, fillColor as Integer, edgeColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a circular ring or a circle to the meter.

Notes: A ring is the region between two concentric circles. This method is most often used for adding circular borders and backgrounds for the meter, and for some other decorative purposes. For example, one can add a circle with a metallic background color as the 'face' of the meter meter using this method.

Parameter	Default	Description
startRadius	(Mandatory)	The starting radius of the ring in pixels. To add a circle, set the start radius to 0.
endRadius	(Mandatory)	The ending radius of the ring in pixels.
fillColor	(Mandatory)	The fill color of the ring.
edgeColor	-1	The edge color of the ring. The default is the same as the fill color.

See also:

- 3.2.14 addRing(startRadius as Integer, endRadius as Integer, fillColor as color, edgeColor as color) 82

3.2.16 addRingSector(startRadius as Integer, endRadius as Integer, a1 as Double, a2 as Double, fillColor as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addRingSector method, but uses color instead of integer data type for passing color values.

See also:

- 3.2.17 addRingSector(startRadius as Integer, endRadius as Integer, a1 as Double, a2 as Double, fillColor as Integer, edgeColor as Integer = -1) 83

3.2.17 addRingSector(startRadius as Integer, endRadius as Integer, a1 as Double, a2 as Double, fillColor as Integer, edgeColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a ring sector or a sector to the meter.

Notes: This method is similar to the `AngularMeter.addRing` method, except that the ring it adds does not span the complete circle, but just part of the circle.

Parameter	Default	Description
<code>startRadius</code>	(Mandatory)	The starting radius of the ring sector in pixels. To add a sector, set the starting radius to 0.
<code>endRadius</code>	(Mandatory)	The ending radius of the ring sector in pixels.
<code>a1</code>	(Mandatory)	The starting angle of the ring sector in degrees. The angle is measured in degrees clockwise, with 0 being the upward pointing direction.
<code>a2</code>	(Mandatory)	The ending angle of the ring sector in degrees. The angle is measured in degrees clockwise, with 0 being the upward pointing direction.
<code>fillColor</code>	(Mandatory)	The fill color of the ring sector.
<code>edgeColor</code>	-1	The edge color of the ring sector. The default is the same as the fill color.

See also:

- 3.2.16 `addRingSector(startRadius as Integer, endRadius as Integer, a1 as Double, a2 as Double, fillColor as color, edgeColor as color)` 83

3.2.18 `addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer = 0, edgeColor as color = &cFFFFFFFF, scaleRadius as Integer = -2147483647)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a background to the meter scale.

Notes: The scale background is a circle segment (a circle with a part cut-off by a chord). Its radius is usually configured as larger than that of the meter scale. In this case, its angular span will also be larger than that of the meter scale. In the above sample, the meter scale spans 180 degrees. The scale background spans more than 180 degrees so that it can maintain a margin at the bottom side (the chord side).

Most arguments are optional and colors can be passed as Integer or color value.

See also:

- 3.2.19 `addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer, edgeColor as color, scaleRadius as Integer, startAngle as Double, endAngle as Double)` 85
- 3.2.20 `addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer = 0, edgeColor as Integer = -1, scaleRadius as Integer = -2147483647)` 86
- 3.2.21 `addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer, edgeColor as Integer, scaleRadius as Integer, startAngle as Double, endAngle as Double)` 87

Argument	Default	Description
bgRadius	(Mandatory)	The radius of the circle segment to be used as the scale background.
fillColor	(Mandatory)	The fill color of the circle segment (the background color).
edgeWidth	0	The edge width of the circle segment. A positive width means the edge is internal to the circle segment. A negative width means the edge is external to the circle segment.
edgeColor	-1	The edge color. The default value of -1 means the edge color is the same as the fill color.
scaleRadius	-0x7ffffff	The radius of the meter scale. ChartDirector uses this value to compute the extra angular span the circle segment must have in order to maintain proper margin at the chord side. This argument is usually not necessary as ChartDirector already know the scale radius from the meter configuration. This argument can be used if you would like to use a "fake" scale radius to draw the scale background for special effects.
startAngle	NoValue	The start angle of the meter scale. ChartDirector uses this value to compute start angle of the circle segment. This argument is usually not necessary as ChartDirector already know the start angle from the meter configuration. This argument can be used if you would like to use a "fake" angle to draw the scale background for special effects.
endAngle	NoValue	The end angle of the meter scale. ChartDirector uses this value to compute end angle of the circle segment. This argument is usually not necessary as ChartDirector already know the end angle from the meter configuration. This argument can be used if you would like to use a "fake" angle to draw the scale background for special effects.

3.2.19 addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer, edgeColor as color, scaleRadius as Integer, startAngle as Double, endAngle as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a background to the meter scale.

Notes: The scale background is a circle segment (a circle with a part cut-off by a chord). Its radius is usually configured as larger than that of the meter scale. In this case, its angular span will also be larger than that of the meter scale. In the above sample, the meter scale spans 180 degrees. The scale background spans more than 180 degrees so that it can maintain a margin at the bottom side (the chord side).

Most arguments are optional and colors can be passed as Integer or color value.

See also:

- 3.2.18 addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer = 0, edgeColor as color = &cFFFFFFFF, scaleRadius as Integer = -2147483647) 84
- 3.2.20 addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer = 0, edgeColor as Integer = -1, scaleRadius as Integer = -2147483647) 86
- 3.2.21 addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer, edgeColor

Argument	Default	Description
bgRadius	(Mandatory)	The radius of the circle segment to be used as the scale background.
fillColor	(Mandatory)	The fill color of the circle segment (the background color).
edgeWidth	0	The edge width of the circle segment. A positive width means the edge is internal to the circle segment. A negative width means the edge is external to the circle segment.
edgeColor	-1	The edge color. The default value of -1 means the edge color is the same as the fill color.
scaleRadius	-0x7ffffff	The radius of the meter scale. ChartDirector uses this value to compute the extra angular span the circle segment must have in order to maintain proper margin at the chord side. This argument is usually not necessary as ChartDirector already know the scale radius from the meter configuration. This argument can be used if you would like to use a "fake" scale radius to draw the scale background for special effects.
startAngle	NoValue	The start angle of the meter scale. ChartDirector uses this value to compute start angle of the circle segment. This argument is usually not necessary as ChartDirector already know the start angle from the meter configuration. This argument can be used if you would like to use a "fake" angle to draw the scale background for special effects.
endAngle	NoValue	The end angle of the meter scale. ChartDirector uses this value to compute end angle of the circle segment. This argument is usually not necessary as ChartDirector already know the end angle from the meter configuration. This argument can be used if you would like to use a "fake" angle to draw the scale background for special effects.

as Integer, scaleRadius as Integer, startAngle as Double, endAngle as Double)

87

3.2.20 addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer = 0, edgeColor as Integer = -1, scaleRadius as Integer = -2147483647)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a background to the meter scale.

Notes: The scale background is a circle segment (a circle with a part cut-off by a chord). Its radius is usually configured as larger than that of the meter scale. In this case, its angular span will also be larger than that of the meter scale. In the above sample, the meter scale spans 180 degrees. The scale background spans more than 180 degrees so that it can maintain a margin at the bottom side (the chord side).

Most arguments are optional and colors can be passed as Integer or color value.

See also:

- 3.2.18 addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer = 0, edgeColor as color = &cFFFFFFF, scaleRadius as Integer = -2147483647) 84
- 3.2.19 addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer, edgeColor as

Argument	Default	Description
bgRadius	(Mandatory)	The radius of the circle segment to be used as the scale background.
fillColor	(Mandatory)	The fill color of the circle segment (the background color).
edgeWidth	0	The edge width of the circle segment. A positive width means the edge is internal to the circle segment. A negative width means the edge is external to the circle segment.
edgeColor	-1	The edge color. The default value of -1 means the edge color is the same as the fill color.
scaleRadius	-0x7ffffff	The radius of the meter scale. ChartDirector uses this value to compute the extra angular span the circle segment must have in order to maintain proper margin at the chord side. This argument is usually not necessary as ChartDirector already know the scale radius from the meter configuration. This argument can be used if you would like to use a "fake" scale radius to draw the scale background for special effects.
startAngle	NoValue	The start angle of the meter scale. ChartDirector uses this value to compute start angle of the circle segment. This argument is usually not necessary as ChartDirector already know the start angle from the meter configuration. This argument can be used if you would like to use a "fake" angle to draw the scale background for special effects.
endAngle	NoValue	The end angle of the meter scale. ChartDirector uses this value to compute end angle of the circle segment. This argument is usually not necessary as ChartDirector already know the end angle from the meter configuration. This argument can be used if you would like to use a "fake" angle to draw the scale background for special effects.

color, scaleRadius as Integer, startAngle as Double, endAngle as Double)

85

- 3.2.21 addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer, edgeColor as Integer, scaleRadius as Integer, startAngle as Double, endAngle as Double) 87

3.2.21 addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer, edgeColor as Integer, scaleRadius as Integer, startAngle as Double, endAngle as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a background to the meter scale.

Notes: The scale background is a circle segment (a circle with a part cut-off by a chord). Its radius is usually configured as larger than that of the meter scale. In this case, its angular span will also be larger than that of the meter scale. In the above sample, the meter scale spans 180 degrees. The scale background spans more than 180 degrees so that it can maintain a margin at the bottom side (the chord side).

Most arguments are optional and colors can be passed as Integer or color value.

See also:

- 3.2.18 addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer = 0, edgeColor

Argument	Default	Description
bgRadius	(Mandatory)	The radius of the circle segment to be used as the scale background.
fillColor	(Mandatory)	The fill color of the circle segment (the background color).
edgeWidth	0	The edge width of the circle segment. A positive width means the edge is internal to the circle segment. A negative width means the edge is external to the circle segment.
edgeColor	-1	The edge color. The default value of -1 means the edge color is the same as the fill color.
scaleRadius	-0x7ffffff	The radius of the meter scale. ChartDirector uses this value to compute the extra angular span the circle segment must have in order to maintain proper margin at the chord side. This argument is usually not necessary as ChartDirector already know the scale radius from the meter configuration. This argument can be used if you would like to use a "fake" scale radius to draw the scale background for special effects.
startAngle	NoValue	The start angle of the meter scale. ChartDirector uses this value to compute start angle of the circle segment. This argument is usually not necessary as ChartDirector already know the start angle from the meter configuration. This argument can be used if you would like to use a "fake" angle to draw the scale background for special effects.
endAngle	NoValue	The end angle of the meter scale. ChartDirector uses this value to compute end angle of the circle segment. This argument is usually not necessary as ChartDirector already know the end angle from the meter configuration. This argument can be used if you would like to use a "fake" angle to draw the scale background for special effects.

as color = &cFFFFFFFF, scaleRadius as Integer = -2147483647) 84

- 3.2.19 addScaleBackground(bgRadius as Integer, fillColor as color, edgeWidth as Integer, edgeColor as color, scaleRadius as Integer, startAngle as Double, endAngle as Double) 85
- 3.2.20 addScaleBackground(bgRadius as Integer, fillColor as Integer, edgeWidth as Integer = 0, edgeColor as Integer = -1, scaleRadius as Integer = -2147483647) 86

3.2.22 addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addZone method, but uses color instead of integer data type for passing color values.

See also:

- 3.2.23 addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1) 89
- 3.2.24 addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as color, edgeColor as color) 89

- 3.2.25 addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as Integer, edgeColor as Integer = -1) 90

3.2.23 addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a colored zone to the meter.

Notes: This method is just a short cut to AngularMeter.addZone, in which the starting radius is always 0, and ending radius is the radius of the meter scale. In other words, the zone is a sector on the meter.

Parameter	Default	Description
startValue	(Mandatory)	The data value that marks the start position of the zone.
endValue	(Mandatory)	The data value that marks the end position of the zone.
fillColor	(Mandatory)	The fill color of the zone.
edgeColor	-1	The edge color of the zone. The default is the same as the fill color.

See also:

- 3.2.22 addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color) 88
- 3.2.24 addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as color, edgeColor as color) 89
- 3.2.25 addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as Integer, edgeColor as Integer = -1) 90

3.2.24 addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addZone method, but uses color instead of integer data type for passing color values.

See also:

- 3.2.22 addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color) 88
- 3.2.23 addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1) 89
- 3.2.25 addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as Integer, edgeColor as Integer = -1) 90

3.2.25 `addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as Integer, edgeColor as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a colored zone to the meter.

Notes:

Parameter	Default	Description
<code>startValue</code>	(Mandatory)	The data value that marks the start position of the zone.
<code>endValue</code>	(Mandatory)	The data value that marks the end position of the zone.
<code>startRadius</code>	(Mandatory)	The starting radius of the zone in pixels.
<code>endRadius</code>	(Mandatory)	The ending radius of the zone in pixels.
<code>fillColor</code>	(Mandatory)	The fill color of the zone.
<code>edgeColor</code>	-1	The edge color of the zone. The default is the same as the fill color.

See also:

- 3.2.22 `addZone(startValue as Double, endValue as Double, fillColor as color, edgeColor as color)` 88
- 3.2.23 `addZone(startValue as Double, endValue as Double, fillColor as Integer, edgeColor as Integer = -1)` 89
- 3.2.24 `addZone(startValue as Double, endValue as Double, startRadius as Integer, endRadius as Integer, fillColor as color, edgeColor as color)` 89

3.2.26 `Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `Constructor` method, but uses color instead of integer data type for passing color values.

See also:

- 3.2.27 `Constructor(width as Integer, height as Integer, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)` 90

3.2.27 `Constructor(width as Integer, height as Integer, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Parameter	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	BackgroundColor	The background color of the chart.
edgeColor	Transparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

Function: Creates a new AngularMeter object.

Notes:

See also:

- 3.2.26 Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0) 90

3.2.28 relativeLinearGradient(gradient() as Double, angle as Double = 0.0, radius as Double = -1.0) as Integer

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a linear gradient color relative to the meter center and radius.

Notes: A general linear gradient color can be created by CDBaseChartMBS.linearGradientColor, which involves specifying the x and y coordinates of the starting and ending points and the color stops. The relativeLinearGradient simplifies creating a linear gradient by assuming the gradient line passes through the meter center at an configurable angle.

The linear gradient is defined using an array of numbers, in which each pair of numbers represents the relative distance and its associated color. The relative distance is the ratio of the absolute distance to the reference radius, which defaults to the meter scale radius (set by AngularMeter.setMeter).

For example, to define a linear gradient with blue (0000FF) at the bottom-left of reference radius, green (00FF00) at the center, and red (FF0000) at the top-right of the reference radius, the angle should be set to 45 degrees, and the array of numbers should be:

```
-1.0, &H0000ff, 0.0, &H00ff00, 1.0, &Hff0000
```

The relative distances in the array should be arranged in increasing order. It is possible to define a relative linear gradient shorter or longer than the reference radius by using a relative distance of magnitude smaller or greater than 1. A negative relative distance means the distance is measured at the opposite side of center.

Returns a 32-bit integer representing the linear gradient color.

Argument	Default	Description
gradient	(Mandatory)	An numeric array defining the color stops. Please refer to the description above for details.
angle	0	The direction of the linear gradient line. It is specified as a clockwise angle in degrees, with 0 being the upward pointing direction.
radius	-1	The reference radius to define the relative distance. The default is the meter scale radius (set by <code>CDAngularMeterMBS.setMeter</code>).

3.2.29 `relativeRadialGradient(gradient() as Double, radius as Double = -1.0) as Integer`

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a radial gradient color relative to the meter center and radius.

Notes: A general radial gradient color can be created by `CDBaseChartMBS.radialGradientColor`, which involves specifying the center x and y coordinates, horizontal and vertical radii, and the color stops. The `relativeRadialGradient` simplifies creating a radial gradient by assuming the center of the radial gradient to be the meter center, and the radial pattern is circular (so that the vertical and horizontal radii are the same), and default to using the meter scale radius (set by `AngularMeter.setMeter`) as the reference radius.

The radial gradient is defined using an array of numbers, in which each pair of numbers represents the relative radius and its associated color. The relative radius is the ratio of the absolute radius to the reference radius.

For example, to define a radial gradient with blue (0000FF) at the center, green (00FF00) at the mid-point between the center and the reference radius, and red at the reference radius, the numbers should be:

```
0.0, &H0000ff, 0.5, &H00ff00, 1.0, &Hff0000
```

The relative radii in the array must be arranged in increasing order. It is possible to define a relative radial gradient smaller or larger than the reference radius by using a relative radius smaller or greater than 1.

Argument	Default	Description
gradient	(Mandatory)	An numeric array defining the color stops. Please refer to the description above for details.
radius	-1	The reference radius of the radial gradient. The default is the meter scale radius (set by <code>CDAngularMeterMBS.setMeter</code>).

Returns a 32-bit integer representing the radial gradient color.

3.2.30 setCap(radius as Integer, fillColor as color, edgeColor as color)

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Set the radius and color of the pointer cap.

Notes: The pointer cap is a circle at the center of the meter above the meter pointer. By default, it is a small circle 3 pixels in radius, so it looks like a "pivot" for the pointer for it to rotate about the center.

You may change the radius and colors of this circle using this method to achieve other visual effects.

Parameter	Default	Description
radius	(Mandatory)	The radius of pointer cap circle in pixels.
fillColor	(Mandatory)	The fill color of the pointer cap circle.
edgeColor	LineColor	The edge color of the pointer cap circle.

See also:

- 3.2.31 setCap(radius as Integer, fillColor as Integer, edgeColor as Integer = &hfff0001) 93

3.2.31 setCap(radius as Integer, fillColor as Integer, edgeColor as Integer = &hfff0001)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Set the radius and color of the pointer cap.

Notes: The pointer cap is a circle at the center of the meter above the meter pointer. By default, it is a small circle 3 pixels in radius, so it looks like a "pivot" for the pointer for it to rotate about the center.

You may change the radius and colors of this circle using this method to achieve other visual effects.

Parameter	Default	Description
radius	(Mandatory)	The radius of pointer cap circle in pixels.
fillColor	(Mandatory)	The fill color of the pointer cap circle.
edgeColor	LineColor	The edge color of the pointer cap circle.

See also:

- 3.2.30 setCap(radius as Integer, fillColor as color, edgeColor as color) 93

3.2.32 `setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. It is designed to be used with new style pointers (added with `AngularMeter.addPointer2`). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

Argument	Default	Description
<code>backColor</code>	<code>0x888888</code>	The fill color of the back circle.
<code>frontColor</code>	<code>0x000000</code>	The fill color of the front circle.
<code>frontEdgeColor</code>	<code>0x888888</code>	The edge color of the front circle.
<code>lightingRatio</code>	<code>NoValue</code>	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
<code>backRadiusRatio</code>	<code>NoValue</code>	The radius of the back circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.15.
<code>frontRadiusRatio</code>	<code>NoValue</code>	The radius of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.11.
<code>frontEdgeWidthRatio</code>	<code>NoValue</code>	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.01.

See also:

- 3.2.33 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double)` 95
- 3.2.34 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double)` 96
- 3.2.35 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)` 97
- 3.2.36 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)` 98
- 3.2.37 `setCap2(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888)` 100
- 3.2.38 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)` 101

3.2. CLASS CDANGULARMETERMBS 95

- 3.2.39 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double) 102
- 3.2.40 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double) 103
- 3.2.41 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double) 104

3.2.33 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. If is designed to be used with new style pointers (added with AngularMeter.addPointer2). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

Argument	Default	Description
backColor	0x888888	The fill color of the back circle.
frontColor	0x000000	The fill color of the front circle.
frontEdgeColor	0x888888	The edge color of the front circle.
lightingRatio	NoValue	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
backRadiusRatio	NoValue	The radius of the back circle, expressed as a ratio to the meter scale radius (set with AngularMeter.setMeter). The default is 0.15.
frontRadiusRatio	NoValue	The radius of the front circle, expressed as a ratio to the meter scale radius (set with AngularMeter.setMeter). The default is 0.11.
frontEdgeWidthRatio	NoValue	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with AngularMeter.setMeter). The default is 0.01.

See also:

- 3.2.32 setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888) 94
- 3.2.34 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double) 96
- 3.2.35 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double) 97

- 3.2.36 `setCap2`(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double) 98
- 3.2.37 `setCap2`(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888) 100
- 3.2.38 `setCap2`(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double) 101
- 3.2.39 `setCap2`(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double) 102
- 3.2.40 `setCap2`(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double) 103
- 3.2.41 `setCap2`(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double) 104

3.2.34 `setCap2`(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. It is designed to be used with new style pointers (added with `AngularMeter.addPointer2`). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

Argument	Default	Description
<code>backColor</code>	<code>0x888888</code>	The fill color of the back circle.
<code>frontColor</code>	<code>0x000000</code>	The fill color of the front circle.
<code>frontEdgeColor</code>	<code>0x888888</code>	The edge color of the front circle.
<code>lightingRatio</code>	NoValue	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
<code>backRadiusRatio</code>	NoValue	The radius of the back circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.15.
<code>frontRadiusRatio</code>	NoValue	The radius of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.11.
<code>frontEdgeWidthRatio</code>	NoValue	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.01.

See also:

3.2. CLASS CDANGULARMETERMBS	97
• 3.2.32 setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888)	94
• 3.2.33 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double)	95
• 3.2.35 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)	97
• 3.2.36 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)	98
• 3.2.37 setCap2(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888)	100
• 3.2.38 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)	101
• 3.2.39 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double)	102
• 3.2.40 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)	103
• 3.2.41 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)	104

3.2.35 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. It is designed to be used with new style pointers (added with AngularMeter.addPointer2). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

See also:

- 3.2.32 setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888) 94
- 3.2.33 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double) 95

Argument	Default	Description
backColor	0x888888	The fill color of the back circle.
frontColor	0x000000	The fill color of the front circle.
frontEdgeColor	0x888888	The edge color of the front circle.
lightingRatio	NoValue	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
backRadiusRatio	NoValue	The radius of the back circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.15.
frontRadiusRatio	NoValue	The radius of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.11.
frontEdgeWidthRatio	NoValue	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.01.

- 3.2.34 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double)` 96
- 3.2.36 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)` 98
- 3.2.37 `setCap2(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888)` 100
- 3.2.38 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)` 101
- 3.2.39 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double)` 102
- 3.2.40 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)` 103
- 3.2.41 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)` 104

3.2.36 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. If is designed to be used with new style pointers (added with `AngularMeter.addPointer2`). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be

sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

Argument	Default	Description
backColor	0x888888	The fill color of the back circle.
frontColor	0x000000	The fill color of the front circle.
frontEdgeColor	0x888888	The edge color of the front circle.
lightingRatio	NoValue	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
backRadiusRatio	NoValue	The radius of the back circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.15.
frontRadiusRatio	NoValue	The radius of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.11.
frontEdgeWidthRatio	NoValue	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.01.

See also:

- 3.2.32 `setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888)` 94
- 3.2.33 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double)` 95
- 3.2.34 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double)` 96
- 3.2.35 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)` 97
- 3.2.37 `setCap2(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888)` 100
- 3.2.38 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)` 101
- 3.2.39 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double)` 102
- 3.2.40 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)` 103
- 3.2.41 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)` 104

3.2.37 `setCap2(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. It is designed to be used with new style pointers (added with `AngularMeter.addPointer2`). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

Argument	Default	Description
<code>backColor</code>	<code>0x888888</code>	The fill color of the back circle.
<code>frontColor</code>	<code>0x000000</code>	The fill color of the front circle.
<code>frontEdgeColor</code>	<code>0x888888</code>	The edge color of the front circle.
<code>lightingRatio</code>	<code>NoValue</code>	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
<code>backRadiusRatio</code>	<code>NoValue</code>	The radius of the back circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.15.
<code>frontRadiusRatio</code>	<code>NoValue</code>	The radius of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.11.
<code>frontEdgeWidthRatio</code>	<code>NoValue</code>	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.01.

See also:

- 3.2.32 `setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888)` 94
- 3.2.33 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double)` 95
- 3.2.34 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double)` 96
- 3.2.35 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)` 97
- 3.2.36 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)` 98
- 3.2.38 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)` 101

- 3.2. CLASS CDANGULARMETERMBS 101
- 3.2.39 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double) 102
- 3.2.40 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double) 103
- 3.2.41 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double) 104

3.2.38 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. It is designed to be used with new style pointers (added with AngularMeter.addPointer2). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

Argument	Default	Description
backColor	0x888888	The fill color of the back circle.
frontColor	0x000000	The fill color of the front circle.
frontEdgeColor	0x888888	The edge color of the front circle.
lightingRatio	NoValue	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
backRadiusRatio	NoValue	The radius of the back circle, expressed as a ratio to the meter scale radius (set with AngularMeter.setMeter). The default is 0.15.
frontRadiusRatio	NoValue	The radius of the front circle, expressed as a ratio to the meter scale radius (set with AngularMeter.setMeter). The default is 0.11.
frontEdgeWidthRatio	NoValue	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with AngularMeter.setMeter). The default is 0.01.

See also:

- 3.2.32 setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888) 94
- 3.2.33 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double) 95
- 3.2.34 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double) 96

- 3.2.35 `setCap2`(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double) 97
- 3.2.36 `setCap2`(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double) 98
- 3.2.37 `setCap2`(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888) 100
- 3.2.39 `setCap2`(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double) 102
- 3.2.40 `setCap2`(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double) 103
- 3.2.41 `setCap2`(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double) 104

3.2.39 `setCap2`(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. It is designed to be used with new style pointers (added with `AngularMeter.addPointer2`). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

Argument	Default	Description
<code>backColor</code>	0x888888	The fill color of the back circle.
<code>frontColor</code>	0x000000	The fill color of the front circle.
<code>frontEdgeColor</code>	0x888888	The edge color of the front circle.
<code>lightingRatio</code>	NoValue	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
<code>backRadiusRatio</code>	NoValue	The radius of the back circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.15.
<code>frontRadiusRatio</code>	NoValue	The radius of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.11.
<code>frontEdgeWidthRatio</code>	NoValue	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.01.

See also:

3.2. CLASS CDANGULARMETERMBS	103
• 3.2.32 setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888)	94
• 3.2.33 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double)	95
• 3.2.34 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double)	96
• 3.2.35 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)	97
• 3.2.36 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)	98
• 3.2.37 setCap2(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888)	100
• 3.2.38 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)	101
• 3.2.40 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)	103
• 3.2.41 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)	104
3.2.40 setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)	

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. It is designed to be used with new style pointers (added with AngularMeter.addPointer2). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

See also:

- 3.2.32 setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888) 94
- 3.2.33 setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double) 95

Argument	Default	Description
backColor	0x888888	The fill color of the back circle.
frontColor	0x000000	The fill color of the front circle.
frontEdgeColor	0x888888	The edge color of the front circle.
lightingRatio	NoValue	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
backRadiusRatio	NoValue	The radius of the back circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.15.
frontRadiusRatio	NoValue	The radius of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.11.
frontEdgeWidthRatio	NoValue	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.01.

- 3.2.34 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double)` 96
- 3.2.35 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)` 97
- 3.2.36 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)` 98
- 3.2.37 `setCap2(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888)` 100
- 3.2.38 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)` 101
- 3.2.39 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double)` 102
- 3.2.41 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)` 104

3.2.41 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the meter to use new style pointer cap.

Notes: The new style pointer cap is a circular object covering the the center of the meter. If is designed to be used with new style pointers (added with `AngularMeter.addPointer2`). The cap consists of two circles, one at the background below the pointer, and one covering the pointer. The pointer will appear to be

sandwiched between these two circles.

Arguments are optional and colors can be passed as Integer or color value.

Argument	Default	Description
backColor	0x888888	The fill color of the back circle.
frontColor	0x000000	The fill color of the front circle.
frontEdgeColor	0x888888	The edge color of the front circle.
lightingRatio	NoValue	By default, lighting effect is applied to the back circle and the edge of the front circle, such that the bottom-right side will appear to be darkened. This argument configures how bright the bottom-right side is. The default value is 0.2, which means the bottom-right side is 20% as bright as the top-left side. Setting this argument to 1 disables the lighting effect.
backRadiusRatio	NoValue	The radius of the back circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.15.
frontRadiusRatio	NoValue	The radius of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.11.
frontEdgeWidthRatio	NoValue	The edge width of the front circle, expressed as a ratio to the meter scale radius (set with <code>AngularMeter.setMeter</code>). The default is 0.01.

See also:

- 3.2.32 `setCap2(backcolor as Color = &c888888, frontColor as Color = &c000000, frontEdgeColor as Color = &c888888)` 94
- 3.2.33 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double)` 95
- 3.2.34 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double)` 96
- 3.2.35 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)` 97
- 3.2.36 `setCap2(backcolor as Color, frontColor as Color, frontEdgeColor as Color, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double, frontEdgeWidthRatio as Double)` 98
- 3.2.37 `setCap2(backcolor as Integer = &h888888, frontColor as Integer = &h000000, frontEdgeColor as Integer = &h888888)` 100
- 3.2.38 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double)` 101
- 3.2.39 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double)` 102
- 3.2.40 `setCap2(backcolor as Integer, frontColor as Integer, frontEdgeColor as Integer, lightingRatio as Double, backRadiusRatio as Double, frontRadiusRatio as Double)` 103

3.2.42 setMeter(cx as Integer, cy as Integer, radius as Integer, startAngle as Double, endAngle as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position and angle range of the meter.

Notes:

Parameter	Default	Description
cx	(Mandatory)	The x-coordinate of the center of the meter.
cy	(Mandatory)	The y-coordinate of the center of the meter.
radius	(Mandatory)	The radius of the meter.
startAngle	(Mandatory)	The start angle of the meter. The angle is measured in degrees clockwise, with 0 being the upward pointing direction.
endAngle	(Mandatory)	The end angle of the meter. The angle is measured in degrees clockwise, with 0 being the upward pointing direction.

3.3 class CDAreaLayerMBS

3.3.1 class CDAreaLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The AreaLayer class represents area layers.

Example:

```
// The data for the area chart
dim data0(-1) as Double = array(42, 49, 33, 38, 51, 46, 29, 41, 44, 57, 59, 52, 37, 34, 51, 56, 56, 60, 70, 76,
63, 67, 75, 64, 51.0)
dim data1(-1) as Double = array(50, 45, 47, 34, 42, 49, 63, 62, 73, 59, 56, 50, 64, 60, 67, 67, 58, 59, 73, 77,
84, 82, 80, 84, 89.0)
dim data2(-1) as Double = array(61, 79, 85, 66, 53, 39, 24, 21, 37, 56, 37, 22, 21, 33, 13, 17, 4, 23, 16, 25, 9,
10, 5, 7, 16.0)
dim labels(-1) as string = array("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14",
"15", "16", "17", "18", "19", "20", "21", "22", "23", "24")

// Create a XYChart object of size 500 x 300 pixels
dim c as new CDXYChartMBS(500, 300)

// Set the plotarea at (90, 30) and of size 300 x 240 pixels.
call c.setPlotArea(90, 30, 300, 240)

// Add a legend box at (405, 100)
call c.addLegend(405, 100)

// Add a title to the chart
call c.addTitle("Daily System Load")

// Add a title to the y axis. Draw the title upright (font angle = 0)
c.yAxis.setTitle("Database"+endofline.unix+"Queries"+endofline.unix+"(per sec)").setFontAngle(0)

// Set the labels on the x axis.
call c.xAxis.setLabels(labels)

// Display 1 out of 2 labels on the x-axis. Show minor ticks for remaining
// labels.
c.xAxis.setLabelStep(2, 1)

// Add an area layer
dim layer as CDAreaLayerMBS
layer = c.addAreaLayer

// Draw the area layer in 3D
layer.set3D
```

```
// Add the three data sets to the area layer
call layer.addDataSet(data0, -1, "Server # 1")
call layer.addDataSet(data1, -1, "Server # 2")
call layer.addDataSet(data2, -1, "Server # 3")

// Output the chart
Backdrop=c.makeChartPicture
```

Notes: Subclass of the CDLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.3.2 Methods

3.3.3 setGapColor(fillColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setGapColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.3.4 setGapColor(fillColor as Integer) 108

3.3.4 setGapColor(fillColor as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color used to fill the area under kNoValue data points.

Notes: By default, if there are kNoValue data points, ChartDirector will interpolate using the remaining data points. The area will remain continuous.

This method can be used to set up an alternative color to represent the area at kNoValue data point positions. In particular, if the fillColor argument is set to Transparent, the kNoValue data points will result in gaps in the area.

Parameter	Default	Description
fillColor	(Mandatory)	The color used to fill the area region at kNoValue data point positions.

See also:

- 3.3.3 setGapColor(fillColor as color) 108

3.3.5 setMinLabelSize(s as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the minimum height (or width if x and y axes are swapped) of an area below which data labels will be hidden.

Notes: In ChartDirector, for an area layer, data labels (Layer.setDataLabelStyle) are drawn internal to the area, while the aggregate labels (Layer.setAggregateLabelStyle) are drawn external to the area.

ChartDirector will disable data labels if the area height (or width if x and y axes are swapped) is too small to contain the data label.

By default, ChartDirector will automatically determine what is meant by "too small". The setMinLabelSize method can be used to manually defined the threshold for "too small".

Sometimes it may be desirable to display the data label even though it cannot be contained within the area. In this case, the setMinLabelSize can be used to set the threshold to 0.

Parameter	Default	Description
s	(Mandatory)	The minimum height (or width if x and y axes are swapped) of an area below which data labels will be hidden.

3.4 class CDArrayMBS

3.4.1 class CDArrayMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The CDArrayMBS class is a utility class used to perform array computations.

Blog Entries

- [MBS Xojo Plugins, version 19.4pr5](#)
- [MBS Real Studio Plugins, version 13.1pr16](#)

3.4.2 Methods

3.4.3 abs

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by its absolute value.

3.4.4 acc

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by the accumulated total of its previous element (including itself).

3.4.5 addArray(value as CDArrayMBS)

Plugin Version: 16.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The array will be added by adding the each array element to the corresponding element of the CDArrayMBS object.

Notes:

Parameter	Default	Description
Values	(Mandatory)	A CDArrayMBS with numbers to be added to the CDArrayMBS object.

This method does not append a new value to the array.

See also:

- 3.4.6 addArray(value() as Double)

3.4.6 addArray(value() as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The array will be added by adding the each array element to the corresponding element of the CDArrayMBS object.

Example:

```
dim src(-1) as Double = Array( 63.1, 10.15, 6.15, 2.88 )
dim data As New CDArrayMBS(array(1.0, 1.0, 1.0, 1.0))
```

```
data.addArray( src )
```

```
dim lines(-1) as string
```

```
lines.Append str(data.count)+" values:"
lines.Append ""
lines.Append str(Data.getvalue(0))
lines.Append str(Data.getvalue(1))
lines.Append str(Data.getvalue(2))
lines.Append str(Data.getvalue(3))
```

```
MsgBox Join(lines,EndOfLine)
```

Notes:

Parameter	Default	Description
b	(Mandatory)	An array of numbers to be added to the CDArrayMBS object.

This method does not append a new value to the array.

See also:

- 3.4.5 addArray(value as CDArrayMBS)

110

3.4.7 addValue(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a number to every element of the CDArrayMBS object.

Example:

```
dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0))
```

```
data.addValue(5)
```

```
dim lines(-1) as string
```

```

lines.Append str(data.count)+" values:"
lines.Append ""
lines.Append str(Data.getvalue(0))
lines.Append str(Data.getvalue(1))
lines.Append str(Data.getvalue(2))
lines.Append str(Data.getvalue(3))

```

```
MsgBox Join(lines,EndOfLine)
```

Notes:

Parameter	Default	Description
b	(Mandatory)	A number to be added to every element of the CDArrayMBS object.

This method does not append a new value to the array.

3.4.8 aggregate(srcArray() as Double, aggregateMethod as Integer, param as Double = 50.0) as CDArrayMBS

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Aggregates a data array by merging groups of elements, similar to the GROUP BY clause in SQL statements.

Notes: This primary usage of this method is for changing data resolution, such as to convert a daily data into weekly data.

If the data is from a database, aggregation can often be more efficiently and conveniently performed by using the GROUP BY clause of SQL. This method should only be used when the data are not coming from a database, or the aggregation method are not supported by the database used.

In this method, the CDArrayMBS object represents the delimiters used to group elements in the srcArray. The positions of all non-kNoValue elements in the CDArrayMBS object will be used as delimiters positions.

For example, if the CDArrayMBS object consists of an array of 50 elements, and only the elements at 0, 10, 20, 30, 40 are not NoValue, then the groups will be defined as positions 0 - 9, 10 - 19, 20 - 29, 30 - 39 and 40 - 49.

Note that a group includes the starting delimiter position but excludes the ending delimiter position.

The CDArrayMBS object is typically created by applying CDArrayMBS.selectStartOfHour, CDArrayMBS.se-

lectStartOfDay, CDArrayMBS.selectStartOfWeek, CDArrayMBS.selectStartOfMonth, CDArrayMBS.selectStartOfYear or CDArrayMBS.selectRegularSpacing to a data array.

For example, to group daily data into weekly data, one may create an CDArrayMBS object with the dates of the daily data, then CDArrayMBS.selectStartOfWeek to select only the elements representing the start of a week. The resulting CDArrayMBS object can then be used to aggregate daily data into weekly data.

Due to aggregation, the length of the output array will usually be shorter than, and never be longer than, the length of the input array.

Parameter	Default	Description
srcArray	(Mandatory)	The array to be aggregated.
aggregateMethod	(Mandatory)	The method to aggregate the data, which must be one of AggregateSum, AggregateAvg, AggregateStdDev, AggregateMin, AggregateMed, AggregateMax, AggregatePercentile, AggregateFirst, AggregateLast, AggregateCount.
param	50	The aggregation parameter, if needed. Currently, only AggregatePercentile needs a parameter to specify the percentile used.

3.4.9 aggregateValues(srcArray() as Double, aggregateMethod as Integer, param as Double = 50.0) as Double()

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Aggregates a data array by merging groups of elements, similar to the GROUP BY clause in SQL statements.

Notes: This primary usage of this method is for changing data resolution, such as to convert a daily data into weekly data.

If the data is from a database, aggregation can often be more efficiently and conveniently performed by using the GROUP BY clause of SQL. This method should only be used when the data are not coming from a database, or the aggregation method are not supported by the database used.

In this method, the CDArrayMBS object represents the delimiters used to group elements in the srcArray. The positions of all non-kNoValue elements in the CDArrayMBS object will be used as delimiters positions.

For example, if the CDArrayMBS object consists of an array of 50 elements, and only the elements at 0, 10, 20, 30, 40 are not NoValue, then the groups will be defined as positions 0 - 9, 10 - 19, 20 - 29, 30 - 39 and 40 - 49.

Note that a group includes the starting delimiter position but excludes the ending delimiter position.

The CDArrayMBS object is typically created by applying CDArrayMBS.selectStartOfHour, CDArrayMBS.selectStartOfDay, CDArrayMBS.selectStartOfWeek, CDArrayMBS.selectStartOfMonth, CDArrayMBS.selectStartOfYear or CDArrayMBS.selectRegularSpacing to a data array.

For example, to group daily data into weekly data, one may create an CDArrayMBS object with the dates of the daily data, then CDArrayMBS.selectStartOfWeek to select only the elements representing the start of a week. The resulting CDArrayMBS object can then be used to aggregate daily data into weekly data.

Due to aggregation, the length of the output array will usually be shorter than, and never be longer than, the length of the input array.

Parameter	Default	Description
srcArray	(Mandatory)	The array to be aggregated.
aggregateMethod	(Mandatory)	The method to aggregate the data, which must be one of AggregateSum, AggregateAvg, AggregateStdDev, AggregateMin, AggregateMed, AggregateMax, AggregatePercentile, AggregateFirst, AggregateLast, AggregateCount.
param	50	The aggregation parameter, if needed. Currently, only AggregatePercentile needs a parameter to specify the percentile used.

3.4.10 avg as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the avg value of the elements of the CDArrayMBS object.

Example:

```
dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0))
MsgBox str(data.avg) // shows 2.5
```

3.4.11 Constructor

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The dummy constructor doing nothing.

See also:

- 3.4.12 Constructor(a as CDArrayMBS) 115
- 3.4.13 Constructor(data() as Double) 115

3.4.12 Constructor(a as CDArrayMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an CDArrayMBS object and initialize it with the given array.

See also:

- 3.4.11 Constructor 114
- 3.4.13 Constructor(data() as Double) 115

3.4.13 Constructor(data() as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an CDArrayMBS object and initialize it with the given array.

See also:

- 3.4.11 Constructor 114
- 3.4.12 Constructor(a as CDArrayMBS) 115

3.4.14 count as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of elements in this array object.

Example:

```
dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0))
MsgBox str(data.count) // shows 4
```

3.4.15 delta(offset as Integer = 1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Subtracts each element of the CDArrayMBS object by an earlier element in the same CDArrayMBS object.

Notes:

Parameter	Default	Description
offset	1	The difference in position between an element and the earlier element to be subtracted from it.

3.4.16 divArray(value as CDArryMBS)

Plugin Version: 16.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Divides the CDArryMBS object by the given array.

Notes: The CDArryMBS object will be divided by dividing each of its elements by the corresponding element in the given array.

Parameter	Default	Description
Values	(Mandatory)	A CDArryMBS with numbers used as divisors to divide the CDArryMBS object.

See also:

- 3.4.17 divArray(value() as Double)

116

3.4.17 divArray(value() as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Divides the CDArryMBS object by the given array.

Example:

```
dim src(-1) as Double = Array( 63.1, 10.15, 6.15, 2.88 )
dim data As New CDArryMBS(array(2.0, 2.0, 2.0, 2.0))
```

```
data.divArray( src )
```

```
dim lines(-1) as string
```

```
lines.Append str(data.count)+" values:"
lines.Append ""
lines.Append str(Data.getvalue(0))
lines.Append str(Data.getvalue(1))
lines.Append str(Data.getvalue(2))
lines.Append str(Data.getvalue(3))
```

```
MsgBox Join(lines,EndOfLine)
```

Notes: The CDArryMBS object will be divided by dividing each of its elements by the corresponding element in the given array.

See also:

3.4. CLASS CDARRAYMBS

117

Parameter	Default	Description
b	(Mandatory)	An array of numbers used as divisors to divide the CDArrayMBS object.

- 3.4.16 divArray(value as CDArrayMBS)

116

3.4.18 divValue(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Divides every element of the CDArrayMBS object by the given number.

Example:

```
dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0))
```

```
data.divValue(5)
```

```
dim lines(-1) as string
```

```
lines.Append str(data.count)+" values:"
```

```
lines.Append ""
```

```
lines.Append str(Data.getvalue(0))
```

```
lines.Append str(Data.getvalue(1))
```

```
lines.Append str(Data.getvalue(2))
```

```
lines.Append str(Data.getvalue(3))
```

```
MsgBox Join(lines,EndOfLine)
```

Notes:

Parameter	Default	Description
b	(Mandatory)	A number to be used as divisor to divide every element of the CDArrayMBS object.

3.4.19 expAvg(smoothingFactor as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by its exponential average.

Notes: The exponential average is computed by :

$$\text{avg}(n) = \text{value}(n) * \text{smoothingFactor} + \text{avg}(n - 1) * (1 - \text{smoothingFactor})$$

where $\text{avg}(n)$ is the exponential average of the n th element, and $\text{value}(n)$ is the value of the n th element.

For the first element ($n = 0$), its exponential average is assumed to be equal to its original value.

Parameter	Default	Description
<code>smoothingFactor</code>	(Mandatory)	The smoothing factor used for computing exponential average. It should be between 0 - 1.

3.4.20 `financeDiv(values() as Double, zeroByZeroValue as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Divides the `CDArraryMBS` object by the given array, with special handling of the case of zero divided by zero.

Notes: In many financial formulas, it is possible to have cases of zero divided by zero. For example, the Relative Strength Index is defined as the ratio between positive price changes and absolute value of all price changes over a period of time. In case the price does not change at all during that period, and the RSI will become zero divided by zero.

Traditionally, under the above case, the RSI will be consider as 50%.

To handle these special cases, the `financeDiv` method has an argument specifying what value to assume in case the division is zero divided by zero.

Parameter	Default	Description
<code>b</code>	(Mandatory)	An array of numbers used as divisors to divide the <code>CDArraryMBS</code> object.
<code>zeroByZeroValue</code>	(Mandatory)	The value to use if the division is zero divided by zero.

3.4.21 `getvalue(index as Integer) as Double`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the vaue with the given index.

Notes: Returns zero if the value does not exist.

3.4.22 `insert(value as Double, len as Integer, insertPoint as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Inserts some constant elements to the CDArrayMBS object.

Notes:

Parameter	Default	Description
c	(Mandatory)	The value of the constant element.
len	(Mandatory)	The number of elements to insert.
insertPoint	-1	The position of the insertion point. The new elements will be inserted just before the insertion point. -1 means inserting elements at the end of the array.

See also:

- 3.4.23 insert(value() as Double, insertPoint as Integer = -1) 119

3.4.23 insert(value() as Double, insertPoint as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Inserts the elements of an array to the CDArrayMBS object.

Notes:

Parameter	Default	Description
a	(Mandatory)	An array to be inserted to the CDArrayMBS object.
insertPoint	-1	The position of the insertion point. The new elements will be inserted just before the insertion point. -1 means inserting elements at the end of the array.

See also:

- 3.4.22 insert(value as Double, len as Integer, insertPoint as Integer = -1) 118

3.4.24 lowess(smoothness as Double = 0.25, iteration as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fits a curve through the data points in the CDArrayMBS object using the LOWESS algorithm.

Example:

```
// generate some values
dim values() as Double
dim x as Double

for i as Integer = 1 to 10
x = x + rnd - 0.5
values.Append x
next
```

```

// make array
dim c as new CDArrayMBS(values)
dim oldValues() as Double = c.Values

// now run Algorithm
c.lowess(1.0) // 1.0 so we see it

// now get new values
dim newValues() as Double = c.Values

// the rest is for displaying in MsgBox:
dim oldValueStrings() as string
dim newValueStrings() as string

for each v as Double in oldValues
oldValueStrings.Append str(v)
next

for each v as Double in newValues
newValueStrings.Append str(v)
next

MsgBox join(oldValueStrings, " ") + EndOfLine + EndOfLine + join(newValueStrings, " ")

```

Notes: The full name of LOWESS is "Robust locally weighted regression and smoothing scatterplots". It is a commonly used algorithm for drawing a smooth curve through a number of points.

LOWESS works by assuming a small segment of any curve can be approximated by a straight line. For each data point, LOWESS finds the *n* nearest points to that data point (*n* is configurable), and performs weighted linear regression using a tricube weighting function. It then adjust the coordinates of the data point based on the result of the weighted linear regression.

LOWESS can run in multiple iterations, in which case it should converge to a stable curve - thus it is called "robust".

In most cases, LOWESS behaves better than many other smoothing algorithms, such as moving average, moving median, exponential average. Curves draw using LOWESS look smoother, yet they track the data points better. Also, LOWESS behaves well at the end points. On the other hand, methods based on moving windows (e.g. moving averages) do not work on the first few data points, because they need sufficient data points to fill the moving window first.

In this method, each element of the CDArrayMBS object will be replaced by the corresponding value computed using the LOWESS algorithm.

Parameter	Default	Description
smoothness	0.25	The smoothness factor. It must be between 0 - 1. It is the portion of points used in finding the n nearest points. In other words, $n = \text{smoothness} * \text{no_of_points}$. A larger value will result in a smoother the curve. A smaller value will result in the curve tracking the data points better.

For LOWESS to have any smoothing effect at all, n must be at least 3. You may need to use a large smoothness factor if you only have a few data points.

iteration	0	The number of additional iteration used in the LOWESS algorithm. Unless your data is extremely noisy, in most case no additional iteration is necessary.
-----------	---	--

See also:

- 3.4.25 lowess(values() as Double, smoothness as Double = 0.25, iteration as Integer = 0) 121

3.4.25 lowess(values() as Double, smoothness as Double = 0.25, iteration as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fits a curve through the data points in the CDArrayMBS object using the LOWESS algorithm, where the spacing of the data points is supplied by the given array.

Notes: Please refer to CDArrayMBS.lowess for a brief description of the LOWESS algorithm.

In this method, each element of the CDArrayMBS object will be replaced by the corresponding value computed using the LOWESS algorithm.

Parameter	Default	Description
b	(Mandatory)	An array of numbers providing the x coordinates of the data points in the CDArrayMBS object.
smoothness	0.25	The smoothness factor. It must be between 0 - 1. It is the portion of points used in finding the n nearest points. In other words, $n = \text{smoothness} * \text{no_of_points}$. A larger value will result in a smoother the curve. A smaller value will result in the curve tracking the data points better.

For LOWESS to have any smoothing effect at all, n must be at least 3. You may need to use a large smoothness factor if you only have a few data points.

See also:

iteration 0 The number of additional iteration used in the LOWESS algorithm. Unless your data is extremely noisy, in most case no additional iteration is necessary.

- 3.4.24 lowess(smoothness as Double = 0.25, iteration as Integer = 0)

119

3.4.26 max as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the maximum value of the elements of the CDArrayMBS object.

Example:

```
dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0))
MsgBox str(data.max) // shows 4.0
```

3.4.27 maxIndex as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the index of the maximum value element of the CDArrayMBS object.

3.4.28 med as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the med value of the elements of the CDArrayMBS object.

Example:

```
dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0, 1.0))
MsgBox str(data.med) // shows 2.0
MsgBox str(data.avg) // shows 2.2
```

3.4.29 min as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the minimum value of the elements of the CDArrayMBS object.

Example:

```
dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0))
MsgBox str(data.min) // shows 1.0
```

3.4.30 minIndex as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the index of the minimum value element of the CDArrayMBS object.

3.4.31 movAvg(interval as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by its moving average.

Notes: The interval parameter specifies the window size for computing moving average. The moving average is computed as the average of the current element with the previous (interval - 1) elements. No moving average can be computed for the first (interval - 1) elements, because there are insufficient previous elements. So the first (interval - 1) elements will be replaced with kNoValue.

Parameter	Default	Description
interval	(Mandatory)	The window size.

3.4.32 movCorr(interval as Integer, value() as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by the moving correlation with another array or with itself.

Notes: The interval parameter specifies the window size for computing moving correlation. The moving correlation is computed as the correlation coefficient between the CDArrayMBS object and the other array, where only the current element and the previous (interval - 1) elements are considered in the computation.

If the other array is not provided, it is assumed to be the sequence of numbers 0, 1, 2, 3, 4.... This is equivalent to checking if the elements of the CDArrayMBS object is linear.

No moving correlation can be computed for the first (interval - 1) elements, because there are insufficient previous elements. So the first (interval - 1) elements will be replaced with kNoValue.

interval	(Mandatory)	The window size.	b	[Empty_Array]	The array to be correlated with the CDArrayMBS object. If this argument is an empty array, the sequence of numbers 0, 1, 2, 3, 4 will be used instead.
----------	-------------	------------------	---	-----------------	---

3.4.33 movMax(interval as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by its moving maximum.

Notes: The interval parameter specifies the window size for computing moving average. The moving average is computed as the maximum of the current element with the previous (interval - 1) elements. No moving maximum can be computed for the first (interval - 1) elements, because there are insufficient previous elements. So the first (interval - 1) elements will be replaced with kNoValue.

Parameter	Default	Description
interval	(Mandatory)	The window size.

3.4.34 movMed(interval as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by its moving median.

Notes: The interval parameter specifies the window size for computing moving average. The moving average is computed as the median of the current element with the previous (interval - 1) elements. No moving median can be computed for the first (interval - 1) elements, because there are insufficient previous elements. So the first (interval - 1) elements will be replaced with kNoValue.

Parameter	Default	Description
interval	(Mandatory)	The window size.

3.4.35 movMin(interval as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by its moving minimum.

Notes: The interval parameter specifies the window size for computing moving average. The moving average is computed as the minimum of the current element with the previous (interval - 1) elements. No moving minimum can be computed for the first (interval - 1) elements, because there are insufficient previous elements. So the first (interval - 1) elements will be replaced with kNoValue.

Parameter	Default	Description
interval	(Mandatory)	The window size.

3.4.36 movPercentile(interval as Integer, percentile as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by its moving percentile.

Notes: The interval parameter specifies the window size for computing moving average. The moving average is computed as the percentile of the current element with the previous (interval - 1) elements. No moving percentile can be computed for the first (interval - 1) elements, because there are insufficient previous elements. So the first (interval - 1) elements will be replaced with kNoValue.

Parameter	Default	Description
interval	(Mandatory)	The window size.
percentile	(Mandatory)	The percentile to be computed. It should be between 0 - 100.

3.4.37 movStdDev(interval as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Replaces each element of the CDArrayMBS object by its moving standard deviation.

Notes: The interval parameter specifies the window size for computing moving average. The moving average is computed as the standard deviation of the current element with the previous (interval - 1) elements. No moving standard deviation can be computed for the first (interval - 1) elements, because there are insufficient previous elements. So the first (interval - 1) elements will be replaced with kNoValue.

Parameter	Default	Description
interval	(Mandatory)	The window size.

3.4.38 mulArray(value as CDArrayMBS)

Plugin Version: 16.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Multiplies an array to the CDArrayMBS object.

Notes: The array will be multiplied by multiplying each array element with the corresponding element of the CDArrayMBS object.

See also:

- 3.4.39 mulArray(value() as Double)

Parameter	Default	Description
Values	(Mandatory)	A CArrayMBS with numbers to be multiplied to the CArrayMBS object.

3.4.39 mulArray(value() as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Multiplies an array to the CArrayMBS object.

Example:

```
dim src(-1) as Double = Array( 63.1, 10.15, 6.15, 2.88 )
dim data As New CArrayMBS(array(2.0, 2.0, 2.0, 2.0))
```

```
data.mulArray( src )
```

```
dim lines(-1) as string
```

```
lines.Append str(data.count)+" values:"
lines.Append ""
lines.Append str(Data.getvalue(0))
lines.Append str(Data.getvalue(1))
lines.Append str(Data.getvalue(2))
lines.Append str(Data.getvalue(3))
```

```
MsgBox Join(lines,EndOfLine)
```

Notes: The array will be multiplied by multiplying each array element with the corresponding element of the CArrayMBS object.

Parameter	Default	Description
b	(Mandatory)	An array of numbers to be multiplied to the CArrayMBS object.

See also:

- 3.4.38 mulArray(value as CArrayMBS)

125

3.4.40 mulValue(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Multiplies a number to every element of the CArrayMBS object.

Example:

```
dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0))
```

```
data.mulValue(5)
```

```
dim lines(-1) as string
```

```
lines.Append str(data.count)+" values:"
```

```
lines.Append ""
```

```
lines.Append str(Data.getvalue(0))
```

```
lines.Append str(Data.getvalue(1))
```

```
lines.Append str(Data.getvalue(2))
```

```
lines.Append str(Data.getvalue(3))
```

```
MsgBox Join(lines,EndOfLine)
```

Notes:

Parameter	Default	Description
b	(Mandatory)	A number to be multiplied to every element of the CDArrayMBS object.

3.4.41 percentile(p as Double) as double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the required percentile value of the elements of the CDArrayMBS object.

Notes:

Parameter	Default	Description
p	(Mandatory)	The percentile to get. It should be between 0 - 100.

3.4.42 rate(offset as Integer = 1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Divides each element of the CDArrayMBS object by an earlier element in the same CDArrayMBS object.

Notes:

Parameter	Default	Description
offset	1	The difference in position between an element and the earlier element to divide it.

3.4.43 `replace(a as Double, b as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finds elements that are equal to a given value and replace it with another value.

Notes:

Parameter	Default	Description
a	(Mandatory)	The value to be replaced.
b	(Mandatory)	The replacing value.

3.4.44 `result as memoryblock`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the content of the CDArrayMBS object as a memoryblock.

3.4.45 `selectEQZ`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is equal to zero.

Notes: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.46 `selectEQZ(decisionArray() as Double, fillValue as Double = 0)`

128

3.4.46 `selectEQZ(decisionArray() as Double, fillValue as Double = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is equal to zero.

Notes: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.45 selectEQZ 128

3.4.47 selectGEZ

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is greater than or equal to zero.

Notes: If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.48 selectGEZ(decisionArray() as Double, fillValue as Double = 0) 129

3.4.48 selectGEZ(decisionArray() as Double, fillValue as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is greater than or equal to zero.

Notes: If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

See also:

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

- 3.4.47 selectGEZ 129

3.4.49 selectGTZ

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is greater than zero.

Notes: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.50 selectGTZ(decisionArray() as Double, fillValue as Double = 0) 130

3.4.50 selectGTZ(decisionArray() as Double, fillValue as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is greater than zero.

Notes: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

See also:

- 3.4.49 selectGTZ 130

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

3.4.51 selectLEZ

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is less than or equal to zero.

Notes: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.52 selectLEZ(decisionArray() as Double, fillValue as Double = 0) 131

3.4.52 selectLEZ(decisionArray() as Double, fillValue as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is less than or equal to zero.

Notes: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.51 selectLEZ

131

3.4.53 selectLTZ

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is less than zero.

Notes: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.54 selectLTZ(decisionArray() as Double, fillValue as Double = 0)

132

3.4.54 selectLTZ(decisionArray() as Double, fillValue as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object by checking if the elements of the given decisionArray is less than zero.

Notes: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.53 selectLTZ

132

3.4.55 selectNEZ

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

Notes: If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.56 selectNEZ(decisionArray() as Double, fillValue as Double = 0) 133

3.4.56 selectNEZ(decisionArray() as Double, fillValue as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The selected elements will be left unchanged. The remaining elements will be replaced by the given fillValue.

Notes: If the decisionArray is empty, the CDArrayMBS object itself will function as the decision array.

Parameter	Default	Description
decisionArray	[Empty_Array]	An array of numbers used to decide whether the corresponding elements in the CDArrayMBS object is selected or not.
fillValue	0	The value used to replace the elements that are not selected.

See also:

- 3.4.55 selectNEZ 133

3.4.57 selectRegularSpacing(majorTickStep as Integer, minorTickStep as Integer = 0, initialMargin as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects an evenly spaced subset of elements of the CDArrayMBS object.

Notes: The primary purpose of this method is to select the regularly spaced elements as ticks on an enumerated axis.

For example, if the `majorTickStep` is 10, and `minorTickStep` is 5, this method will select one out of 10 elements as major tick elements, and one of out of 5 elements as minor tick elements.

Major tick elements will remain intact. The remaining minor ticks elements will have their values replaced with `MinorTickOnly`. Non-selected elements will have their values replaced with `kNoValue`, so they will not appear on the axis. The resulting array can be used directly in `Axis.setLabels`.

Parameter	Default	Description
<code>majorTickStep</code>	(Mandatory)	The spacing between major ticks.
<code>minorTickStep</code>	0	The spacing between minor ticks. The default value of 0 means no minor tick will be used.
<code>initialMargin</code>	0	The position of the first tick.

3.4.58 `selectStartOfDay(majorTickStep as Integer = 1, initialMargin as Double = 10800.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the `CDArraryMBS` object that represents a different day from the previous element.

Notes: This method assumes the elements of the `CDArraryMBS` object are dates/times.

The primary purpose of this method is to select the appropriate elements as ticks on an enumerated axis.

Suppose you want to plot a variable against time. The x-values of the data points will be an array of dates/times. If an enumerated x-axis is used (see `Axis.setLabels`), there will be a tick at every data point, which may be too dense if there are too many data points.

This method can be used to reduce the ticks to one tick per day (or one tick per multiple days). This is by selecting the dates/times in the data array only if it is not the same day as the previous element in the array. The selected elements will be left unchanged, while the elements not select will be replaced by `kNoValue`. The resulting array can be used directly in `Axis.setLabels`.

For the first data point, there is no previous data point to compare, so it is handled differently. The first data point will be selected if it is near the beginning of the day it represents. By default, near means within 3 hours (10800 seconds). This is configurable using the `initialMargin` argument.

Note that if the data points does not have data in a certain time range, no element can be selected in that time range, and so there will be no tick for that time range. This is appropriate for many chart types, such as finance charts, in which missing time ranges (non-trading hours, holidays, etc) are traditionally skipped.

However, if the data points may contain missing time ranges, but the ticks cannot be skipped, it may be more appropriate to use a non-enumerated x-axis by using `Layer.setXData` and `Axis.setDateScale`.

Parameter	Default	Description
<code>majorTickStep</code>	1	The tick step. The default value of 1 means one tick per day. A value of n means one tick per n days.
<code>initialMargin</code>	10800	The margin for the first data point. The first data point will be selected if it is within <code>initialMargin</code> number of seconds from beginning of the day it represents. The default is 3 hours (10800 seconds).

3.4.59 `selectStartOfHour(majorTickStep as Integer = 1, initialMargin as Double = 300.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the `CDArrayMBS` object that represents a different hour from the previous element.

Notes: This method assumes the elements of the `CDArrayMBS` object are dates/times.

The primary purpose of this method is to select the appropriate elements as ticks on an enumerated axis.

Suppose you want to plot a variable against time. The x-values of the data points will be an array of dates/times. If an enumerated x-axis is used (see `Axis.setLabels`), there will be a tick at every data point, which may be too dense if there are too many data points.

This method can be used to reduce the ticks to one tick per hour (or one tick per multiple hours). This is by selecting the dates/times in the data array only if it is not the same hour as the previous element in the array. The selected elements will be left unchanged, while the elements not select will be replaced by `kNoValue`. The resulting array can be used directly in `Axis.setLabels2`.

For the first data point, there is no previous data point to compare, so it is handled differently. The first data point will be selected if it is near the beginning of the hour it represents. By default, near means within 300 seconds. This is configurable using the `initialMargin` argument.

Note that if the data points does not have data in a certain time range, no element can be selected in that time range, and so there will be no tick for that time range. This is appropriate for many chart types, such as finance charts, in which missing time ranges (non-trading hours, holidays, etc) are traditionally skipped.

However, if the data points may contain missing time ranges, but the ticks cannot be skipped, it may be more appropriate to use a non-enumerated x-axis by using `Layer.setXData` and `Axis.setDateScale`.

Parameter	Default	Description
majorTickStep	1	The tick step. The default value of 1 means one tick per hour. A value of n means one tick per n hours.
initialMargin	300	The margin for the first data point. The first data point will be selected if it is within initialMargin number of seconds from beginning of the hour it represents. The default is 300 seconds.

3.4.60 selectStartOfMinute(majorTickStep as Integer = 1, initialMargin as Double = 5.0)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the array that represents a different minute from the previous element.

Notes: This method assumes the elements of the ArrayMath object are dates/times.

The primary purpose of this method is to select the appropriate elements as ticks on an enumerated axis.

Suppose you want to plot a variable against time. The x-values of the data points will be an array of dates/times. If an enumerated x-axis is used (see CDAxisMBS.setLabels), there will be a tick at every data point, which may be too dense if there are too many data points.

This method can be used to reduce the ticks to one tick per minute (or one tick per multiple minutes). This is by selecting the dates/times in the data array only if it is not the same minute as the previous element in the array. The selected elements will be left unchanged, while the elements not select will be replaced by kNoValue. The resulting array can be used directly in CDAxisMBS.setLabels2.

For the first data point, there is no previous data point to compare, so it is handled differently. The first data point will be selected if it is near the beginning of the minute it represents. By default, near means within 5 seconds. This is configurable using the initialMargin argument.

Note that if the data points does not have data in a certain time range, no element can be selected in that time range, and so there will be no tick for that time range. This is appropriate for many chart types, such as finance charts, in which missing time ranges (non-trading hours, holidays, etc) are traditionally skipped.

However, if the data points may contain missing time ranges, but the ticks cannot be skipped, it may be more appropriate to use a non-enumerated x-axis by using CDLayerMBS.setXData and CDAxisMBS.setDateScale3 (or CDAxisMBS.setDateScale or CDAxisMBS.setDateScale2).

Argument	Default	Description
majorTickStep	1	The tick step. A value of n means one tick per n minutes.
initialMargin	5	The margin for the first data point. The first data point will be selected if it is within initialMargin number of seconds from beginning of the minute it represents. The default is 5 seconds.

3.4.61 selectStartOfMonth(majorTickStep as Integer = 1, initialMargin as Double = 432000.0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object that represents a different month from the previous element.

Notes: This method assumes the elements of the CDArrayMBS object are dates/times.

The primary purpose of this method is to select the appropriate elements as ticks on an enumerated axis.

Suppose you want to plot a variable against time. The x-values of the data points will be an array of dates/times. If an enumerated x-axis is used (see `Axis.setLabels`), there will be a tick at every data point, which may be too dense if there are too many data points.

This method can be used to reduce the ticks to one tick per month (or one tick per multiple months). This is by selecting the dates/times in the data array only if it is not the same month as the previous element in the array. The selected elements will be left unchanged, while the elements not select will be replaced by `kNoValue`. The resulting array can be used directly in `Axis.setLabels`.

For the first data point, there is no previous data point to compare, so it is handled differently. The first data point will be selected if it is near the beginning of the month it represents. By default, near means within 5 days (432000 seconds). This is configurable using the `initialMargin` argument.

Note that if the data points does not have data in a certain time range, no element can be selected in that time range, and so there will be no tick for that time range. This is appropriate for many chart types, such as finance charts, in which missing time ranges (non-trading hours, holidays, etc) are traditionally skipped.

However, if the data points may contain missing time ranges, but the ticks cannot be skipped, it may be more appropriate to use a non-enumerated x-axis by using `Layer.setXData` and `Axis.setDateScale`.

Parameter	Default	Description
majorTickStep	1	The tick step. The default value of 1 means one tick per month. A value of n means one tick per n months.
initialMargin	432000	The margin for the first data point. The first data point will be selected if it is within initialMargin number of seconds from beginning of the month it represents. The default is 5 days (432000 seconds).

3.4.62 selectStartOfSecond(majorTickStep as Integer = 1, initialMargin as Double = 0.1)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the ArrayMath object that represents a different second from the previous element.

Notes: This method assumes the elements of the ArrayMath object are dates/times.

The primary purpose of this method is to select the appropriate elements as ticks on an enumerated axis.

Suppose you want to plot a variable against time. The x-values of the data points will be an array of dates/times. If an enumerated x-axis is used (see CDAxisMBS.setLabels), there will be a tick at every data point, which may be too dense if there are too many data points.

This method can be used to reduce the ticks to one tick per second (or one tick per multiple seconds). This is by selecting the dates/times in the data array only if it is not the same second as the previous element in the array. The selected elements will be left unchanged, while the elements not select will be replaced by NoValue. The resulting array can be used directly in CDAxisMBS.setLabels2.

For the first data point, there is no previous data point to compare, so it is handled differently. The first data point will be selected if it is near the beginning of the second it represents. By default, near means within 0.1 second. This is configurable using the initialMargin argument.

Note that if the data points does not have data in a certain time range, no element can be selected in that time range, and so there will be no tick for that time range. This is appropriate for many chart types, such as finance charts, in which missing time ranges (non-trading hours, holidays, etc) are traditionally skipped.

However, if the data points may contain missing time ranges, but the ticks cannot be skipped, it may be more appropriate to use a non-enumerated x-axis by using CDLayerMBS.setXData and CDAxisMBS.setDateScale3 (or CDAxisMBS.setDateScale or CDAxisMBS.setDateScale2).

Argument	Default	Description
majorTickStep	1	The tick step. A value of n means one tick per n seconds.
initialMargin	0.1	The margin for the first data point. The first data point will be selected if it is within initialMargin number of seconds from beginning of the second it represents. The default is 0.1 second.

3.4.63 selectStartOfWeek(majorTickStep as Integer = 1, initialMargin as Double = 172800.0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object that represents a different week from the previous element.

Notes: This method assumes the elements of the CDArrayMBS object are dates/times.

The primary purpose of this method is to select the appropriate elements as ticks on an enumerated axis.

Suppose you want to plot a variable against time. The x-values of the data points will be an array of dates/times. If an enumerated x-axis is used (see `Axis.setLabels`), there will be a tick at every data point, which may be too dense if there are too many data points.

This method can be used to reduce the ticks to one tick per week (or one tick per multiple weeks). This is by selecting the dates/times in the data array only if it is not the same week as the previous element in the array. The selected elements will be left unchanged, while the elements not select will be replaced by `kNoValue`. The resulting array can be used directly in `Axis.setLabels`.

For the first data point, there is no previous data point to compare, so it is handled differently. The first data point will be selected if it is near the beginning of the week it represents. By default, near means within 2 days (172800 seconds). This is configurable using the `initialMargin` argument.

Note that if the data points does not have data in a certain time range, no element can be selected in that time range, and so there will be no tick for that time range. This is appropriate for many chart types, such as finance charts, in which missing time ranges (non-trading hours, holidays, etc) are traditionally skipped.

However, if the data points may contain missing time ranges, but the ticks cannot be skipped, it may be more appropriate to use a non-enumerated x-axis by using `Layer.setXData` and `Axis.setDateScale`.

Parameter	Default	Description
majorTickStep	1	The tick step. The default value of 1 means one tick per week. A value of n means one tick per n weeks.
initialMargin	172800	The margin for the first data point. The first data point will be selected if it is within initialMargin number of seconds from beginning of the week it represents. The default is 2 days (172800 seconds).

3.4.64 selectStartOfYear(majorTickStep as Integer = 1, initialMargin as Double = 5184000.0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the elements of the CDArrayMBS object that represents a different year from the previous element.

Notes: This method assumes the elements of the CDArrayMBS object are dates/times. The primary purpose of this method is to select the appropriate elements as ticks on an enumerated axis.

Suppose you want to plot a variable against time. The x-values of the data points will be an array of dates/times. If an enumerated x-axis is used (see `Axis.setLabels`), there will be a tick at every data point, which may be too dense if there are too many data points.

This method can be used to reduce the ticks to one tick per year (or one tick per multiple years). This is by selecting the dates/times in the data array only if it is not the same year as the previous element in the array. The selected elements will be left unchanged, while the elements not select will be replaced by `kNoValue`. The resulting array can be used directly in `Axis.setLabels`.

For the first data point, there is no previous data point to compare, so it is handled differently. The first data point will be selected if it is near the beginning of the year it represents. By default, near means within 60 days (5184000 seconds). This is configurable using the `initialMargin` argument.

Note that if the data points does not have data in a certain time range, no element can be selected in that time range, and so there will be no tick for that time range. This is appropriate for many chart types, such as finance charts, in which missing time ranges (non-trading hours, holidays, etc) are traditionally skipped.

However, if the data points may contain missing time ranges, but the ticks cannot be skipped, it may be more appropriate to use a non-enumerated x-axis by using `Layer.setXData` and `Axis.setDateScale`.

Parameter	Default	Description
majorTickStep	1	The tick step. The default value of 1 means one tick per year. A value of n means one tick per n years.
initialMargin	5184000	The margin for the first data point. The first data point will be selected if it is within initialMargin number of seconds from beginning of the year it represents. The default is 60 days (5184000 seconds).

3.4.65 shift(offset as Integer = 1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shifts the array "rightwards".

Notes: If the array does not have any kNoValue data points, this method will shift the point at position "n" to "n + offset". On the "right" side of the array (the side with the largest index), points that are shifted outside the array will be discarded. On the "left" side of the array, fillValue data points will be shifted in.

If the array contains kNoValue data points, these points are not shifted. Conceptually, one can imagine the non-NoValue points being shifted to the next non-NoValue positions, and the process repeats offset number of times.

Parameter	Default	Description
offset	1	The number of positions to shift the array "rightwards."
fillValue	kNoValue	The new value to be shifted into the array.

See also:

- 3.4.66 shift(offset as Integer, fillValue as Double)

141

3.4.66 shift(offset as Integer, fillValue as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shifts the array "rightwards".

Notes: If the array does not have any kNoValue data points, this method will shift the point at position "n" to "n + offset". On the "right" side of the array (the side with the largest index), points that are shifted outside the array will be discarded. On the "left" side of the array, fillValue data points will be shifted in.

If the array contains kNoValue data points, these points are not shifted. Conceptually, one can imagine the non-NoValue points being shifted to the next non-NoValue positions, and the process repeats offset number of times.

See also:

Parameter	Default	Description
offset	1	The number of positions to shift the array "rightwards."
fillValue	kNoValue	The new value to be shifted into the array.

- 3.4.65 shift(offset as Integer = 1)

141

3.4.67 stdDev as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the stdDev value of the elements of the CArrayMBS object.

Example:

```
dim data As New CArrayMBS(array(1.0, 2.0, 3.0, 4.0))
MsgBox str(data.stdDev) // shows 1.118034
```

3.4.68 subArray(value as CArrayMBS)

Plugin Version: 16.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Subtracts an array from the CArrayMBS object.

Notes: The array will be subtracted by subtracting each array element from the corresponding element of the CArrayMBS object.

Parameter	Default	Description
Values	(Mandatory)	A CArrayMBS with numbers to be subtracted from the CArrayMBS object.

See also:

- 3.4.69 subArray(value() as Double)

142

3.4.69 subArray(value() as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Subtracts an array from the CArrayMBS object.

Example:

```
dim src(-1) as Double = Array( 63.1, 10.15, 6.15, 2.88 )
dim data As New CArrayMBS(array(1.0, 1.0, 1.0, 1.0))
```

```

data.subArray( src )

dim lines(-1) as string

lines.Append str(data.count)+" values:"
lines.Append ""
lines.Append str(Data.getvalue(0))
lines.Append str(Data.getvalue(1))
lines.Append str(Data.getvalue(2))
lines.Append str(Data.getvalue(3))

MsgBox Join(lines,EndOfLine)

```

Notes: The array will be subtracted by subtracting each array element from the corresponding element of the CDArrayMBS object.

Parameter	Default	Description
b	(Mandatory)	An array of numbers to be subtracted from the CDArrayMBS object.

See also:

- 3.4.68 subArray(value as CDArrayMBS)

142

3.4.70 subValue(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Subtracts a number from every element of the CDArrayMBS object.

Example:

```

dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0))

data.subValue(5)

dim lines(-1) as string

lines.Append str(data.count)+" values:"
lines.Append ""
lines.Append str(Data.getvalue(0))
lines.Append str(Data.getvalue(1))
lines.Append str(Data.getvalue(2))
lines.Append str(Data.getvalue(3))

MsgBox Join(lines,EndOfLine)

```

Notes:

Parameter	Default	Description
b	(Mandatory)	A number to be subtracted from every element of the CDArrayMBS object.

3.4.71 sum as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the sum value of the elements of the CDArrayMBS object.

Example:

```
dim data As New CDArrayMBS(array(1.0, 2.0, 3.0, 4.0))
MsgBox str(data.sum) // shows 10.0
```

3.4.72 trim(startIndex as Integer = 0, len as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Trims the CDArrayMBS object by keeping only some elements in the middle.

Notes:

Parameter	Default	Description
startIndex	0	The index for the first element to keep.
len	-1	The number of elements to keep. -1 means keeping all elements from the

startIndex to the end of the array.

3.4.73 Values as Double()

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies all the values of this array object into a Xojo array.

Notes: On error the array returned is empty.

3.5 class CDAxisMBS

3.5.1 class CDAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Axis class represents x and y axes in XY charts.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [Charts with more than one x or y axis](#)
- [MBS Xojo Plugins, version 21.2pr1](#)
- [MBS Xojo Plugins, version 17.3pr3](#)
- [MBS Xojo / Real Studio Plugins, version 15.3pr1](#)
- [Exceptions and Private Constructors](#)

3.5.2 Methods

3.5.3 addLabel(pos as Double, label as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an extra label on the axis.

Notes:

Parameter	Default	Description
pos	(Mandatory)	The position on the axis to add the label.
label	(Mandatory)	The text label to add.

3.5.4 addMark(value as Double, lineColor as color, text as string = "", font as string = "", fontsize as Double = 8) as CDMarkMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addMark method, but uses color instead of integer data type for passing color values.

See also:

- [3.5.5 addMark\(value as Double, lineColor as Integer, text as string = "", font as string = "", fontsize as Double = 8\) as CDMarkMBS](#)

3.5.5 addMark(value as Double, lineColor as Integer, text as string = "", font as string = "", fontsize as Double = 8) as CDMarkMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a mark line to the chart.

Notes: A mark line is a line drawn on the plot area. This line is usually used to indicate some special values, such as a "target value", "threshold value", "target date", etc.

A mark line attached to the horizontal axis will be vertical across the plot area. A mark line drawn using the vertical axis will be horizontal across the plot area. In either case, the mark line label will be added to the axis at the mark line position.

The location of the mark line label can be changed by using `TextBox.setAlignment`. For example, by setting the alignment to `TopCenter`, the mark line label will be drawn on the top center of the mark line.

By default, the mark line is drawn at the front of the chart layers. You may change it to draw at the back of the plot area (that is, like grid lines) using `Mark.setDrawOnTop`.

Parameter	Default	Description
value	(Mandatory)	The value on the axis to draw the mark line.
lineColor	(Mandatory)	The color of the mark line.
text	""	The text label for the mark line. An empty string means there is no text label. By default, the text label and the tick on the axis will be drawn using the same color as the mark line. You can modify the colors by using the <code>Mark.setMarkColor</code> method.
font	""	The font used to draw the text label.
fontSize	8	The font size used to draw the text label in points.

See also:

- 3.5.4 `addMark(value as Double, lineColor as color, text as string = "", font as string = "", fontsize as Double = 8) as CDMarkMBS` 145

3.5.6 addZone(startValue as Double, endValue as Double, colorvalue as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addZone` method, but uses color instead of integer data type for passing color values.

See also:

- 3.5. *CLASS CDAXISMBS* 147
- 3.5.7 addZone(startValue as Double, endValue as Double, colorvalue as Integer) 147

3.5.7 addZone(startValue as Double, endValue as Double, colorvalue as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a zone to the chart.

Notes: A zone is a range of values. For example, "10 to 20" is a zone. Typically, a zones are used to classify data ranges. For example, you may classify 0 - 60 as the normal zone, 60 - 90 as the warning zone, and 90 - 100 as the critical zone.

A zone based on a horizontal axis will be drawn as a vertical band. A zone based on a vertical axis will be drawn as a horizontal band. Zones are always drawn at the back of the plot area.

Parameter	Default	Description
startValue	(Mandatory)	The start value (the lower bound) for the zone.
endValue	(Mandatory)	The end value (the upper bound) for the zone.
color	(Mandatory)	The color of the zone.

See also:

- 3.5.6 addZone(startValue as Double, endValue as Double, colorvalue as color) 146

3.5.8 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.5.9 copyAxis(axis as CDAxisMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies the scale and labels from another axis.

Notes: This method is typically used to ensure the axes on different charts are identical. For example, if multiple charts are draw and are stacked up, and you want the x-axes of the charts to be the same for easy comparison, you may copy the x-axis from one chart to the other charts.

The differences between copyAxis and Axis.syncAxis are:

Axis.syncAxis relates the axes with a linear relationship, so the axes may not be exact copies of one another.

Axis.syncAxis synchronizes axis scale and copies only scale related labels, such as the labels generated by auto-scaling, Axis.setLinearScale, Axis.setLogScale or Axis.setDateScale.

copyAxis copies all labels, including labels created using Axis.setLabels, which may be arbitrary text.

Parameter	Default	Description
axis	(Mandatory)	The axis to copy from.

3.5.10 getAlignment as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the side of the plot area that the axis is associated with.

Notes: This method is only applicable to XYChart objects.

3.5.11 getAxisImageMap(noOfSegments as Integer, mapWidth as Integer, url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates an HTML image map for the axis itself.

Notes: This method is similar to Axis.getHTMLImageMap. The difference is instead of generating an image map for the labels, it generates an image map for the axis itself. The axis will be divided into a number of segments, with an image map entry created for each segment.

3.5.12 getCoor(value as Double) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the geometric coordinates given the data value.

3.5.13 getFormattedLabel(v as Double, options as string = "") as string

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the label at the specified position on the axis, formatting one if necessary.

Notes:

Parameter	Default	Description
noOfSegments	(Mandatory)	The number of segments to divide the axis into.
mapWidth	(Mandatory)	The width of the axis used for the purpose of generating the image map.
url	(Mandatory)	The URL to be used in the "href" attribute of the image map. Parameter Substitution and Formatting is supported. Use an empty string if no href attribute is needed.
queryFormat	""	A text string representing the template of the query parameters to be appended to the URL. Parameter Substitution and Formatting is supported. The special keyword " { default } " represents the default query parameters. This is useful for specifying appending to the default. Note that an empty string means to use the default query query parameters. To specify no query parameter, use a space character.
extraAttr	""	A text string to specify additional attributes to add to the <area>tag. Parameter Substitution and Formatting is supported.
offsetX	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
offsetY	0	An offset to be added to all y coordinates in the image map. See offsetX above for description.
Argument	Default	Description
v	(Mandatory)	The position specified as a value on the axis scale.
formatString	""	The format string used to format a label if necessary. Please refer to Axis.setLabelFormat for the syntax of the format string. An empty string means the format will be automatically determined.

3.5.14 getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates an HTML image map for the axis labels.

Notes: This method should be called only after creating the chart image (eg. using BaseChart.makeChart). The image map cannot be determined without creating the chart image first.

This method accepts a URL as its argument. When generating an image map, it appends query parameters to the URL to indicate which legend entry the user has clicked.

The following is an example image map generated for an axis with 3 labels.

```
<area shape="rect" coords="30,220,70,239" href="handler.asp?value=0&label=John">
<area shape="rect" coords="70,220,110,239" href="handler.asp?value=1&label=Mary">
<area shape="rect" coords="110,220,150,239" href="handler.asp?value=2&label=Peter">
```

The image map consists of multiple `<area>` tags, one for each label. In the "href" attributes, query parameters are appended to the URL to provide information on the label clicked.

The image map produced by ChartDirector does not include the `<map>` and `</map>` tag. This is intentional so that you can add additional custom `<area>` tags to the image map, or append multiple image maps together.

The format of the appended URL parameters is determined using the `queryFormat` argument, which by default is:

```
value= { value } &label= { label }
```

The texts in curly brackets (i.e. `{ value }`, `{ label }`) will be replaced by the actual values when generating the image map. For example, `{ label }` will be replaced by the label text.

Please refer to Parameter Substitution and Formatting on all available parameters and how to format them.

In addition to customizing the query parameters, ChartDirector supports additional HTML attributes in the `<area>` tags using the `extraAttr` argument.

For example, the following `extraAttr` will add a "title" HTML attribute to every `<area>` tag. The "title" attribute will be displayed as "tool tip" when the mouse moves over the image map.

```
title='Click me for details on { label } '
```

Another common usage of the `extraAttr` argument is to add "onmouseover" and "onmouseout" HTML attributes to handle user interaction using Javascript on the browser.

Parameter	Default	Description
<code>url</code>	(Mandatory)	The URL to be used in the "href" attribute of the image map.

Parameter Substitution and Formatting is supported. Use an empty string if no href attribute is needed.

3.5.15 `getLabel(i as Double)` as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the label at the specified position on the axis.

Notes:

queryFormat	""	A text string representing the template of the query parameters to be appended to the URL. Parameter Substitution and Formatting is supported. The special keyword " { default } " represents the default query parameters. This is useful for specifying appending to the default. Note that an empty string means to use the default query query parameters. To specify no query parameter, use a space character.
extraAttr	""	A text string to specify additional attributes to add to the <area>tag. Parameter Substitution and Formatting is supported.
offsetX	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
offsetY	0	An offset to be added to all y coordinates in the image map. See offsetX above for description.

Parameter	Default	Description
i	(Mandatory)	The position specified as a value on the axis scale.

Return Value

Returns the label at the specified position, or a "" string if there is no label at that position.

3.5.16 getLabelTable as CDMLTableMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the CDML table created by CDAxisMBS.makeLabelTable.

3.5.17 getMaxValue as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the upper bound of the axis.

Notes: The upper bound of an axis is known only after auto-scaling. So this method should be called only after ChartDirector has finished auto-scaling (e.g. after calling BaseChart.layout to explicitly auto-scale the axis).

3.5.18 getMinValue as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the lower bound of the axis.

Notes: The lower bound of an axis is known only after auto-scaling. So this method should be called only after ChartDirector has finished auto-scaling (e.g. after calling BaseChart.layout to explicitly auto-scale the axis).

3.5.19 getThickness as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the thickness of the axis.

Notes: The axis can be considered as a line, with one side facing the internal of the plot area, and the other side not facing the plot area. By default, the latter side includes the axis labels and axis title (although ChartDirector allows the axis to be configured so that the labels and titles are internal to the plot area).

The thickness of an axis only refers to the thickness of the side not facing the plot area. For a vertical axis, it is the width of the bounding box of that side, including the axis ticks, labels and title if they are on that side. For a horizontal axis, it is the height of the bounding box.

The intention of this method is to allow the chart to be adjusted to leave enough space for the axis labels and title.

This method should be called only after axis layout (after calling CDXYChartMBS.layoutAxes, CDBaseChartMBS.layout or CDXYChartMBS.packPlotArea).

Arguments:

None

Return Value

The thickness of the axis in pixels.

3.5.20 getTicks as CDArrayMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the values of the ticks.

Notes: Return Value

An array of numbers representing the values of the ticks.

3.5.21 getX as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x-coordinate of starting point of the axis.

Notes: For an XYChart object, the starting point of a horizontal axis is its left end point, and the starting point of a vertical axis is its bottom end point. For a PolarChart object, the starting point of a radial axis is the origin. For other types of charts, the starting point of an axis is undefined.

3.5.22 getY as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y-coordinate of starting point of the axis.

Notes: For an XYChart object, the starting point of a horizontal axis is its left end point, and the starting point of a vertical axis is its bottom end point. For a PolarChart object, the starting point of a radial axis is the origin. For other types of charts, the starting point of an axis is undefined.

3.5.23 makeLabelTable as CDMLTableMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a CDML table and docks it to the axis, with one row (for horizontal axis) or column (for vertical axis) containing the axis labels.

Notes: Before calling this method, it is necessarily to set the labels on the axis first using CDAxisMBS.setLabels. You should only use this method on a label based axis with no label stepping.

This method will automatically indent the axis (see CDAxisMBS.setIndent), and set the tick offset to 0.5 (see CDAxisMBS.setTickOffset). This is to ensure the plot area grid lines align with the table grid lines (instead of align with the labels).

You may use the returned CDMLTableMBS object to insert additional rows and/or columns to the table. A common application of this method is to insert the data values to the table, so the chart will have a data table docked to the axis, aligned with the chart contents.

Arguments:

None

Return Value

A CDMLTable object containing the axis labels.

3.5.24 `setAngle`(angle as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the angular coordinates of the data points.

3.5.25 `setAutoScale`(topExtension as Double = 0.1, bottomExtension as Double = 0.1, zeroAffinity as Double = 0.8)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the margins at the two ends of the axis during auto-scaling, and whether to start the axis from zero.

Notes: During auto-scaling, it is often desirable to leave some margins at the ends of the axis. For example, suppose in a bar chart, the longest bar is 10 units. If auto-scaling chooses 0 - 10 as the scale, the longest bar will touch the top edge of the plot area. In many cases, the chart will look better if there is some margin so that the longest bar does not touch the top edge.

The `setAutoScale` can be used to reserve some margins at the ends of the axis by using a scale that is larger than necessary. For example, in the above case, if a scale of 0 - 12 is used, then the longest bar will not touch the top edge.

Other common reasons for reserving margins at the ends of the axis include making sure the data labels (which may be drawn on top of the data points) will not go outside the plot area, and that objects put at the top or bottom of the plot area (such as legend box and custom text box) will not overlap with the data points.

Note that there is an alternative way to reserve space at the ends of the axis - the `Axis.setMargin` method.

In the `setAutoScale` method, the amount of margins reserved is controlled by the `topExtension` and `bottomExtension` arguments. These arguments determine the portion of the axis where no data point can reach. For example, a `topExtension` of 0.2 will ensure no data point can fall within the top 20% of the axis.

Note that `ChartDirector` will not extend the scale across the 0 point.

For example, suppose the data range is 0.1 - 9.9. If 10% margin is added to the bottom end of the axis, the bottom end may become negative. In this case, `ChartDirector` will extend the bottom end to 0 at most.

In other words, if the data range is completely positive, `ChartDirector` will not extend the axis to negative, as it would be undesirable in most applications. The same applies if the data range is completely negative.

For a purely positive axis, the bottom end has "zero affinity". That means `ChartDirector` will tend to choose 0 as the bottom end because zero is a natural starting point for the axis. However, if the data range is too

extreme (e.g. the data is in the range 10000 - 10005), it may be "unreasonable" to choose 0 as the axis starting point. In this case, ChartDirector will not use 0 as the axis starting point.

ChartDirector will determine that it is "unreasonable" to use 0 as the axis starting point if the data fluctuation (the difference between the maximum and minimum data values) is too small compare with the data value. ChartDirector test the "too small" condition using the formula:

$$\text{maxDataValue} * \text{zeroAffinity} < \text{minDataValue}$$

where zeroAffinity by default is 0.8.

Similar "zero affinity" mechanism applies to the top end of the axis for a purely negative axis. If the data range contains both positive and negative values, the zero point is always included.

The zeroAffinity argument of the setAutoScale method allows you to modify the zero affinity when performing auto-scaling. Zero affinity should be between 0 and 1. A large value encourages ChartDirector to start the axis from zero.

A zero affinity of 1 means the axis always includes the zero point. A zero affinity of 0 means that the axis is scaled purely according to the data range, without any preference for the zero point.

Note that zero affinity is ignored for log scale axis as log scale axis cannot contain 0.

Parameter	Default	Description
topExtension	0.1	The top portion of the axis that no data point should fall into. For example, a value of 0.2 means no data value will fall within the top 20% of the axis. The topExtension must be between 0 to 1.
bottomExtension	0.1	The bottom portion of the axis that no data point should fall into. For example, a value of 0.2 means no data value will fall within the bottom 20% of the axis. The bottomExtension must be between 0 to 1.
zeroAffinity	0.8	The tendency of ChartDirector to include zero in the axis during auto-scaling. The value must be between 0 and 1. A large value encourages ChartDirector to start the axis from zero. A value of 1 means the axis always includes the zero point. A value of 0 means there is no preference for the zero point during auto-scaling.

3.5.26 setColors(axisColor as color, labelColor as color, titleColor as color, tickColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setColors` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.5.27 `setColors(axisColor as Integer, labelColor as Integer = &hfff0002, titleColor as Integer = -1, tickColor as Integer = -1)` 156

3.5.27 `setColors(axisColor as Integer, labelColor as Integer = &hfff0002, titleColor as Integer = -1, tickColor as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors of the axis itself, axis label, axis title and axis ticks.

Notes: By default, the axis and axis ticks are drawn using the `LineColor`, while the axis label and axis title are drawn using the `TextColor`. You may use this method to change their colors.

Parameter	Default	Description
<code>axisColor</code>	(Mandatory)	The color of the axis itself.
<code>labelColor</code>	<code>TextColor</code>	The color of the axis labels.
<code>titleColor</code>	-1	The color of the axis title. -1 means the axis title color is the same as the axis label color.
<code>tickColor</code>	-1	The color of the axis ticks. -1 means the axis ticks color is the same as the axis color.

See also:

- 3.5.26 `setColors(axisColor as color, labelColor as color, titleColor as color, tickColor as color)` 155

3.5.28 `setDateScale(formatString as string = "")`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use date auto-scale.

Notes:

Parameter	Default	Description
<code>formatString</code>	""	The format used for the labels on the axis. Please refer to <code>Axis.setLabelFormat</code> for the syntax of the format string. An empty string means the format will be automatically determined.

See also:

- 3.5.29 `setDateScale(lowerLimit as Double, upperLimit as Double, labels() as string)` 157

- 3.5.30 setDateScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0) 157

3.5.29 setDateScale(lowerLimit as Double, upperLimit as Double, labels() as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use the given date scale and the given labels.

Notes:

Parameter	Default	Description
lowerLimit	(Mandatory)	The lower bound of the axis, representing using one of the ChartDirector supported date format.
upperLimit	(Mandatory)	The upper bound of the axis, representing using one of the ChartDirector supported date format.
labels	(Mandatory)	An array of text strings to be used as the labels on the axis. ChartDirector will distribute the labels evenly on the axis. By default, all labels are associated with major ticks. These can be modified by using '.' or ' textasciitilde ' as the first character. Please refer to Axis.setLabels for details.

See also:

- 3.5.28 setDateScale(formatString as string = "") 156
- 3.5.30 setDateScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0) 157

3.5.30 setDateScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use the given date scale.

Notes:

The value 30 * 86400 will be assume to mean one month (which actually contains a variable number of seconds), and 60 * 86400 will be assumed to mean 2 months and so on. The value 360 * 86400 therefore means 12 months, or 1 year.

The default value of 0 means no minor tick is used.

See also:

- 3.5.28 setDateScale(formatString as string = "") 156
- 3.5.29 setDateScale(lowerLimit as Double, upperLimit as Double, labels() as string) 157

Parameter	Default	Description
lowerLimit	(Mandatory)	The lower bound of the axis, representing using one of the ChartDirector supported date format.
upperLimit	(Mandatory)	The upper bound of the axis, representing using one of the ChartDirector supported date format.
majorTickInc	0	Adds major ticks to the axis, where the major ticks are separated by majorTickInc seconds. Each major tick will have an associated text label for the value if the axis at the tick. The value 30 * 86400 will be assume to mean one month (which actually contains a variable number of seconds), and 60 * 86400 will be assumed to mean 2 months and so on. The value 360 * 86400 therefore means 12 months, or 1 year. The default value of 0 means the major ticks will be automatically determined. In this case, the lowerLimit and upperLimit may be automatically adjusted to align with the ticks. Use kNoValue to disable major ticks.
minorTickInc	0	Adds minor ticks to the x-axis, where the minor ticks are separated by minorTickInc seconds.

3.5.31 setFormatCondition(condition as string, operand as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Specifies the condition that subsequent Axis.setLabelFormat and Axis.setMultiFormat will become applicable.

Notes: In some applications, the axis range can vary greatly. For example, for a date/time axis, the ticks on the axis can be hourly ticks if the duration is short, or daily ticks (or even monthly or yearly) if the duration is long. One can always use a universal axis format that is applicable in all cases (such as "mmm dd, yyyy<*br*>hh:nn:ss"), or one can leave ChartDirector to automatically come up with a suitable axis format.

Sometimes it may be desirable to explicitly specify different axis formats depending on tick types. For example, one may want to specify a certain format if the ticks are hourly, and another format if the ticks are daily.

If it is possible to predict which kind of ticks will be on the axis, one can always use "if" statements to specify different axis formats for various cases.

The setFormatCondition method is for cases in which it is difficult to predict which type of ticks would appear on the axis. For example, in an auto-scaled axis in which the duration can vary continuously, in some marginal cases, it is difficult to predict if auto-scaling will choose hourly or daily ticks.

The setFormatCondition method allows you to specify a condition to be tested against the auto-scaling result, so that subsequent Axis.setLabelFormat or Axis.setMultiFormat will be applicable only if the condition is true.

The types of condition supported are:

Condition	Description
"align"	Test if all the ticks are aligned to the operand. Use 3600 for testing hourly alignment, 86400 (1 day = 86400) for daily alignment, 30 * 86400 for monthly alignment and 360 * 86400 for yearly alignment. (Note: The values for monthly and yearly alignments are special values recognized by ChartDirector. They works even if a month or year is not exactly 30 or 360 days.)
">"	Test if the maximum absolute value of the ticks are larger than the operand. This is useful when one needs to apply different formats to large numbers (eg. use scientific formats if the axis range exceed certain values).
">="	Test if the maximum absolute value of the ticks is larger than or equal to the operand.
"<"	Test if the maximum absolute value of the ticks is less than the operand.
"<="	Test if the maximum absolute value of the ticks is less than or equal to the operand.
"=="	Test if the maximum absolute value of the ticks is equal to the operand.
"="	Same as "==" above.
"!="	Test if the maximum absolute value of the ticks is not equal to the operand.
"<>"	Same as "!=" above.
"else"	This condition is true if and only if the previous condition is false.
"true"	This condition is always true. It is useful as a "catch all" condition.
"false"	This condition is always false.

Parameter	Default	Description
condition	(Mandatory)	The type of condition to test. Must be one of the condition types in the above table. This text string is case sensitive.
operand	0	The operand for condition types that need an operand.

3.5.32 setIndent(indent as boolean)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Specifies if the axis should be "indented" or not.

Notes: Normally, the x-axis is automatically scaled so that x coordinate of first data point is at the beginning of the x-axis, and the x coordinate last data point is at the end of the axis. If a line layer is drawn, the line will span from the left border of the plot area to the right border of the plot area.

However, for bar layer, if the x-axis is scaled as above, half of the first bar and half of the last bar will be outside the plot area. The same applies to HLOC layers, candlestick layers and box-whisker layers.

When the axis is "indented", some margins will be reserved at the ends of the axis, so that all data representation are within the plot area.

By default, "indented" mode is automatically used in x-axis for charts that contain bar, HLOC, candlestick or box-whisker layers.

The `setIndent` method allows you to manually configure whether "indented" mode is used or not. One common usage is to align the x-axes in different charts.

For example, suppose a web page contains a bar chart and a line chart. The bar chart will be using "indented" mode x-axis, while the line chart will be using "non-indented" mode. If the two charts are on top of one another for ease of comparison (common for finance style charts), it is desirable that their x-axes should align. In this case, the `setIndent` method can be used to force the line chart to use indented mode x-axis.

Parameter	Default	Description
<code>indent</code>	(Mandatory)	A true value sets the axis to indented mode. A false value sets the axis to non-indented mode.

3.5.33 `setLabelAlignment(alignment as Integer, minLabelSpace as Integer = 3)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the label alignment.

3.5.34 `setLabelFormat(formatString as string)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the format for numeric or date/time axis labels.

Example:

```
dim c as CDAxisMBS
```

```
// you can use label formats like this:
```

```
c.setLabelFormat("<*block,halign=left*><*font=timesbi.ttf,size=12,underline=1*>{ label } </font*><br*>US$  
{ value } K ( { percent } %)")
```

// we can reduce that to this:

```
c.setLabelFormat(" { label } { value } { percent } %")
```

// and it shows 3 numbers. With | 1 after the variable name, we define the decimals after dot:

```
c.setLabelFormat(" { label } { value | 1 } { percent | 1 } %")
```

// and

```
c.setLabelFormat(" { label } { value | 1., } { percent | 1., } %")
```

// uses dot for thousands and comma for decimal separator.

Notes: By default, ChartDirector will try to guess if the axis represents numbers or dates. If the axis represents numbers, it will use " { value } " as the default format. If the axis represents dates, it will guess the format based on resolution of the dates (e.g. whether the dates contain hourly data or monthly data, etc). It may use formats such as { value | mm/dd/yy<*br*>hh:nn:ss } , { value | mm/dd/yy hh:nn:ss } , { value | mm/dd/yyyy } , { value | mm/yyyy } or { value | yyyy } .

Please refer to Parameter Substitution and Formatting on all available parameters and how to format them.

Parameter	Default	Description
formatString	(Mandatory)	The format string.

3.5.35 setLabelGap(d as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the distance between the axis labels and the ticks on the axis.

Notes:

Parameter	Default	Description
d	(Mandatory)	The distance between the axis label and the tick in pixels.

3.5.36 setLabelOffset(offset as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shifts the axis labels from its default position along the axis.

Notes: This method expects the offset along the axis being a value on the axis scale. The Box.setPos

method of the axis label prototype (obtained using `Axis.setLabelStyle`) can also be used to shift the labels, with the offsets specified in pixel units.

Parameter	Default	Description
offset	(Mandatory)	The distance to shift the labels along the axis as a value on the axis scale.

3.5.37 `setLabels(labels() as Double, formatString as string = "") as CDTextBoxMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the numeric/date/time labels to be used on the axis.

Notes: This method is typically used to set the x-axis to enumerated scale. For more details on what is enumerated axis scale, please refer to `Axis.setLabels`.

This method assumes the labels are in their "native" form (that is, not formatted). Please refer to `Date Representation` for the native date/time formats supported in `ChartDirector`.

If the labels are already formatted into human readable form (that is, they are text strings), use `Axis.setLabels` instead.

One common issue is that there may be too many labels on the axis. In this case, the `Axis.setLabelStep` method may be used show only a regularly spaced subset of labels on the axis.

For date/time labels, another alternative is to use `Axis.setMultiFormat`, which uses filters to select important dates/times (such as dates/times representing the start of a month) for display as labels.

A third method to avoid too many labels is to remove some labels by replacing them with `kNoValue` before passing them to `ChartDirector`. If you want to remove the label text but leave a minor tick, use `MinorTickOnly` as the label value.

labels	(Mandatory)	An array of numbers/dates to be used as the axis labels.
formatString	""	A format string to specified how to format the labels into human readable form. Please refer to <code>Axis.setLabelFormat</code> for the syntax of the format string. An empty string means the format will be automatically determined.

Return Value

A `TextBox` object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See also:

- 3.5.38 setLabels(labels() as string) as CDTextBoxMBS

3.5.38 setLabels(labels() as string) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the text labels to be used on the axis.

Notes: This method is typically used to set the x-axis to enumerated scale. In enumerated scale, the data points are associated with the x-axis by position. The first data point will be plotted at the first label position on the x-axis, the second data point at the second label position, and so on.

Enumerated axis is a very flexible axis type. It is most suitable for chart types where the data points are evenly spaced on the x-axis. The axis labels can be any text. They do not need to be numbers or dates. If they are numbers or dates, you can format them in any way you like before calling this method.

Internally, ChartDirector will assign a value of 0 to the first axis label, 1 to the second axis label, and so on. These values are not visible. Only the axis labels are visible. However, these values may be useful for some ChartDirector features that need to reference the axis position by value, such as adding mark lines using `Axis.addMark`.

By default, all axis labels will be associated major ticks. To associate a label with a minor tick, use `'-'` as the first character of the label. To draw a label without any tick at all, use `'` textasciitilde `'` as the first character of the label.

Leading `'-'` or `'`

textasciitilde `'` characters are tick specification characters and will not appear on the labels. They just specify the tick style to be associated with the labels. If you want have a label that actually begins these characters, add `'\'` as the first character as the escape character.

One common issue is that there may be too many labels on the axis. In this case, the `Axis.setLabelStep` method may be used show only a regularly spaced subset of labels on the axis.

Another method is to remove some labels is to replace them with empty strings before passing them to ChartDirector. If you want to remove the label text but leave a major tick, use a space character `" "` as the label text. If you want to remove the label text but leave a major tick, use `"-"` as the label text.

Parameter	Default	Description
text	(Mandatory)	An array of strings containing the text of the labels.

Return Value

A `TextBox` object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See also:

- 3.5.37 `setLabels(labels() as Double, formatString as string = "") as CDTextBoxMBS` 162

3.5.39 `setLabelStep(majorTickStep as Integer, minorTickStep as Integer = 0, majorTickOffset as Integer = 0, minorTickOffset as Integer = -2147483647)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shows a regularly spaced subset of the axis labels on the axis.

Notes: This method is typically used in conjunction with

`Axis.setLabels`. These two methods define the full set of labels on the axis, one for each data point in a data set. In many cases, there may be too many labels and the axis may become overcrowded with labels. The `setLabelStep` method will cause the axis to show a regularly spaced subset of labels instead all labels.

Parameter	Default	Description
<code>majorTickStep</code>	(Mandatory)	The spacing between visible labels (major ticks). For example, a value of 10 means displaying 1 label for every 10 labels.
<code>minorTickStep</code>	0	For labels that are not displayed, ChartDirector can optionally put a minor tick in its place. The argument specifies the spacing between minor ticks. For example, a value of 5 means displaying 1 minor tick for every 5 labels. The default value of 0 means no minor tick is used.
<code>majorTickOffset</code>	0	The offset used for selecting the labels. For example, if <code>majorTickStep</code> is set to 10, by default, ChartDirector will select the labels with index 0, 10, 20, 30 and so on. If the <code>majorTickOffset</code> argument is set to 3, then ChartDirector will select labels with index 3, 13, 23, 33 and so on.
<code>minorTickOffset</code>	-7ffffff	The offset used for selecting minor tick points. The usage is the same as the <code>majorTickOffset</code> argument, except it applies to minor tick points. The default is to use the same value as <code>majorTickOffset</code> .

3.5.40 `setLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the font style used to for the axis labels.

Notes: See Font Specification for details on various font attributes.

Parameter	Default	Description
<code>font</code>	""	The font used to draw the labels.
<code>fontSize</code>	8	The font size used to draw the labels in points.
<code>fontColor</code>	<code>TextColor</code>	The color used to draw the labels.
<code>fontAngle</code>	0	The rotation angle of the labels.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.5.41 `setLabelStyle(font as string, fontsize as Double, fontcolor as color, fontAngle as Double = 0)` as CDTextBoxMBS 165

3.5.41 `setLabelStyle(font as string, fontsize as Double, fontcolor as color, fontAngle as Double = 0)` as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setLabelStyle` method, but uses color instead of integer data type for passing color values.

See also:

- 3.5.40 `setLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0)` as CDTextBoxMBS 164

3.5.42 `setLength(length as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the length of the axis.

3.5.43 `setLinearScale(formatString as string = "")`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use linear auto-scale.

Notes:

Parameter	Default	Description
<code>formatString</code>	<code>""</code>	The format used for the labels on the axis. Please refer to <code>Axis.setLabelFormat</code> for the syntax of the format string. An empty string means the format will be automatically determined.

See also:

- 3.5.44 `setLinearScale(lowerLimit as Double, upperLimit as Double, labels() as string)` 166
- 3.5.45 `setLinearScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)` 166

3.5.44 `setLinearScale(lowerLimit as Double, upperLimit as Double, labels() as string)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use the given linear scale and the given labels.

Notes:

Parameter	Default	Description
<code>lowerLimit</code>	(Mandatory)	The lower bound of the axis.
<code>upperLimit</code>	(Mandatory)	The upper bound of the axis.
<code>labels</code>	(Mandatory)	An array of text strings to be used as the labels on the axis. ChartDirector will distribute the labels evenly on the axis. By default, all labels are associated with major ticks. These can be modified by using <code>'</code> or <code>'</code> .

`textasciitilde` ' as the first character. Please refer to `Axis.setLabels` for details.

See also:

- 3.5.43 `setLinearScale(formatString as string = "")` 165
- 3.5.45 `setLinearScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)` 166

3.5.45 `setLinearScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use the given linear scale.

Notes:

Parameter	Default	Description
<code>lowerLimit</code>	(Mandatory)	The lower bound of the axis.
<code>upperLimit</code>	(Mandatory)	The upper bound of the axis.
<code>majorTickInc</code>	0	Adds major ticks to the axis, where the major ticks are separated by <code>majorTickInc</code> in value. Each major tick will have an associated text label for the value if the axis at the tick. The default value of 0 means the major ticks will be automatically determined. In this case, the <code>lowerLimit</code> and <code>upperLimit</code> may be automatically adjusted to align with the ticks. Use <code>kNoValue</code> to disable major ticks.
<code>minorTickInc</code>	0	Adds minor ticks to the axis, where the minor ticks are separated by <code>minorTickInc</code> in value. The default value of 0 means no minor tick is used.

See also:

- 3.5. CLASS CDAXISMBS 167
- 3.5.43 setLinearScale(formatString as string = "") 165
- 3.5.44 setLinearScale(lowerLimit as Double, upperLimit as Double, labels() as string) 166

3.5.46 setLogScale(formatString as string = "")

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use logarithmic auto-scale.

Notes:

Parameter	Default	Description
formatString	""	The format used for the labels on the axis. Please refer to Axis.setLabelFormat for the syntax of the format string. An empty string means the format will be automatically determined.

See also:

- 3.5.47 setLogScale(lowerLimit as Double, upperLimit as Double, labels() as string) 167
- 3.5.48 setLogScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0) 168

3.5.47 setLogScale(lowerLimit as Double, upperLimit as Double, labels() as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use the given logarithmic scale and the given labels.

Notes:

Parameter	Default	Description
lowerLimit	(Mandatory)	The lower bound of the axis.
upperLimit	(Mandatory)	The upper bound of the axis.
labels	(Mandatory)	An array of text strings to be used as the labels on the axis. ChartDirector will distribute the labels evenly on the axis. By default, all labels are associated with major ticks. These can be modified by using 'L' or 'l' as the first character. Please refer to Axis.setLabels for details.

See also:

- 3.5.46 setLogScale(formatString as string = "") 167
- 3.5.48 setLogScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0) 168

3.5.48 `setLogScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis to use the given logarithmic scale.

Notes:

Parameter	Default	Description
<code>lowerLimit</code>	(Mandatory)	The lower bound of the axis.
<code>upperLimit</code>	(Mandatory)	The upper bound of the axis.
<code>majorTickInc</code>	0	Adds major ticks to the axis, where the major ticks are separated by <code>majorTickInc</code> in ratio. For example, a value of 10 means each tick will be 10 times the value of the previous tick. Each major tick will have an associated text label for the value if the axis at the tick. The special predefined constant <code>LogTick</code> (= 1.6e308) means the tick increment will be using the non-regular ratio 1, 2, 5, 10, 20, 50, 100, The default value of 0 means the major ticks will be automatically determined. In this case, the <code>lowerLimit</code> and <code>upperLimit</code> may be automatically adjusted to align with the ticks. Use <code>kNoValue</code> to disable major ticks.
<code>minorTickInc</code>	0	Adds minor ticks to the axis, where the minor ticks are separated by <code>minorTickInc</code> in ratio.

The special constant `LogTick` means the tick increment will be using the non-regular ratio 1, 2, 5, 10, 20, 50, 100,

The default value of 0 means no minor tick is used.

See also:

- 3.5.46 `setLogScale(formatString as string = "")` 167
- 3.5.47 `setLogScale(lowerLimit as Double, upperLimit as Double, labels() as string)` 167

3.5.49 `setMargin(topMargin as Integer, bottomMargin as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reserve margins at the ends of the axis.

Notes: This method is usually used to reserve space in the plot area. For example, if some margin is reserved at the top of the vertical y-axis, the top of the plot area will contain no data points. It is because this region would be outside the active range of the y-axis. If anything is put at the top of the plot area (such as a legend box or custom text), it will not overlap with the data points.

Note if auto-scaling is used, there is an alternative way to reserve space at the top and/or bottom of the plot area - the `Axis.setAutoScale` method.

Parameter	Default	Description
topMargin	(Mandatory)	The margin reserved at the top end (or right end for horizontal axis) of the axis in pixels.
bottomMargin	0	The margin reserved at the bottom end (or left end for horizontal axis) of the axis in pixels.

3.5.50 setMinTickInc(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the minimum distance between two ticks on the axis for auto-scaled axis.

Notes: The most common use of this method is to ensure the ticks are of integer values (the minimum distance set to 1).

ChartDirector auto-scaling will automatically determine the optimal number of ticks and labels on the axis. However, in some cases, the nature of the data may be such that the ticks should assume only certain discrete values (such as must be integers). This method will inform ChartDirector for this constraint when performing auto-scaling.

For a log scale axis, because the axis ticks are unevenly spaced, the minimum distance will be handled as the minimum value allowed for the axis scale.

Parameter	Default	Description
inc	(Mandatory)	The minimum distance between two ticks on the axis as a data value.

3.5.51 setMultiFormat(filter as Integer, format as string, labelSpan as Integer = 1, promoteFirst as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds one filter and format string to the multi-format lists.

Notes: Please refer to Axis.setMultiFormat on how to use this method.

Parameter	Default	Description
filterId	(Mandatory)	The filter that defines a subset of labels.
formatString	(Mandatory)	The format string for formatting the subset defined by the above filter.
labelSpan	1	The number of label positions that are claimed by one label. If a label occupies more than 1 position, ChartDirector will not put labels on nearby positions, even if they meet the filter criteria.
promoteFirst	true	If set to true, the first label will be promoted to a higher filter category in the multi-format list.

See also:

- 3.5.52 `setMultiFormat(filter1 as Integer, format1 as string, filter2 as Integer, format2 as string, labelSpan as Integer = 1, promoteFirst as boolean=true)` 170

3.5.52 `setMultiFormat(filter1 as Integer, format1 as string, filter2 as Integer, format2 as string, labelSpan as Integer = 1, promoteFirst as boolean=true)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets multiple formats for numeric or date/time axis labels.

Notes: This method is typically used to format date/time labels. For example, for an axis that spans 90 days, this method can be used to format the first labels of each month in bold as "mmm-dd", while other labels are shown in normal font as "dd".

The multiple formats are defined as a list of filters and format strings. If the label value satisfies the first filter, it will be formatted using the first format string. Otherwise if the label value satisfies the second filter, it will be formatted using the second format string, and so on. If a label value does not satisfy any filter, it will be discarded.

In the above example, the first filter can be a "start of month" filter (created using `Chart::StartOfMonthFilter`), and the first format string can be " { value | mmm-dd } ". The second filter can be the "all pass" filter (created using `Chart::AllPassFilter`), and the format string can be " { value | dd } ".

Supported filters in `ChartDirector` include:

Filter	Description
<code>Chart::StartOfHourFilter</code>	Creates a data filter that matches date/times that represent the start of a new hour in a date/time series.
<code>Chart::StartOfDayFilter</code>	Creates a data filter that matches date/times that represent the start of a new day in a date/time series.
<code>Chart::StartOfWeekFilter</code>	Creates a data filter that matches date/times that represent the start of a new week in a date/time series.
<code>Chart::StartOfMonthFilter</code>	Creates a data filter that matches date/times that represent the start of a new month in a date/time series.
<code>Chart::StartOfYearFilter</code>	Creates a data filter that matches date/times that represent the start of a new year in a date/time series.
<code>Chart::RegularSpacingFilter</code>	Creates a data filter that matches 1 out of every N elements.
<code>Chart::AllPassFilter</code>	Creates a data filter that matches every element.
<code>Chart::NonePassFilter</code>	Creates a data filter that matches no element.
<code>Chart::SelectedItemFilter</code>	Creates a data filter that matches one specified item.

For the format strings, please refer to [Parameter Substitution and Formatting](#) on their syntax.

A single `setMultiFormat` method supports two filters and two format strings. You can use multiple `setMultiFormat` methods to add more filters and format strings.

One common issue in putting labels on the axis is that there may be too many labels on the axis. The `setMultiFormat` method supports a `labelSpan` argument that specifies how many label positions a single label will occupy. If a label occupies more than 1 position, `ChartDirector` will not put labels on nearby positions, even if they meet the filter criteria.

The `setMultiFormat` method supports special handling of the first label on the axis. For example, in the above example, we may want to show the the first label as "mmm-dd", even if it is not actually the "start of month". The `promoteFirst` argument, if set to true, will cause the first label to be promoted to a higher filter category in the multi-format list. For example, if the first label satisfies only the second filter, it will be formatted as if it satisfies the first filter.

Parameter	Default	Description
<code>filter1</code>	(Mandatory)	The filter that defines the first kind of labels.
<code>format1</code>	(Mandatory)	The format string for the first kind of labels.
<code>filter2</code>	(Mandatory)	The filter that defines the second kind of labels.
<code>format2</code>	(Mandatory)	The format string for the second kind of labels.
<code>labelSpan</code>	1	The number of label positions that are claimed by one label. If a label occupies more than 1 position, <code>ChartDirector</code> will not put labels on nearby positions, even if they meet the filter criteria.
<code>promoteFirst</code>	true	If set to true, the first label will be promoted to a higher filter category in the multi-format list.

See also:

- 3.5.51 `setMultiFormat(filter as Integer, format as string, labelSpan as Integer = 1, promoteFirst as boolean=true)` 169

3.5.53 `setOffset(x as Integer, y as Integer)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the positional offset of the axis.

Notes: By default, `ChartDirector` draws horizontal axes at the top/bottom borders of the plot area, and vertical axes at the left/right borders of the plot area.

This method can be used to shift an axis from its standard position.

Arguments:

Argument	Default	Description
x	(Mandatory)	The x offset in pixels.
y	(Mandatory)	The y offset in pixels.

3.5.54 setReverse(value as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reverse the axis.

Notes: For a normal vertical axis, the axis starts from the bottom and increase its value towards the top. For a normal horizontal axis, the axis starts from the left and increase its value towards the right. The setReverse method can be used to reverse the direction of the axis.

Parameter	Default	Description
b	true	A true value means the axis is reversed. A false value means the axis is not reversed.

3.5.55 setRounding(roundMin as boolean, roundMax as boolean)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Controls whether to round the ends of the axis to align with tick positions.

Notes: For example, if the axis is from 0.33 - 9.7, ChartDirector may round it to 0 - 10 so that the ends 0 and 10 are properly aligned with the ticks.

By default, ChartDirector will round the axis ends for the y-axis, but not for the x-axis. An exception is a chart containing a scatter layer, where both x and y axes will be rounded.

Parameter	Default	Description
roundMin	(Mandatory)	A true value means the lesser end of the axis should be rounded to align with tick positions. A false value means no rounding.
roundMax	(Mandatory)	A true value means the greater end of the axis should be rounded to align with tick positions. A false value means no rounding.

3.5.56 setTickColor(majorTickColor as color, minorTickColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setTickColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.5.57 `setTickColor(majorTickColor as Integer, minorTickColor as Integer = -1)` 173

3.5.57 `setTickColor(majorTickColor as Integer, minorTickColor as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors of the axis ticks.

Notes:

Parameter	Default	Description
<code>majorTickColor</code>	(Mandatory)	The color of the major ticks.
<code>minorTickColor</code>	-1	The color of the major ticks. -1 means the color is the same as <code>majorTickColor</code>

See also:

- 3.5.56 `setTickColor(majorTickColor as color, minorTickColor as color)` 172

3.5.58 `setTickDensity(majorTickSpacing as Integer, minorTickSpacing as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the density of the axis ticks.

Notes: Tick density is the desired distance between two ticks in pixels. When `ChartDirector` performs auto-scaling, it will decide how many ticks to put on the axis based on tick density.

Note that the actual tick density chosen in auto-scaling may not be exactly the same as the desired tick density. It is because `ChartDirector` may have other constraints in choosing the ticks, such as the ticks and axis range should be neat numbers, and the axis must contain an integral number of ticks, etc. `ChartDirector` may use a tick distance that is larger than suggested, but never smaller.

Parameter	Default	Description
<code>majorTickSpacing</code>	(Mandatory)	Specify the desired distance between two major ticks in pixels.
<code>minorTickSpacing</code>	-1	Specify the desired distance between two minor ticks in pixels. -1 means minor ticks are not used.

3.5.59 `setTickLength(majorTickLen as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the length of the axis ticks.

Notes: A positive tick length means the ticks are drawn outside the plot area. A negative tick length means

the ticks are drawn inside the plot area.

Parameter	Default	Description
majorTickLen	(Mandatory)	The length of the major ticks in pixels. The length of the minor ticks will automatically be set to half the length of the major ticks.

See also:

- 3.5.60 `setTickLength(majorTickLen as Integer, minorTickLen as Integer)` 174

3.5.60 `setTickLength(majorTickLen as Integer, minorTickLen as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the length of the major and minor axis ticks.

Notes: A positive tick length means the ticks are drawn outside the plot area. A negative tick length means the ticks are drawn inside the plot area.

Parameter	Default	Description
majorTickLen	(Mandatory)	The length of the major ticks in pixels.
minorTickLen	(Mandatory)	The length of the minor ticks in pixels.

See also:

- 3.5.59 `setTickLength(majorTickLen as Integer)` 173

3.5.61 `setTickOffset(offset as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shifts the position of the ticks along the axis.

Notes: By default, ChartDirector draws the ticks at the label position. This method is typically used to shift the ticks so that the ticks are between two labels, rather than aligned with the label.

The offset is specified as a value on the axis scale. For a label based axis (configured using `Axis.setLabels`), shifting with an offset of 0.5 means shifting half the label interval. This will achieve the effect of putting the ticks in between the labels.

Parameter	Default	Description
offset	(Mandatory)	The distance to shift the ticks along the axis as a value on the axis scale.

3.5.62 setTickWidth(majorTickWidth as Integer, minorTickWidth as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the width of the axis ticks.

Notes:

Parameter	Default	Description
majorTickWidth	(Mandatory)	The width of the major ticks in pixels.
minorTickWidth	-1	The width of the minor ticks in pixels. -1 means the width is the same as majorTickWidth.

3.5.63 setTitle(text as string, font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a title to the axis.

Notes: See Font Specification for details on various font attributes.

Parameter	Default	Description
text	(Mandatory)	The title text.
font	""	The font used to draw the title. If no font is specified, the default is "bold".
fontSize	8	The size of the font in points.
fontColor	TextColor	The color used to draw the title text.

Return Value

A TextBox object representing the axis title. This may be used to fine-tune the appearance of the axis title.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.5.64 setTitle(text as string, font as string, fontsize as Double, fontcolor as color) as CDTextBoxMBS

175

3.5.64 setTitle(text as string, font as string, fontsize as Double, fontcolor as color) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setTitle method, but uses color instead of integer data type for passing color values.

See also:

- 3.5.63 setTitle(text as string, font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002) as CDTextBoxMBS 175

3.5.65 setTitlePos(alignment as Integer, titleGap as Integer = 3)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position of the axis title relative to the axis.

Notes: By default, the axis title will be drawn at the middle of the axis outside the plot area. You may change the location of the title. For example, instead of drawing the y-axis title at the middle of the axis, you may want draw it at the top of the axis.

Parameter	Default	Description
alignment	(Mandatory)	The position of the title relative to the axis.
titleGap	3	The distance between the axis title and the axis in pixels.

3.5.66 setWidth(width as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the line width of the axis.

Notes:

Parameter	Default	Description
width	(Mandatory)	The line width of the axis in pixels.

3.5.67 syncAxis(axis as CDAxisMBS, slope as Double = 1.0, intercept as Double = 0.0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Synchronizes this axis with another axis using a linear formula.

Notes: This method is typically used if the two axes represent the same quantity but in different units. For example, one axis may represent temperature in Celsius, and the other in Fahrenheit, or they may represent lengths in meters and feet.

The scale of this axis (value) will be related to the scale of the another axis (value2) using the following formula:

$$\text{value} = \text{value2} * \text{slope} + \text{intercept}$$

Typically, one of the axis will be for actual charting, and its scale will be determined in the standard way using auto or manual scaling. Then the second axis is set to synchronize with the first axis.

Parameter	Default	Description
axis	(Mandatory)	The axis to synchronize to.
slope	1	The slope for synchronizing this axis to the other axis.
intercept	0	The intercept parameter for synchronizing this axis to the other axis.

3.5.68 syncScale(axis as CDAxisMBS = nil, slope as double = 1.0, intercept as double = 0.0)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Synchronizes the scale of this axis with another axis using a linear formula.

Notes: The scale of this axis (value) will be related to the scale of the source axis (value2) using the following formula:

$$\text{value} = \text{value2} * \text{slope} + \text{intercept}$$

The differences between syncScale and Axis.syncAxis are:

- Axis.syncAxis assumes the two axes are of the same length. In addition to the scale, it will also synchronize the ticks and labels, and configurations that may affect the positions of ticks and labels, such as axis margins and axis indentation. This ensures the two axes scale align properly.
- syncScale does not assume the two axes are of the same length. For example, the y-axis of one chart can be synchronized with the CDColorAxisMBS of a different chart. The two axes can have different labels to fit their different lengths.

Argument	Default	Description
axis	(Mandatory)	The axis to synchronize to. A value of null means not to synchronize to any axis.
slope	1	The slope for synchronizing this axis to the other axis.
intercept	0	The intercept parameter for synchronizing this axis to the other axis.

3.6 class CDBarLayerMBS

3.6.1 class CDBarLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The BarLayer class represents bar layers. BarLayer is a subclass of Layer.

Notes: Subclass of the CDLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [Create a Bar Chart in Xojo with the ChartDirector](#)
- [Chart Diagrams with Xojo](#)
- [Beware of the plugin limit in Real Studio](#)

Xojo Developer Magazine

- [7.4, page 31: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [17.2, page 37: More Beyond JSON, Develop APIs That Generate Barcodes, Charts, and More by Timothy Dietrich](#)
- [17.2, page 35: More Beyond JSON, Develop APIs That Generate Barcodes, Charts, and More by Timothy Dietrich](#)
- [17.2, page 33: More Beyond JSON, Develop APIs That Generate Barcodes, Charts, and More by Timothy Dietrich](#)

3.6.2 Methods

3.6.3 setBarGap(barGap as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gap between the bars in a bar chart layer.

Notes: In the setBarGap method, the gap between the bars is expressed as the portion of the space between the bars. For example, a bar gap of 0.2 means 20% of the distance between two adjacent bars is the gap between the bars.

A bar gap of 0 means there is no gap in between the bars. The bar gap can be negative. In this case, the bars will overlap. The predefined constant TouchBar (= -1.7E-100) means the bars will "touch", that is, the bar borders overlap, so two adjacent bars will share the same border.

For multi-bar layers (bar layers using the Side data combine method, or for stacked bar layers with multiple data groups), `barGap` refers to the portion of the space between bar groups, while `subBarGap` refers to the portion of the space between bars within the bar group.

Parameter	Default	Description
<code>barGap</code>	(Mandatory)	The portion of the space between the bars (or between bar groups for multi-bar layers).
<code>subBarGap</code>	0.2	This argument only applies to multi-bar charts. It is the portion of the space between the bars in a bar group.

See also:

- 3.6.4 `setBarGap(barGap as Double, subBarGap as Double)`

179

3.6.4 `setBarGap(barGap as Double, subBarGap as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gap between the bars in a bar chart layer.

Notes: In the `setBarGap` method, the gap between the bars is expressed as the portion of the space between the bars. For example, a bar gap of 0.2 means 20% of the distance between two adjacent bars is the gap between the bars.

A bar gap of 0 means there is no gap in between the bars. The bar gap can be negative. In this case, the bars will overlap. The predefined constant `TouchBar` (`= -1.7E-100`) means the bars will "touch", that is, the bar borders overlap, so two adjacent bars will share the same border.

For multi-bar layers (bar layers using the Side data combine method, or for stacked bar layers with multiple data groups), `barGap` refers to the portion of the space between bar groups, while `subBarGap` refers to the portion of the space between bars within the bar group.

Parameter	Default	Description
<code>barGap</code>	(Mandatory)	The portion of the space between the bars (or between bar groups for multi-bar layers).
<code>subBarGap</code>	0.2	This argument only applies to multi-bar charts. It is the portion of the space between the bars in a bar group.

See also:

- 3.6.3 `setBarGap(barGap as Double)`

178

3.6.5 `setBarShape(shape as Integer, dataGroup as Integer = -1, dataItem as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of the bar(s) using built-in shapes.

Notes:

Parameter	Default	Description
shape	(Mandatory)	The bar shape to use. Please refer to Shape Specification on how shapes are specified.
dataGroup	-1	The index of the data group that the shape applies to. In a multi-stacked bar, the data sets that are stacked up into a single bar forms a data group. In a multi-bar chart, each data set is a data group by itself. The default value of -1 means the shape applies to all data groups.
dataItem	-1	The index of the data points that the shape applies to. For example, if set to 3, the shape will only apply to the 4th bar (index starts from 0) in each data series. The default value of -1 means the shape applies to all data points.

See also:

- 3.6.6 `setBarShape(shape() as Integer, dataGroup as Integer = -1, dataItem as Integer = -1)` 180

3.6.6 `setBarShape(shape() as Integer, dataGroup as Integer = -1, dataItem as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of the bar(s) to a custom shape defined using an array of (x, y) coordinates.

Notes:

Parameter	Default	Description
shape	(Mandatory)	The bar shape to use. Please refer to Shape Specification on how shapes are specified.
dataGroup	-1	The index of the data group that the shape applies to. In a multi-stacked bar, the data sets that are stacked up into a single bar forms a data group. In a multi-bar chart, each data set is a data group by itself. The default value of -1 means the shape applies to all data groups.
dataItem	-1	The index of the data points that the shape applies to. For example, if set to 3, the shape will only apply to the 4th bar (index starts from 0) in each data series. The default value of -1 means the shape applies to all data points.

See also:

- 3.6.5 `setBarShape(shape as Integer, dataGroup as Integer = -1, dataItem as Integer = -1)` 180

3.6.7 setBarWidth(barWidth as Integer, subBarWidth as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the width of the bars.

Notes:

Parameter	Default	Description
barWidth	(Mandatory)	The width of the bars (or bar groups for multi-bar layers) in pixels.
subBarWidth	-1	This argument only applies to multi-bar charts. It is the width of the bars in a bar group in pixels.

3.6.8 setIconSize(height as Integer, width as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the icon(s) to be used in legend box.

Notes: By default, if a legend box is available on the chart, ChartDirector will insert an legend entry for any named data sets in the legend box. The icons for the data sets will be the shape of the bars. The size of the icons will be determined using the key size settings of the legend box (see `LegendBox.setKeySize`).

This method can be used to override the legend box settings to specify a custom width/height for the icons of the current BarLayer.

If the icon size is set to 0, a square color box will be used as the icon without regarding the shape of the bar.

Parameter	Default	Description
height	(Mandatory)	The height of the icon in pixels.
width	-1	The width of the icon in pixels. The default value of -1 means the width is the same as the height.

3.6.9 setMinImageMapSize(s as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the minimum height (or width if the x-axis and y-axis are swapped with `CDXYChartMBS.swapXY`) of the hot spots defined by the image map.

Notes: By default, the hot spots defined by image map exactly cover the bars. For very short bars, the hot spots may be very small, and it may become difficult to move the mouse over the hot spots.

This method ensures the hot spots are at least of certain size, even if the bars are shorter. The default is 5 pixels.

Arguments:

Argument	Default	Description
size	(Mandatory)	The minimum height (or width if the x-axis and y-axis are swapped with <code>CDXYChartMBS.swapXY</code>) of the hot spots defined by the image map.

3.6.10 `setMinLabelSize(s as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the minimum height (or width for horizontal bars) of the bar segments below which data labels will be hidden.

Notes: In `ChartDirector`, for a bar layer, data labels (`Layer.setDataLabelStyle`) are drawn internal to the bar segment, while the aggregate labels (`Layer.setAggregateLabelStyle`) are drawn external to the bar.

`ChartDirector` will disable data labels for a bar segment if the bar segment is too short to contain the data label.

By default, `ChartDirector` will automatically determine what is meant by "too short". The `setMinLabelSize` method can be used to manually defined the threshold for "too short".

Sometimes it may be desirable to display the data label even though it cannot be contained within the bar segment. In this case, the `setMinLabelSize` can be used to set the threshold to 0.

Parameter	Default	Description
s	(Mandatory)	The minimum length of the bar segments in pixels, below which data labels will be hidden.

3.6.11 `setOverlapRatio(overlapRatio as Double, firstOnTop as boolean=true)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets overlapping among bars in within a bar group in a multi-bar chart.

Notes: By default, in a multi-bar chart, the bars are not overlapped, but are separated with a gap in between. This method can be used to set an alternative style, where the bars partially overlap.

3.6.12 `setRoundedCorners`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Parameter	Default	Description
overlapRatio	(Mandatory)	The overlapping ratio between adjacent bars in a bar group. Should be between 0 to 1.
firstOnTop	true	If this value is true, the first data set will be on top of the second data during overlapping, and the second data set will be on top of the third data set, and so on. If this value is false, then the overlapping order will be reversed.

Function: Configures rounded corners for rectangular bars.

Notes: A rectangular bar has 4 corners. Two of them are at the base of the bar, while the other two are at the "data end" of the bar. Note that the base of the bar can be any side of the bar, depending on whether the bar is positive or negative, vertical or horizontal.

For the 2 base corners, one of them is at the negative x direction while the other is at the positive x direction. The same applies to the two data end corners.

This method accepts 0, 1, 2 or 4 radius values. The usage of these values depend how many values are provided as follows.

Values Provided	Description
0	The radii of the data end corners are automatically determined. The radii of the base corners are set to 0.
1	The radii of the data end corners are set to the specified value. The radii of the base corners are set to 0.
2	The radii of the data end corners are set to the first value. The radii of the base corners are set to the second value.
4	The radius of the data end corner at the negative x direction is set to the first value. The radius of the data end corner at the positive x direction is set to the second value. The radius of the base corner at the negative x direction is set to the third value. The radius of the base corner at the positive x direction is set to the fourth value.

Argument	Default	Description
r1	-\&h7ffffff	The first radius. See description above on how this radius value is used.
r2	-\&h7ffffff	The second radius. See description above on how this radius value is used.
r3	-\&h7ffffff	The third radius. See description above on how this radius value is used.
r4	-\&h7ffffff	The fourth radius. See description above on how this radius value is used.

See also:

- 3.6.13 setRoundedCorners(r1 as Integer, r2 as Integer = -2147483647, r3 as Integer = -2147483647, r4 as Integer = -2147483647) 184

3.6.13 setRoundedCorners(r1 as Integer, r2 as Integer = -2147483647, r3 as Integer = -2147483647, r4 as Integer = -2147483647)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures rounded corners for rectangular bars.

Notes: A rectangular bar has 4 corners. Two of them are at the base of the bar, while the other two are at the "data end" of the bar. Note that the base of the bar can be any side of the bar, depending on whether the bar is positive or negative, vertical or horizontal.

For the 2 base corners, one of them is at the negative x direction while the other is at the positive x direction. The same applies to the two data end corners.

This method accepts 0, 1, 2 or 4 radius values. The usage of these values depend how many values are provided as follows.

Values Provided	Description
0	The radii of the data end corners are automatically determined. The radii of the base corners are set to 0.
1	The radii of the data end corners are set to the specified value. The radii of the base corners are set to 0.
2	The radii of the data end corners are set to the first value. The radii of the base corners are set to the second value.
4	The radius of the data end corner at the negative x direction is set to the first value. The radius of the data end corner at the positive x direction is set to the second value. The radius of the base corner at the negative x direction is set to the third value. The radius of the base corner at the positive x direction is set to the fourth value.

Argument	Default	Description
r1	-&h7ffffff	The first radius. See description above on how this radius value is used.
r2	-&h7ffffff	The second radius. See description above on how this radius value is used.
r3	-&h7ffffff	The third radius. See description above on how this radius value is used.
r4	-&h7ffffff	The fourth radius. See description above on how this radius value is used.

See also:

- 3.6.12 setRoundedCorners

3.7 class CDBaseBoxLayerMBS

3.7.1 class CDBaseBoxLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function:

The BaseBoxLayer class is the base class for layers that uses isolated symbols to represent multiple data sets. These include BoxWhiskerLayer, HLOCLayer and CandleStickLayer.

Notes:

Subclass of the CDLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.7.2 Methods

3.7.3 setDataGap(gap as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the symbol width by specifying the gap ratio between adjacent symbols.

Notes:

Parameter	Default	Description
gap	(Mandatory)	The gap between two adjacent symbols as the portion of the space between the midpoints of the elements. The gap must be in the range 0 - 1. A value of 0 (the default) means there is no gap between two adjacent symbols.

3.7.4 setDataWidth(width as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the symbol width by specifying the pixel width of the symbols.

Notes:

Parameter	Default	Description
width	(Mandatory)	The width of the symbols in pixels.

3.7.5 setMinImageMapSize(size as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the minimum height (or width if the x-axis and y-axis are swapped with `CDXYChartMBS.swapXY`) of the hot spots defined by the image map.

Notes: By default, the hot spots defined by image map exactly cover the bars. For very short bars, the hot spots may be very small, and it may become difficult to move the mouse over the hot spots.

This method ensures the hot spots are at least of certain size, even if the bars are shorter. The default is 5 pixels.

Arguments:

Argument	Default	Description
size	(Mandatory)	The minimum height (or width if the x-axis and y-axis are swapped with <code>CDXYChartMBS.swapXY</code>) of the hot spots defined by the image map.

3.7.6 setRoundedCorners(r1 as Integer)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures rounded corners for the boxes.

Notes: This method currently is only supported by the `BoxWhiskerLayer`.

A box has 4 corners, two of them are at "box-top", and two of them are at "box-bottom". Note that "box-top" and "bottom-bottom" refer to the sides of the box representing the `boxTop` and `boxBottom` parameters in `CDXYChartMBS.addBoxWhiskerLayer` or `CDXYChartMBS.addBoxLayer`. Note that they are not necessarily the geometric top or bottom of the box. For example, if the y-axis is reversed (such as configured using `CDAxisMBS.setReverse`), the top and bottom side may be reversed. For horizontal boxes (with `CDXYChartMBS.swapXY` in effect), the box-top and box-bottom sides may actually be the right and left sides.

Of the 2 box-top corners, one of them is at the negative x direction while the other is at the positive x direction. The same applies to the box-bottom corners.

This method accepts 0, 1, 2 or 4 radius values. The usage of these values depend how many values are provided as follows.

See also:

- 3.7.7 `setRoundedCorners(r1 as Integer, r2 as Integer, r3 as Integer = -2147483647, r4 as Integer = -2147483647)` 187

Values Provided	Description
0	The radii of all the box corners are automatically determined.
1	The radii of all the box corners are set to the specified value.
2	The radii of the box-top corners are set to the first value. The radii of the box-bottom corners are set to the second value.
4	The radius of the box-top corner at the negative x direction is set to the first value. The radius of the box-top corner at the positive x direction is set to the second value. The radius of the box-bottom corner at the negative x direction is set to the third value. The radius of the box-bottom corner at the positive x direction is set to the fourth value.

Argument	Default	Description
r1	-0x7ffffff	The first radius. See description above on how this radius value is used.
r2	-0x7ffffff	The second radius. See description above on how this radius value is used.
r3	-0x7ffffff	The third radius. See description above on how this radius value is used.
r4	-0x7ffffff	The fourth radius. See description above on how this radius value is used.

3.7.7 setRoundedCorners(r1 as Integer, r2 as Integer, r3 as Integer = -2147483647, r4 as Integer = -2147483647)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures rounded corners for the boxes.

Notes: This method currently is only supported by the BoxWhiskerLayer.

A box has 4 corners, two of them are at "box-top", and two of them are at "box-bottom". Note that "box-top" and "bottom-bottom" refer to the sides of the box representing the boxTop and boxBottom parameters in CDXYChartMBS.addBoxWhiskerLayer or CDXYChartMBS.addBoxLayer. Note that they are not necessarily the geometric top or bottom of the box. For example, if the y-axis is reversed (such as configured using CDAxisMBS.setReverse), the top and bottom side may be reversed. For horizontal boxes (with CDXYChartMBS.swapXY in effect), the box-top and box-bottom sides may actually be the right and left sides.

Of the 2 box-top corners, one of them is at the negative x direction while the other is at the positive x direction. The same applies to the box-bottom corners.

This method accepts 0, 1, 2 or 4 radius values. The usage of these values depend how many values are provided as follows.

See also:

- 3.7.6 setRoundedCorners(r1 as Integer)

Values Provided	Description
0	The radii of all the box corners are automatically determined.
1	The radii of all the box corners are set to the specified value.
2	The radii of the box-top corners are set to the first value. The radii of the box-bottom corners are set to the second value.
4	The radius of the box-top corner at the negative x direction is set to the first value. The radius of the box-top corner at the positive x direction is set to the second value. The radius of the box-bottom corner at the negative x direction is set to the third value. The radius of the box-bottom corner at the positive x direction is set to the fourth value.

Argument	Default	Description
r1	-0x7ffffff	The first radius. See description above on how this radius value is used.
r2	-0x7ffffff	The second radius. See description above on how this radius value is used.
r3	-0x7ffffff	The third radius. See description above on how this radius value is used.
r4	-0x7ffffff	The fourth radius. See description above on how this radius value is used.

3.8 class CDBaseChartMBS

3.8.1 class CDBaseChartMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The BaseChart class is an abstract class containing methods that are common to all chart types.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [HiDPI for ChartDirector](#)
- [Scintilla in Xojo for Linux 64-bit on ARM](#)
- [News from the MBS Xojo Plugins Version 21.2](#)
- [Track Line with Data Labels Retina](#)
- [Register MBS Xojo Plugins](#)
- [Tip of Day: ChartTime back to date object](#)
- [YZoneColor](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [New ChartDirector Control](#)
- [MonkeyBread Software Releases the MBS REALbasic plug-ins 9.2](#)

Xojo Developer Magazine

- [7.5, pages 32 to 33: Easy Charts and Graphs Part 2, Using the ChartDirector Plugin](#)
- [7.4, page 35: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [21.6, pages 84 to 86: From 0 to 100, Creating Gauges with MBS and ChartDirector by Stefanie Juchmes](#)
- [21.6, page 82: From 0 to 100, Creating Gauges with MBS and ChartDirector by Stefanie Juchmes](#)
- [20.4, page 42: PDF Pie Charts, Adding Xojo Charts to Your PDFs by Stefanie Juchmes](#)
- [20.4, page 36: PDF Pie Charts, Adding Xojo Charts to Your PDFs by Stefanie Juchmes](#)
- [20.1, pages 45 to 46: Cool Charts and Heatmaps, Using Monkeybread Software's ChartDirector Plugin by Stefanie Juchmes](#)
- [18.5, pages 77 to 78: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)
- [18.5, pages 69 to 72: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)
- [18.5, page 67: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.8.2 Methods

3.8.3 addExtraField(numbers() as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an array of numbers/dates to be used as an extra field in various places.

Notes: This method merely stores the data inside the chart object. The Parameter Substitution and Formatting mechanism will determine how the data are to be used.

A common use for extra fields is to specify extra information (such as a custom serial number for the data points) to be displayed on data labels or on tool tips, or to supply extra query parameters in clickable charts. All these are achieved by specifying the extra field on the data label template or image map templates during parameter substitution.

Argument	Default	Description
numbers	(Mandatory)	An array of numbers/dates to be stored inside the chart object.

See also:

- 3.8.4 addExtraField(paths() as folderitem) 190
- 3.8.5 addExtraField(texts() as string) 191

3.8.4 addExtraField(paths() as folderitem)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an array of file paths to be used as an extra field in various places.

Notes: This method merely stores the data inside the chart object. The Parameter Substitution and Formatting mechanism will determine how the data are to be used.

A common use for extra fields is to specify extra information (such as a custom serial number for the data points) to be displayed on data labels or on tool tips, or to supply extra query parameters in clickable charts. All these are achieved by specifying the extra field on the data label template or image map templates during parameter substitution.

Argument	Default	Description
texts	(Mandatory)	An array of text to be stored inside the chart object.

See also:

- 3.8.3 addExtraField(numbers() as Double) 190

3.8. CLASS CDBASECHARTMBS	191
• 3.8.5 addExtraField(texts() as string)	191

3.8.5 addExtraField(texts() as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an array of text to be used as an extra field in various places.

Notes: This method merely stores the data inside the chart object. The Parameter Substitution and Formatting mechanism will determine how the data are to be used.

A common use for extra fields is to specify extra information (such as a custom serial number for the data points) to be displayed on data labels or on tool tips, or to supply extra query parameters in clickable charts. All these are achieved by specifying the extra field on the data label template or image map templates during parameter substitution.

Argument	Default	Description
texts	(Mandatory)	An array of text to be stored inside the chart object.

Great to add point labels to the chart.

See also:

• 3.8.3 addExtraField(numbers() as Double)	190
• 3.8.4 addExtraField(paths() as folderitem)	190

3.8.6 addLegend(x as Integer, y as Integer, noOfCols as Integer, font as string = "", fontsize as Double = 10) as CDLegendBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a legend box to the chart with grid layout.

Notes: In grid layout, the legend box will be divided into a table in which all cells are of of same width. The legend entries will fill the cells from left to right, top to bottom.

The number of columns can be specified using the noOfCols argument. If this argument is set to the special constant AutoGrid (= -2), the number of columns will be automatically determined based on the longest legend entry. If this argument is 0, the legend box will use a flow layout (from left to right and then top to bottom, in which the entries may not be vertically aligned).

In any case, the number of rows in the table is automatically determined so as to have enough cells for all legend entries.

Argument	Default	Description
x	(Mandatory)	The x coordinate of the reference point of the legend box. By default, the reference point is the top-left corner of the box, but can be configured by using <code>TextBox.setAlignment</code> .
y	(Mandatory)	The y coordinate of the reference point of the legend box. By default, the reference point is the top-left corner of the box, but can be configured by using <code>TextBox.setAlignment</code> .
noOfCols	(Mandatory)	The number of columns in the legend box. The special value <code>AutoGrid (= -2)</code> means the number of columns is automatically determined. If this argument is 0, the legend box will use a flow layout (from left to right and then top to bottom, in which the entries may not be vertically aligned).
font	""	The font name of the font for drawing the legend text. The default is "normal". See Font Specification for details on various font attributes.
fontSize	10	The font size of the legend text.

Return Value

A `LegendBox` object representing the legend box. You may use this object to fine-tune the appearance of the legend box.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.8.7 `addLegend(x as Integer, y as Integer, vertical as boolean=true, font as string = "", fontsize as Double = 10) as CDLegendBoxMBS` 192

3.8.7 `addLegend(x as Integer, y as Integer, vertical as boolean=true, font as string = "", fontsize as Double = 10) as CDLegendBoxMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a legend box to the chart.

Notes: The entries in the legend box will flow from top to bottom (one line per entry), or from left to right and then top to bottom (like flowing text), depending on the vertical argument. For grid layout, use `BaseChart.addLegend2`.

Return Value

A `LegendBox` object representing the legend box. You may use this object to fine-tune the appearance of the legend box.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

Argument	Default	Description
x	(Mandatory)	The x coordinate of the reference point of the legend box. By default, the reference point is the top-left corner of the box, but can be configured by using <code>TextBox.setAlignment</code> .
y	(Mandatory)	The y coordinate of the reference point of the legend box. By default, the reference point is the top-left corner of the box, but can be configured by using <code>TextBox.setAlignment</code> .
vertical	true	A true value means the legend keys are laid out vertically (one line per entry). A false value means the legend keys are laid out horizontal and flow like text (from left to right, top to bottom).
font	""	The font name of the font for drawing the legend text. The default is "normal". See <code>Font Specification</code> for details on various font attributes.
fontSize	10	The font size of the legend text.

- 3.8.6 `addLegend(x as Integer, y as Integer, noOfCols as Integer, font as string = "", fontsize as Double = 10)` as `CDLegendBoxMBS` 191

3.8.8 `addLine(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, color-value as color, lineWidth as Integer = 1)` as `CDLineMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addLine` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.8.9 `addLine(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, colorvalue as Integer = &hfff0001, lineWidth as Integer = 1)` as `CDLineMBS` 193

3.8.9 `addLine(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, color-value as Integer = &hfff0001, lineWidth as Integer = 1)` as `CDLineMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a line to the chart.

Notes:

Argument	Default	Description
x1	(Mandatory)	The x coordinate of the first endpoint of the line.
y1	(Mandatory)	The y coordinate of the first endpoint of the line.
x2	(Mandatory)	The x coordinate of the second endpoint of the line.
y2	(Mandatory)	The y coordinate of the second endpoint of the line.
color	<code>LineColor</code>	The color of the line.
lineWidth	1	The width of the line.

Return Value

A Line object representing the line added. You may use this object to fine-tune the appearance of the line. See also:

- 3.8.8 `addLine(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, colorvalue as color, lineWidth as Integer = 1) as CDLineMBS` 193

3.8.10 `addTable(x as Integer, y as Integer, alignment as Integer, col as Integer, row as Integer) as CDMLTableMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a CDML table to the chart.

Notes: Arguments:

Argument	Default	Description
x	(Mandatory)	The x-coordinate of the reference point used to position the table.
y	(Mandatory)	The y-coordinate of the reference point used to position the table.
alignment	(Mandatory)	The alignment of the table with respect to the reference point. For example, a value of <code>kTopLeft</code> means the reference point is the top-left corner of the table. See Alignment Specification for supported alignment types..
col	(Mandatory)	The number of columns in the table.
row	(Mandatory)	The number of rows in the table.

Returns a CDMLTableMBS object representing the CDML table added.

3.8.11 `addText(x as Integer, y as Integer, text as string, font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002, alignment as Integer = 7, angle as Double = 0, vertical as boolean=false) as CDTextBoxMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a text box to the chart.

Notes: By default, only the text is visible, the box is transparent. This method returns a TextBox object that can be used to change the appearance of the text box.

Return Value

A TextBox object representing the text box. This may be used to fine-tune the appearance of the text box.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

Argument	Default	Description
x	(Mandatory)	The x coordinate of the left of the text box.
y	(Mandatory)	The y coordinate of the top of the text box.
text	(Mandatory)	The text to shown in the text box. See ChartDirector Mark Up Language on how to embed special tags in the text for sophisticated formatting.
font	""	The font used to draw the text. See Font Specification for details on various font attributes.
fontSize	8	The font size used to draw the text.
fontColor	TextColor	The color used to draw the text.
alignment	TopLeft	The alignment of the text within the text box. See Alignment Specification for supported alignment types.
angle	0	The rotation angle of the text within the text box.
vertical	false	Indicate whether the text should be laid out vertically (from top to bottom) or horizontally (from left to right).

See also:

- 3.8.12 addText(x as Integer, y as Integer, text as string, font as string, fontsize as Double, fontcolor as color, alignment as Integer = 7, angle as Double = 0, vertical as boolean=false) as CDTextBoxMBS
195

3.8.12 addText(x as Integer, y as Integer, text as string, font as string, fontsize as Double, fontcolor as color, alignment as Integer = 7, angle as Double = 0, vertical as boolean=false) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addText method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.11 addText(x as Integer, y as Integer, text as string, font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002, alignment as Integer = 7, angle as Double = 0, vertical as boolean=false) as CDTextBoxMBS
194

3.8.13 addTitle(alignment as Integer, text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a title to the chart.

Notes:

Return Value

Argument	Default	Description
alignment	(Mandatory)	The position on the title on the chart. See Alignment Specification for supported alignment types.
text	(Mandatory)	The text for the title. See ChartDirector Mark Up Language on how to embed special tags in the text for sophisticated formatting.
font	""	The font used to draw the title text. The default is "bold". See Font Specification for details on various font attributes.
fontSize	12	The font size in points for the title text.
fontColor	TextColor	The color of the title text.
bgColor	Transparent	The background color of the title box.
edgeColor	Transparent	The border color of the title box.

A TextBox object representing the title box. This may be used to fine-tune the appearance of the title box.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.8.14 addTitle(alignment as Integer, text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS 196
- 3.8.15 addTitle(text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS 197
- 3.8.16 addTitle(text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS 197

3.8.14 addTitle(alignment as Integer, text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addTitle method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.13 addTitle(alignment as Integer, text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS 195
- 3.8.15 addTitle(text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS 197

- 3.8.16 addTitle(text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS 197

3.8.15 addTitle(text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a title at the top center of the chart.

Notes: The title is contained within a box, of which the width is the same as the width of the chart, and the height is automatically adjusted to fit the text. The box is initially invisible, but can be made visible by setting the bgColor and edgeColor.

Argument	Default	Description
text	(Mandatory)	The text for the title. See ChartDirector Mark Up Language on how to embed special tags in the text for sophisticated formatting.
font	""	The font used to draw the title text. The default is "bold". See Font Specification for details on various font attributes.
fontSize	12	The font size in points for the title text.
fontColor	TextColor	The color of the title text.
bgColor	Transparent	The background color of the title box.
edgeColor	Transparent	The border color of the title box.

Return Value

A TextBox object representing the title box. This may be used to fine-tune the appearance of the title box.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.8.13 addTitle(alignment as Integer, text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS 195
- 3.8.14 addTitle(alignment as Integer, text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS 196
- 3.8.16 addTitle(text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS 197

3.8.16 addTitle(text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addTitle method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.13 addTitle(alignment as Integer, text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS 195
- 3.8.14 addTitle(alignment as Integer, text as string, font as string, fontsize as Double, fontColor as color, bgColor as color, edgeColor as color) as CDTextBoxMBS 196
- 3.8.15 addTitle(text as string, font as string = "", fontsize as Double = 12, fontColor as Integer = &hfff0002, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000) as CDTextBoxMBS 197

3.8.17 adjustBrightness(ColorValue as color, brightness as Double) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other adjustBrightness method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.18 adjustBrightness(ColorValue as Integer, brightness as Double) as Integer 198

3.8.18 adjustBrightness(ColorValue as Integer, brightness as Double) as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a color that is a darkened or brightened version of the given color.

Notes: A brightness less than 1 means the color is darkened, while a brightness greater than 1 means the color is brightened. For example, a brightness of 0.5 means the color is half as bright as the original color. If the original color is red, the color will become dark red. Conversely, a brightness of 2 means the color is twice as bright as the original color. If the original color is red, the color will become light red.

Arguments:

Argument	Default	Description
c	(Mandatory)	The given color.
brightness	(Mandatory)	A non-negative number represent the factor to darken or brighten the color.

Return Value

A 32-bit integer representing the darkened or brightened color.

See also:

- 3.8.17 adjustBrightness(ColorValue as color, brightness as Double) as Integer 198

3.8.19 AllPassFilter as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches every element.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` as a "catch all" filter.

Return Value

An integer filter id representing the filter.

3.8.20 ArrowShape(angle as Double = 0.0, widthRatio as Double = 1, stemWidthRatio as Double = 0.5, stemLengthRatio as Double = 0.5) as Integer

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the shape id that represents an arrow shape.

Notes: Please refer to Shape Specification for samples and more information on using shapes in `ChartDirector`.

Argument	Default	Description
angle	0	The clockwise angle of the arrow relative to the upward pointing position.
widthRatio	1	The width of the base of the arrow head as a ratio to the height of the entire arrow. Must be between 0 and 1.
stemWidthRatio	0.5	The width of the arrow stem as a ratio to the width of the base of the arrow head. Must be between 0 and 1.
stemLengthRatio	0.5	The length of the arrow stem as a ratio to the height of the entire arrow. Must be between 0 and 1.

Returns an integer shape id representing the arrow shape.

3.8.21 barLighting(startBrightness as Double = 0.75, endBrightness as Double = 1.5) as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A special shading effect for rectangular and polygonal bars on a `CDBarLayerMBS`.

Notes: It shades all surfaces of a 2D or 3D bar with gradient colors.

For the front surface of a bar, the gradient is from base line of the bar to the end of the bar. For example, for a vertical bar pointing upwards, the gradient is from bottom to the top. Similarly, for a horizontal bar pointing rightwards, the gradient is from left to right.

For a 3D bar, the gradient for the top surface of a vertical bar or the right surface of a horizontal bar is from front to back. The gradients for the side surfaces are the same as for the front surface.

The gradient is specified with two brightness values at the gradient end points. A brightness less than 1 means the color is darkened, while a brightness greater than 1 means the color is brightened. For example, a brightness of 0.5 means the color is half as bright as the original color. If the original color is red, the color will become dark red. Conversely, a brightness of 2 means the color is twice as bright as the original color. If the original color is red, the color will become light red.

The followings are some examples demonstrating this effect.

The `barLighting` method returns an integer representing this effect. The integer can be used as the second argument to `CDLayerMBS.setBorderColor` for `CDBarLayerMBS` objects to apply the effect to bars.

Arguments:

Argument	Default	Description
<code>startBrightness</code>	0.75	The brightness at the starting point.
<code>endBrightness</code>	1.5	The brightness at the ending point.

Return Value

An integer representing the bar lighting effect.

3.8.22 `blueMetalGradient` as `Integer`()

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A constant array of integers to represent a gradient that looks like a blue metallic color.

Notes: The array is in a format that can be directly used in `BaseChart.gradientColor2` and `DrawArea.gradientColor2`. Its contents (in hex) is:

```
00 9898E0 60 F0F0FF B0 D8D8F0 100 9898E0
```

See Color Specification on how colors are represented in `ChartDirector`.

3.8.23 `brushedGoldColor(texture as Integer = 2, angle as Integer = 90)` as `Integer`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a brushed golden color, most commonly used as a background color.

Notes: This method is a short cut to the `CDBaseChartMBS.brushedMetalColor` method, using yellow (FFEE44 in hex) as the base color.

Arguments:

Argument	Default	Description
texture	2	The strength of the brushed texture. Must be 0, 1, 2 or 3 for no texture, light texture, medium texture and strong texture.
angle	90	The direction for brightness modulation, specified as a clockwise angle in degrees, with 0 being the upward pointing direction.

Return Value:

A 32-bit integer representing the brushed golden color.

3.8.24 `brushedMetalColor(c as Integer, texture as Integer = 2, angle as Integer = 90) as Integer`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a color by modulates the brightness of another color to create brushed metallic shiny effects.

Notes: The brightness of the color will vary smoothly across the image in a given a direction, so as to produce a shiny effect. The color will then be modulated with a horizontal texture to create the brushed effect. The modulation period will be the same as the size of the image, so this method is best use to create background colors.

Arguments:

Argument	Default	Description
c	(Mandatory)	The color to be modulated.
texture	2	The strength of the brushed texture. Must be 0, 1, 2 or 3 for no texture, light texture, medium texture and strong texture.
angle	90	The direction for brightness modulation, specified as a clockwise angle in degrees, with 0 being the upward pointing direction.

Return Value:

A 32-bit integer representing the brushed metallic color.

3.8.25 brushedSilverColor(texture as Integer = 2, angle as Integer = 90) as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a brushed silver color, most commonly used as a background color.

Notes: This method is a short cut to the CDBaseChartMBS.brushedMetalColor method, using grey (DDDDDD in hex) as the base color.

Arguments:

Argument	Default	Description
texture	2	The strength of the brushed texture. Must be 0, 1, 2 or 3 for no texture, light texture, medium texture and strong texture.
angle	90	The direction for brightness modulation, specified as a clockwise angle in degrees, with 0 being the upward pointing direction.

Return Value

A 32-bit integer representing the brushed silver color.

3.8.26 bSearch(values() as Double, value as Double) as Double

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses binary search to search for a value in an array.

Notes: This method returns the array index of the value in the array, which must be sorted in ascending order.

If the value is in between two elements of the array, this method returns a non-integer that interpolates the indexes of the two elements. For example, suppose the array consists of 3 elements [4, 6, 10] . If this method is used to search for the value 7, it will return 1.25.

If the value is smaller or larger than all the elements in the array, this method returns the nearest index, which must be either 0 or the index of the last element of the array.

Argument	Default	Description
a	(Mandatory)	The array to be searched.
v	(Mandatory)	The value to search for.

Returns the index of the value within the array. If the value is in between two elements of the array, this method returns a non-integer that interpolates the indexes of the two elements. If the value is smaller or

larger than all the elements in the array, this method returns the nearest index.

3.8.27 chartTime(t as Integer) as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts a UNIX time (seconds elapsed since 01-01-1970 00:00:00 GMT) to the date/time format used by ChartDirector.

Notes: The UNIX time will be converted assuming based on local time zone (the time zone settings of the operating system).

Argument	Default	Description
t	(Mandatory)	A time represented as seconds elapsed since 01-01-1970 00:00:00 GMT.

Return Value

The second elapsed since 01-01-0001 00:00:00 to the given time.

See also:

- 3.8.28 chartTime(year as Integer, month as Integer, day as Integer, hour as Integer = 0, minute as Integer = 0, second as Integer = 0) as Double 203

3.8.28 chartTime(year as Integer, month as Integer, day as Integer, hour as Integer = 0, minute as Integer = 0, second as Integer = 0) as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Obtain the second elapsed since 01-01-0001 00:00:00 to the given time, which is the date/time format used by ChartDirector.

Notes:

Argument	Default	Description
y	(Mandatory)	The year component of the given time.
m	(Mandatory)	The month component of the given time
d	(Mandatory)	The day of month component of the given time.
h	0	The hour component of the given time.
n	0	The minute component of the given time.
s	0	The second component of the given time.

Return Value

The second elapsed since 01-01-0001 00:00:00 to the given time.

See also:

- 3.8.27 chartTime(t as Integer) as Double

3.8.29 ClearTypeColor(gamma as Double = 0) as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets a value to represent that standard ClearType or a similar technology be used for font rendering.

Notes: This constant is used in BaseChart.setAntiAlias and DrawArea.setAntiAlias.

ClearType is a Microsoft font rendering technology that renders text more accurately with subpixel rendering. Similar technologies include Quartz in Mac OS X, Adobe CoolType and other unnamed algorithms. Although this API uses ClearType in its name, ChartDirector may use ClearType or a similar technology depending on the operating system and programming language.

In most modern flat panel displays, a pixel is consisted of 3 sub-pixels R, G and B for the red, green and blue colors, typically arranged horizontally. So a row of pixels is actually a sequence of subpixels like RG-BRGBRGRGB... To display a white pixel, the display hardware turns on the RGB subpixels.

It happens a white dot can also be made by turning on the GB subpixels of one pixel, and the R subpixel of the next pixel to the right. This will result in 3 consecutive subpixels GBR, which is also white. This white dot will be in between the two pixels, with 2/3 on the left pixel, and 1/3 on the right pixel. In other words, it is possible to position a white dot in between two pixels without blurring it. ClearType uses this effect to render text with subpixel accuracy.

Apart from subpixel rendering, ClearType may adjust the glyph shapes to fit the pixel grid (technically called hinting) differently from classical rendering. As a result, the glyph shapes and sizes in ClearType may be different from classical rendering.

One issue with ClearType is that it is hardware dependent. To work perfectly, it needs to know the subpixels ordering of the display. In many applications, the charts are rendered on one computer, but viewed using another computer. For example, in a web application, the charts can be rendered on the server but displayed on the browser computer. The chart generating computer may not know the subpixel configuration of the viewing computer. If ClearType is used, there is a risk that the text may look suboptimal on the viewing computer.

To address the above issues, ChartDirector supports a hardware independent ClearType method, represented by ClearTypeMono. This is basically ClearType with subpixel rendering disabled. The resulting text has no color distortion. Black and white text will not be as sharp as standard ClearType, but is still better than classical anti-alias. The charts rendered can be viewed equally well with different types of displays. This is especially useful for applications in which the charts are rendered and viewed by different computers.

Argument	Default	Description
gamma	0	Specifies the level of gamma correction used for ClearType font rendering. This is usually a number between 1 to 2.5. A value of 0 means the default gamma level.

3.8.30 ClearTypeMono(gamma as Double = 0) as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets a value to represent that hardware independent ClearType or a similar technology be used for font rendering.

Notes: This constant is used in BaseChart.setAntiAlias and DrawArea.setAntiAlias.

Please refer to ClearTypeColor on the detail explanation of what is standard ClearType and hardware independent ClearType.

Argument	Default	Description
gamma	0	Specifies the level of gamma correction used for ClearType font rendering. This is usually a number between 1 to 2.5. A value of 0 means the default gamma level.

3.8.31 ColorToInteger(c as color, alpha as Integer = 0) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts a Xojo color object to an integer for ChartDirector.

3.8.32 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.8.33 Cross2Shape(width as Double = 0.5) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the shape id that represents a 'X' shape.

Notes: Please refer to Shape Specification for samples and more information on using shapes in ChartDi-

rector.

Argument	Default	Description
width	(Mandatory)	The percentage width of the arms of the a 'X' relative to the entire shape. Must be between 0 and 1.

Return Value

An integer shape id representing the a 'X' shape.

3.8.34 CrossShape(width as Double = 0.5) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the shape id that represents a '+' shape.

Notes: Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.

Argument	Default	Description
width	(Mandatory)	The percentage width of the arms of the a '+' relative to the entire shape. Must be between 0 and 1.

Return Value

An integer shape id representing the a '+' shape.

3.8.35 cylinderEffect(orientation as Integer = 5, ambientIntensity as Double = 0.5, diffuseIntensity as Double = 0.5, specularIntensity as Double = 0.75, shininess as Integer = 8) as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A special shading effect that emulates the lighting of a cylinder surface.

Notes: This effect adjusts the brightness of the color of a rectangular box to make it look like cylindrical. The brightness is adjusted as according to the Phong lighting model, in which the light source is from the viewer direction and is far away.

The cylinderEffect method returns an integer representing this effect. The integer can be used as the third argument to CDBoxMBS.setBackground to apply the effect to Box objects (including derived objects such as CDTextBoxMBS objects).

This effect is automatically used for cylindrical bars in a CDBarLayerMBS. You may use this method to adjust the lighting parameters by using its return value as the second argument to CDLayerMBS.setBorderColor.

Arguments:

Argument	Default	Description
orientation	kCenter	The orientation of the cylinder. A value of kTop or kBottom means the cylinder is vertical. A value of kLeft or kRight means the cylinder is horizontal. A value of kCenter means the orientation is automatically determined. For a CDBarLayerMBS, the orientation will be the same as the orientation of the bars. For other objects (eg. CDTextBoxMBS objects), if the object height is bigger than its width, the cylinder will be treated as vertical, otherwise it will be treated as horizontal.
ambientIntensity	0.5	The ambient reflection coefficient of the Phong lighting model.
diffuseIntensity	0.5	The diffuse reflection coefficient of the Phong lighting model.
specularIntensity	0.75	The specular reflection coefficient of the Phong lighting model.
shininess	8	The shininess coefficient of the Phong lighting model.

Return Value

An integer representing the cylinder effect.

3.8.36 dashLineColor(colorvalue as color, patternCode as Integer = &h0505) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other dashLineColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.37 dashLineColor(colorvalue as Integer, patternCode as Integer = &h0505) as Integer 207

3.8.37 dashLineColor(colorvalue as Integer, patternCode as Integer = &h0505) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A constant equals to 0505 (in hex) to represent a dash line pattern for use in dash colors.

Notes: See Color Specification on how colors are represented in ChartDirector.

See also:

- 3.8.36 dashLineColor(colorvalue as color, patternCode as Integer = &h0505) as Integer 207

3.8.38 defaultPalette as Integer()

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the default palette.

3.8.39 Destructor

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The destructor for the base chart.

3.8.40 enableVectorOutput

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Enables true vector graphics output.

Notes: By default, when creating the output image, ChartDirector draws directly onto an output buffer representing the bitmap of the image. For example, for a chart 800 x 600 pixels in size, the output buffer may represent a 800 x 600 bitmap. The output buffer size is unchanged no matter how many items are drawn onto it. Even if the output contains 1 million elements (eg. 1 million symbols), the size of the output buffer is still the same.

On the other hand, a true vector output is indefinitely scalable and can be considered as having infinite resolution. To produce a true vector output, it is necessary to remember the graphics operations for every element in the output buffer. The output buffer size is therefore proportional to the number of elements to draw.

This method tells ChartDirector that it needs to remember the graphics operations to prepare for true vector output. If true vector output is needed, this method should be called immediately after creating the BaseChart object.

If this method is not called, and a vector graphics output format is used (such as SVG), instead of a true vector output, ChartDirector will output a raster image using the vector graphics format. (Most vector graphics formats support embedded raster images.)

3.8.41 flatBorder(thickness as Integer) as Integer

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Specifies a flat border of a given width.

Example:

```
// set edge to be 2 pixel wide line
Dim table As CDMLTableMBS
Dim t As CDTextBoxMBS = table.getStyle
Dim x As Integer = c.flatBorder(2)
t.setBackground(&heeeeee, CDXYChartMBS.kLineColor, x)
```

Notes: The flatBorder method returns an integer representing this effect. The integer can be used as the third argument to Box.setBackground to apply the effect to Box objects (including derived objects such as TextBox objects). It may also be used as the second argument to Layer.setBorderColor for BarLayer or BoxWhiskerLayer objects to apply the effect to bars or boxes.

Argument	Default	Description
width	(Mandatory)	The border width. A positive value means the border is drawn inside the box or bar. A negative value means the border is outside the box or bar.

Returns an integer representing using flat border of the specified width.

3.8.42 formatValue(value as Double, formatstring as string) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Formats a number/date using the ChartDirector formatting syntax as is in Parameter Substitution and Formatting.

Notes:

Argument	Default	Description
value	(Mandatory)	The value to be formatted.
formatString	(Mandatory)	The format string, using { value } to denote the value. For example, \$ { value 2, } can be used to format the value to 2 decimal points, using "," as the thousand separator, and with a '\$ ' sign in front.

3.8.43 getAbsOffsetX as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x offset of the chart relative to the outermost MultiChart container.

Notes: Returns an integer representing the x offset in pixels relative to the outermost MultiChart container, or 0 if the chart is not within a MultiChart container.

3.8.44 getAbsOffsetY as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y offset of the chart relative to the outermost MultiChart container.

Notes: Returns an integer representing the y offset in pixels relative to the outermost MultiChart container, or 0 if the chart is not within a MultiChart container.

3.8.45 getChartMetrics as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the chart metrics for passing to CChartViewer to support view ports.

Notes: The format of the chart metrics is not published.

Return Value

A text string representing the chart metrics.

3.8.46 getChartWeekDay(t as Double) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the weekday represented by a ChartDirector date/time.

Notes:

Argument	Default	Description
t	(Mandatory)	A ChartDirector date/time, that is, a number representing seconds elapsed since 01-01-0001 00:00:00.

Return Value

An integer from 0 - 6 representing Sun - Sat.

3.8.47 getChartYMD(t as Double) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the year, month and day represented by a ChartDirector date/time.

Notes:

Argument	Default	Description
t	(Mandatory)	A ChartDirector date/time, that is, a number representing seconds elapsed since 01-01-0001 00:00:00.

Return Value

An integer which when represented in decimal notation is yyyyymmdd, where yyyy is the year, mm is the month (1 - 12), and dd is the day (1 - 31).

3.8.48 getColor(index as Integer) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the color at the specified position of the palette.

Notes: See Color Specification on how colors are represented in ChartDirector.

Argument	Default	Description
paletteEntry	(Mandatory)	An index to the palette.

Return Value

The requested color.

3.8.49 getCopyright as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the copyright string of the ChartDirector library.

Example:

```
MsgBox CDBaseChartMBS.getCopyright
// displays "Copyright (c) 2006 Advanced Software Engineering Limited"
```

3.8.50 getDescription as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the library description.

Example:

```
MsgBox CDBaseChartMBS.getDescription // displays "ChartDirector"
```

3.8.51 getDrawArea as CDDrawAreaMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the internal DrawArea object that is used to draw the chart.

Notes: The most common reason of accessing the internal DrawArea object is to add custom drawings (lines, texts, shapes, etc) to the chart.

If the custom drawings are drawn before drawing the chart image (eg. using BaseChart.makeChart, BaseChart.makeChart or BaseChart.makeChart3), the custom drawings will be at the background of the chart.

Return Value

A DrawArea object that can be used to add custom text and shapes to the chart.

3.8.52 getHeight as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the height of the chart.

3.8.53 getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates an HTML image map for the chart.

Notes: This method generates an image map to represent all data points on the chart. It does not include legend box, title box or custom text box. Please use LegendBox.getHTMLImageMap or Box.getImageCoor to produce image maps for these objects.

This method should be called only after creating the chart image (eg. using BaseChart.makeChart, BaseChart.makeChart or BaseChart.makeChart3). The image map cannot be determined without creating the chart image first.

This method accepts a URL as its argument. When generating an image map, it appends query parameters to the URL to indicate which data point the user has clicked.

The following is an example image map generated for a bar chart with 3 bars.

```
<area shape="rect" coords="34,219,63,139" href="myurl.cpp?x=0&xLabel=Mon&dataSet=0&dataSetName=Revenue&value=100">
<area shape="rect" coords="74,219,103,119" href="myurl.cpp?x=1&xLabel=Tue&dataSet=0&dataSetName=Revenue&value=125">
<area shape="rect" coords="114,219,143,22" href="myurl.cpp?x=2&xLabel=Wed&dataSet=0&dataSetName=Rev-
```

```
enue&value=245.78">
```

The image map consists of multiple `<area>` tags, one for each bar in the chart. In the "href" attributes, query parameters are appended to the URL to provide information on the bar clicked.

The image map produces by ChartDirector does not include the `<map>` and `</map>` tag. This is intentional so that you can add additional custom `<area>` tags to the image map, or append multiple image maps together.

The type of query parameters to append to the URL depends on the chart type and layer type. The default query parameters are as follows.

Chart/Layer Type Default Query Format

Pie chart	sector= { sector } &label= { label } &value= { value } &percent= { percent }
Bar, Line, Spline, Step Line, Area and Scatter layers	x= { x } &xLabel= { xLabel } &dataSet= { dataSet } &dataSetName= { dataSetName } &value= { value }
Percentage Bar and Percentage Area layers	x= { x } &xLabel= { xLabel } &dataSet= { dataSet } &dataSetName= { dataSetName } &value= { value } &percent= { percent }
HLOC and CandleStick layers	x= { x } &xLabel= { xLabel } &high= { high } &low= { low } &open= { open } &close= { close }
Box-Whisker layer	x= { x } &xLabel= { xLabel } &top= { top } &bottom= { bottom } &max= { max } &min= { min } &med= { med }
Trend layer	dataSetName= { dataSetName }
Vector layer and Polar Vector layer	x= { x } &xLabel= { xLabel } &dataSetName= { dataSetName } &value= { value } &dir= { dir } &len= { len }
Polar Line, Area, Spline Line and Spline Area layers	"x= { x } &label= { label } &name= { name } &value= { value } "

The texts in curly brackets (e.g. { sector } , { dataSet } , etc.) will be replaced by the actual values when generating the image map. For example, { sector } will be replaced by the sector number of the sector.

ChartDirector allows developers to modify the query parameters by using the queryFormat argument. For example, if "x= { x } &v= { value } " is used as the queryFormat for a XYChart, only the x position and the value of the data point will be included in query parameters.

Please refer to Parameter Substitution and Formatting on all available parameters and their meanings.

In addition to customizing the query parameters, ChartDirector supports additional HTML attributes in the `<area>` tags by using the extraAttr argument.

For example, the following extraAttr will add an "title" HTML attribute to every `<area>` tag. The attribute which will contain the x-axis label and the value of the data point. The "title" attribute will be displayed as "tool tip" when the mouse moves over the image map.

title=' { xlabel } : { value } '

Another common usage of the extraAttr argument is to add "onmouseover" and "onmouseout" HTML attributes to handle user interaction using Javascript on the browser.

Argument	Default	Description
url	(Mandatory)	The URL to be used in the "href" attribute of the image map. Parameter Substitution and Formatting is supported. Use an empty string if no href attribute is needed.
queryFormat	""	A text string representing the template of the query parameters to be appended to the URL. Parameter Substitution and Formatting is supported.

The special keyword " { default } " represents the default query parameters. This is useful for specifying appending to the default.

Note that an empty string means to use the default query query parameters. To specify no query parameter, use a space character.

extraAttr	""	A text string to specify additional attributes to add to the <area>tag. Parameter Substitution and Formatting is supported.
offsetX	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
offsetY	0	An offset to be added to all y coordinates in the image map. See offsetX above for description.

Return Value

A text string containing the image map generated.

3.8.54 getLegend as CDLegendBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the LegendBox object representing the legend box in the chart.

Notes: The LegendBox obtains using this method is not fixed. This allows you to add more keys to the legend box, and change the legend box fonts, etc. However, this also means that the legend box size (Box.getWidth and Box.getHeight) is undefined.

To obtain the width and height of the legend box, use BaseChart.layoutLegend.

Return Value

The LegendBox object representing the legend box in the chart.

3.8.55 GetPath(path as folderitem) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets a path string for a folderitem.

Notes: Returns the path in the UTF8 format as ChartDirector needs them.

3.8.56 getVersion as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets ChartDirector version information.

Example:

```
dim v as Integer = CDBaseChartMBS.getVersion
```

```
dim v1 as Integer = Bitwise.BitAnd( Bitwise.ShiftRight(v, 24), 255)
```

```
dim v2 as Integer = Bitwise.BitAnd( Bitwise.ShiftRight(v, 16), 255)
```

```
dim v3 as Integer = Bitwise.BitAnd( v, 65535)
```

```
MsgBox str(v1)+"."+str(v2)+"."+str(v3)
```

Notes: The version number is encoded as a 32-bit integer. The most significant 8 bits is the major version number. The next 8 bits are the minor version number. The least significant 16 bits are the build number.

Return Value

The version information encoded in a 32-bit integer.

3.8.57 getWidth as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the width of the chart.

3.8.58 glassEffect(glareSize as Integer = 3, glareDirection as Integer = 8, raisedEffect as Integer = 5) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A complex shading effect that emulates tinted glass or semi-transparent plastic material.

Notes: This effect involves glare and variation of lighting caused by reflection and refraction inside the material. It is best explained by viewing the examples.

ExampleLocation of lighting effect

Error Line Chart	The title is shaded using glass effect, with NormalGlare, lighting from Top, and raised effect of 5 pixels.
Donut Chart	Both the title and the sector labels are shaded using glass effect with ReducedGlare, lighting from Top, and raised effect of 5 pixels.
Glass Multi-Bar Chart	The bars are shaded using glass effect, with NormalGlare, lighting from Left, and raised effect of 5 pixels.

Currently, this effect only works well for long, thin objects, such as title bars, text boxes and bars in bar charts.

The `glassEffect` method returns an integer representing this effect. The integer can be used as the third argument to `Box.setBackground` to apply the effect to objects derived from `Box` (such as labels and titles represented by `TextBox`). It may also be used as the second argument to `Layer.setBorderColor` for `BarLayer` objects to apply the effect to bars.

Argument	Default	Description
<code>glareSize</code>	<code>NormalGlare</code>	The amount of glare. Must be one of the predefined constants <code>NormalGlare</code> , <code>ReducedGlare</code> or <code>NoGlare</code> . With <code>NormalGlare</code> and <code>ReducedGlare</code> , the glare will cover around 50% and 35% of the object. If <code>NoGlare</code> is used, there will be no glare and the effect will not look like glass at all, but is equivalent to the <code>CDBaseChartMBS.softLighting</code> effect.

Usually, `NormalGlare` is best for thin objects with lightly colored background, while `ReducedGlare` is best for dark background objects or not-so-thin objects (eg. text boxes using white text on a dark background, or with more than 1 line of text).

<code>glareDirection</code>	<code>Top</code>	The direction of the glare, which must be one of the predefined constants <code>Top</code> , <code>Right</code> , <code>Bottom</code> , <code>Left</code> .
<code>raisedEffect</code>	<code>5</code>	With glass shading effect, the object will appear to have some 3D depth. The <code>raisedEffect</code> argument controls the amount of 3D depth in pixels.

Return Value

An integer representing the glass effect.

3.8.59 `goldColor(angle as Integer = 90) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a golden color, most commonly used as a background color.

Notes: This method is a short cut to the `CDBaseChartMBS.metalColor` method, using yellow (FFEE44 in

hex) as the base color.

Argument	Default	Description
angle	90	The direction for brightness modulation, specified as a clockwise angle in degrees, with 0 being the upward pointing direction.

Return Value

A 32-bit integer representing the golden color.

3.8.60 goldGradient as Integer()

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A constant array of integers to represent a gradient that looks like a golden color.

Notes: The array is in a format that can be directly used in BaseChart.gradientColor and DrawArea.gradientColor. Its contents (in hex) is:

```
00 FFE743 60 FFFFE0 B0 FFF0B0 100 FFE743
```

See Color Specification on how colors are represented in ChartDirector.

3.8.61 gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other gradientColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.62 gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer 217
- 3.8.63 gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer 218
- 3.8.64 gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer 219

3.8.62 gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a multi-point linear gradient color.

Notes: This method is for backward compatibility. Use `BaseChart.linearGradientColor2` instead.

Argument	Default	Description
colorArray	(Mandatory)	An array defining the positions and colors of the pixels along the reference gradient line segment.
angle	90	The direction of the reference gradient line segment in degrees, measured clockwise, with 0 degree as the upward pointing direction. The default direction is horizontal from left to right (90 degrees).
scale	1.0	The scaling factor for the reference gradient line segment. By default, the reference gradient line segment is 256 pixels in length. The scaling factor can be use to stretch or compress the gradient line segment.
startX	0	The x coordinate of the starting point of the reference gradient line segment.
startY	0	The y coordinate of the starting point of the reference gradient line segment.

Return Value

A 32-bit integer representing the linear gradient color.

See also:

- 3.8.61 `gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 217
- 3.8.63 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer` 218
- 3.8.64 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer` 219

3.8.63 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `gradientColor` method, but uses `color` instead of `integer` data type for passing color values.

See also:

- 3.8.61 `gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 217
- 3.8.62 `gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 217
- 3.8.64 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer` 219

3.8.64 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a two-point linear gradient color.

Notes: This method is for backward compatibility. Use `BaseChart.linearGradientColor` instead.

Argument	Default	Description
<code>startX</code>	(Mandatory)	The x coordinate of the starting point of the reference gradient line segment.
<code>startY</code>	(Mandatory)	The y coordinate of the starting point of the reference gradient line segment.
<code>endX</code>	(Mandatory)	The x coordinate of the ending point of the reference gradient line segment.
<code>endY</code>	(Mandatory)	The y coordinate of the ending point of the reference gradient line segment.
<code>startColor</code>	(Mandatory)	The color at the starting point of the reference gradient line segment.
<code>endColor</code>	(Mandatory)	The color at the ending point of the reference gradient line segment.

Return Value

A 32-bit integer representing the linear gradient color.

See also:

- 3.8.61 `gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 217
- 3.8.62 `gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 217
- 3.8.63 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer` 218

3.8.65 `greenMetalGradient as Integer()`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A constant array of integers to represent a gradient that looks like a green metallic color.

Notes: The array is in a format that can be directly used in `BaseChart.gradientColor2` and `DrawArea.gradientColor2`. Its contents (in hex) is:

```
00 98E098 60 F0FFF0 B0 D8F0D8 100 98E098
```

See Color Specification on how colors are represented in `ChartDirector`.

3.8.66 `halfColor(c as Integer) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a color that is half the intensity of the given color.

Notes:

Argument	Default	Description
c	(Mandatory)	The given color.

Return Value

A 32-bit integer representing the half intensity color.

3.8.67 `initWithDynamicLayer` as `CDDrawAreaMBS`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Initializes a dynamic layer for drawing text and shapes.

Notes: This method clears the existing dynamic layer, or creates a new one if there is no existing dynamic layer. This method returned a `DrawArea` object that can be used to draw on the dynamic layer. The dynamic layer can later be removed using `BaseChart.removeDynamicLayer`.

The design of the dynamic layer is for drawing small, rapidly updatable contents for desktop applications. For example, the dynamic layer can be used to implement a cross-hair mouse cursor, with text showing the location of the mouse cursor. To do this, in the mouse move event handler, `BaseChart.initDynamicLayer` can be used to create or clear the dynamic layer. The returned `DrawArea` object can then be used to draw the cross hair cursor (as two straight lines) and the text. When the mouse cursor leaves the chart, `BaseChart.removeDynamicLayer` can be used in the mouse out event handler to remove the cross-hair cursor and the text.

Note that as long as the dynamic layer is not removed with `BaseChart.removeDynamicLayer`, only the returned `DrawArea` object should be used to draw things on the dynamic layer. No other objects should be used to draw on the chart.

Returns a `DrawArea` object that can be used to add text and shapes to the dynamic layer.

3.8.68 `kDataBound as Double`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: One of the special constants.

Notes: A constant equals to $-1.7E-100$ to represent that the interpolated z values should not exceed the minimum or maximum values of the original data.

This constant is used in `ContourLayer.setZBounds`.

3.8.69 kLinearTick as Double

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: One of the special constants.

Notes: A constant equals to $+1.5E+308$ to represent that the ticks in a log scale axis should be 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 style increments.

3.8.70 kLogTick as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: One of the special constants.

Notes: A constant equals to $+1.6e308$ to represent that the ticks in a log scale axis should be 1 - 2 - 5 - 10 style increments.

This constant is used in `Axis.setLogScale`.

3.8.71 kMicroTickOnly as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: One of the special constants.

Notes: A constant equals to $-1.6e308$ to represent that the label position should contain a micro tick only. This constant is used in `Axis.setLabels` and `BaseMeter.setScale`.

3.8.72 kMinorTickOnly as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: One of the special constants.

Notes: A constant equals to $-1.7e308$ to represent that the label position should contain a minor tick only. This constant is used in `Axis.setLabels`.

3.8.73 kNoValue as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: One of the special constants.

Notes: A constant equals to $1.7E+308$ to represent missing values in `ChartDirector`. Please refer to No Value Specification for more details.

3.8.74 kTickInc as Double

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: One of the special constants.

Notes: A constant equals to +1E+200 to represent the distance between major ticks. This constant is used in CDAxisMBS.setLogScale.

3.8.75 kTouchBar as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: One of the special constants.

Notes: A constant equals to -1.7e-100 to represent that the bars in a bar layer should touch each others with no gap in between.

This constant is used in BarLayer.setBarGap.

3.8.76 layout

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Perform auto-scaling of the axis and compute the positions of the various objects in the chart.

Notes: BaseChart.layout is automatically called when drawing the chart image (eg. using BaseChart.makeChart, BaseChart.makeChart or BaseChart.makeChart3). There is usually no need to call BaseChart.layout explicitly.

However, if you would like to add custom objects to the chart whose positions depend on the axis scales or position of other objects, you may need to call BaseChart.layout explicit to auto-scaling the axis. An example is to draw a custom label at the maximum value point of a data line.

3.8.77 layoutLegend as CDLegendBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Lays out and gets the LegendBox object representing the legend box in the chart.

Notes: Once the legend box has been laid out, the width and height is known and can be obtained using Box.getWidth and Box.getHeight. However, you may not perform actions that may affect the legend box sizes, such as changing the fonts or adding more data. You may still move the legend box around by using Box.setPos.

The most common use of this method is to position the legend box based on its actual size.

Return Value

The LegendBox object representing the legend box in the chart.

3.8.78 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other linearGradientColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.79 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer 223
- 3.8.80 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer 224
- 3.8.81 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer 225

3.8.79 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a multi-point linear gradient color.

Notes: In this method, the color points are defined as an array of positions and colors along a reference line segment, in the following format:

position0, color0, position1, color1, positionN, colorN

The positions are specified as a number from 0 - 256 (0 - 100 in hex), in which 0 represents the starting point of the reference line segment, and 256 (100 in hex) represents the ending point of the reference line segment.

For example, the array (in hex):

000000, FF0000, 000080, FFFF00, 000100, 00FF00

means the starting point (000000) is red (FF0000), the mid-point (000080 in hex) is yellow (FFFF00), and the ending point (000100 in hex) is green (00FF00).

One common usage of multi-point gradient colors is to define colors that have metallic look and feel. Chart-Director comes from several predefined gradient color arrays as follows.

NameValue (in Hex)

goldGradient	000000, FFE743, 000060, FFFF00, 0000B0, FFF0B0, 000100, FFE743
silverGradient	000000, C8C8C8, 000060, F8F8F8, 0000B0, E0E0E0, 000100, C8C8C8
redMetalGradient	000000, E09898, 000060, FFF0F0, 0000B0, F0D8D8, 000100, E09898
greenMetalGradient	000000, 98E098, 000060, F0FFF0, 0000B0, D8F0D8, 000100, 98E098
blueMetalGradient	000000, 9898E0, 000060, F0F0FF, 0000B0, D8D8F0, 000100, 9898E0

Argument	Default	Description
startX	(Mandatory)	The x coordinate of the starting point of the reference gradient line segment.
startY	(Mandatory)	The y coordinate of the starting point of the reference gradient line segment.
endX	(Mandatory)	The x coordinate of the ending point of the reference gradient line segment.
endY	(Mandatory)	The y coordinate of the ending point of the reference gradient line segment.
colorArray	(Mandatory)	An array defining the positions and colors of the pixels along the reference gradient line segment.
periodic	false	Specifies whether the gradient will repeat itself periodically. If the gradient does not repeat itself, the points that lie beyond the end points of the gradient line segment will assume the colors of the end points.

Return Value

A 32-bit integer representing the linear gradient color.

See also:

- 3.8.78 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer` 223
- 3.8.80 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer` 224
- 3.8.81 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer` 225

3.8.80 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `linearGradientColor` method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.78 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer` 223

- 3.8.79 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer 223
- 3.8.81 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer 225

3.8.81 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a two-point linear gradient color.

Notes:

Argument	Default	Description
startX	(Mandatory)	The x coordinate of the starting point of the reference gradient line segment.
startY	(Mandatory)	The y coordinate of the starting point of the reference gradient line segment.
endX	(Mandatory)	The x coordinate of the ending point of the reference gradient line segment.
endY	(Mandatory)	The y coordinate of the ending point of the reference gradient line segment.
startColor	(Mandatory)	The color at the starting point of the reference gradient line segment.
endColor	(Mandatory)	The color at the ending point of the reference gradient line segment.
periodic	false	Specifies whether the gradient will repeat itself periodically. If the gradient does not repeat itself, the points that lie beyond the end points of the gradient line segment will assume the colors of the end points.

Return Value

A 32-bit integer representing the linear gradient color.

See also:

- 3.8.78 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer 223
- 3.8.79 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer 223
- 3.8.80 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer 224

3.8.82 makeChart as CDDrawAreaMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates the chart in internal format and return a DrawArea object to allow adding custom drawings on top of the chart.

Notes: If you want to add custom drawings at the background of the chart, use the BaseChart.getDrawArea

method to obtain the DrawArea instead.

After finish adding custom drawings, the resulting chart can then be output using other chart output methods.

Return Value

A DrawArea object that can be used to add custom text and shapes to the chart.

See also:

- 3.8.83 `makeChart(format as Integer) as string` 226
- 3.8.84 `makeChart(path as folderitem) as boolean` 227

3.8.83 `makeChart(format as Integer) as string`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates the chart as an image in memory.

Notes: This method is most often used to output the chart directly to an HTTP stream.

ChartDirector supports PNG, JPG, GIF, WBMP and BMP formats, denoted by the following predefined constants:

ConstantValueDescription

kPNG	0	The PNG format.
kGIF	1	The GIF format.
kJPG	2	The JPEG format.
kWMP	3	The WAP bitmap format.
kBMP	4	The BMP format.
kSVG	5	Normal SVG.
kSVGZ	6	Compressed SVG
kPDF	7	PDF format

For vector output (SVG), please call `enableVectorOutput` early.

Argument	Default	Description
format	(Mandatory)	A constant representing the format of the image.

Return Value

A memory block containing the binary image of the chart in the requested format.

See also:

3.8. CLASS CDBASECHARTMBS	227
• 3.8.82 makeChart as CDDrawAreaMBS	225
• 3.8.84 makeChart(path as folderitem) as boolean	227

3.8.84 makeChart(path as folderitem) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates the chart image and save it into a file.

Notes: ChartDirector supports PNG, JPG, GIF, WBMP, SVG, PDF and BMP. The format used are selected based on file extension, which should be png, jpg, jpeg, gif, wbmp, wmp, pdf, svg or bmp.

Argument	Default	Description
filename	(Mandatory)	The name of the file to save the image.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

See also:

• 3.8.82 makeChart as CDDrawAreaMBS	225
• 3.8.83 makeChart(format as Integer) as string	226

3.8.85 makeChartPicture as picture

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates the chart as a picture.

Notes: Returns nil on any error.

3.8.86 metalColor(c as Integer, angle as Integer = 90) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a color by modulates the brightness of another color to create metallic shiny effects.

Notes: The brightness of the color will vary smoothly across the image in a given a direction, so as to produce a shiny effect. The modulation period will be the same as the size of the image, so this method is best use to create background colors.

Return Value

A 32-bit integer representing the metallic color.

Argument	Default	Description
c	(Mandatory)	The color to be modulated.
angle	90	The direction for brightness modulation, specified as a clockwise angle in degrees, with 0 being the upward pointing direction.

3.8.87 NonePassFilter as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches no element.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` as a "deny all" filter.

Return Value

An integer filter id representing the filter.

3.8.88 patternColor(colorvalues() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `patternColor` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.8.89 `patternColor(colorvalues() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer` 228
- 3.8.90 `patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer` 229
- 3.8.91 `patternColor(path as string, startX as integer = 0, startY as integer = 0) as integer` 230
- 3.8.92 `patternColor(pic as picture, startX as Integer = 0, startY as Integer = 0) as Integer` 230

3.8.89 patternColor(colorvalues() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a pattern color using an array of colors as the bitmap pattern.

Notes: A pattern color is a dynamic color that changes according to a 2D periodic pattern. When it is used to fill an area, the area will look like being tiled with a wallpaper pattern.

Return Value

A 32-bit integer representing the pattern color.

See also:

Argument	Default	Description
colorArray	(Mandatory)	An array of colors representing the colors of the bitmap pixels. The color of the pixel at (x, y) should correspond to index (x + y * width - 1) of the array.
height	(Mandatory)	The height of the bitmap in pixels. (The width is automatically computed as the size of the color array divided by the height.)
startX	0	The x coordinate of a reference point to align with the top-left corner the pattern.
startY	0	The y coordinate of a reference point to align with the top-left corner the pattern.

- 3.8.88 patternColor(colorvalues() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer 228
- 3.8.90 patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer 229
- 3.8.91 patternColor(path as string, startX as integer = 0, startY as integer = 0) as integer 230
- 3.8.92 patternColor(pic as picture, startX as Integer = 0, startY as Integer = 0) as Integer 230

3.8.90 patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a pattern color by loading the pattern from an image file.

Notes: A pattern color is a dynamic color that changes according to a 2D periodic pattern. When it is used to fill an area, the area will look like being tiled with a wallpaper pattern.

ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to BaseChart.setSearchPath on the directory that ChartDirector will search for the file.

Argument	Default	Description
filename	(Mandatory)	An image file providing the pattern.
startX	0	The x coordinate of a reference point to align with the top-left corner the pattern.
startY	0	The y coordinate of a reference point to align with the top-left corner the pattern.

Return Value

A 32-bit integer representing the pattern color.

See also:

- 3.8.88 `patternColor(colorvalues() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer` 228
- 3.8.89 `patternColor(colorvalues() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer` 228
- 3.8.91 `patternColor(path as string, startX as integer = 0, startY as integer = 0) as integer` 230
- 3.8.92 `patternColor(pic as picture, startX as Integer = 0, startY as Integer = 0) as Integer` 230

3.8.91 `patternColor(path as string, startX as integer = 0, startY as integer = 0) as integer`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates pattern color with named item.

See also:

- 3.8.88 `patternColor(colorvalues() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer` 228
- 3.8.89 `patternColor(colorvalues() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer` 228
- 3.8.90 `patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer` 229
- 3.8.92 `patternColor(pic as picture, startX as Integer = 0, startY as Integer = 0) as Integer` 230

3.8.92 `patternColor(pic as picture, startX as Integer = 0, startY as Integer = 0) as Integer`

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a pattern color using a picture.

Notes: A pattern color is a dynamic color that changes according to a 2D periodic pattern. When it is used to fill an area, the area will look like being tiled with a wallpaper pattern.

Argument	Default	Description
<code>pic</code>	(Mandatory)	A picture. The color of the pixel at (x, y) should correspond to index (x + y * width - 1) of the array.
<code>startX</code>	0	The x coordinate of a reference point to align with the top-left corner the pattern.
<code>startY</code>	0	The y coordinate of a reference point to align with the top-left corner the pattern.

Return Value

A 32-bit integer representing the pattern color.

See also:

- 3.8. *CLASS CDBASECHARTMBS* 231
- 3.8.88 `patternColor(colorvalues() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer` 228
 - 3.8.89 `patternColor(colorvalues() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer` 228
 - 3.8.90 `patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer` 229
 - 3.8.91 `patternColor(path as string, startX as integer = 0, startY as integer = 0) as integer` 230

3.8.93 `phongLighting(ambientIntensity as Double = 0.5, diffuseIntensity as Double = 0.5, specularIntensity as Double = 0.75, shininess as Integer = 8) as Integer`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the parameters for the phong lighting effect.

Notes: The return value of this method can be used as the second argument to `Layer.setBorderColor` to configure phong lighting effect for the layer.

Argument	Default	Description
<code>ambientIntensity</code>	0.5	The ambient reflection coefficient of the Phong lighting model.
<code>diffuseIntensity</code>	0.5	The diffuse reflection coefficient of the Phong lighting model.
<code>specularIntensity</code>	0.75	The specular reflection coefficient of the Phong lighting model.
<code>shininess</code>	8	The shininess coefficient of the Phong lighting model.

3.8.94 `Polygon2Shape(slide as Integer) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the shape id that represents a polygon in an alternative orientation.

Notes: Please refer to Shape Specification for samples and more information on using shapes in `ChartDirector`.

Argument	Default	Description
<code>side</code>	(Mandatory)	The number of sides the polygon has.

Return Value

An integer shape id representing the polygon in an alternative orientation.

3.8.95 PolygonShape(slide as Integer) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the shape id that represents a polygon.

Notes: Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.

Argument	Default	Description
side	(Mandatory)	The number of sides the polygon has.

Return Value

An integer shape id representing the polygon.

3.8.96 PolynomialRegression(n as Integer) as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the degree of the polynomial regression to be used in a trend layer.

Notes: This method is used to specify the polynomial regression type in CDTrendLayerMBS.setRegressionType.

Arguments:

Argument	Default	Description
n	(Mandatory)	The degree of the polynomial.

Return Value

An integer representing a polynomial regression of degree n to be used as an argument to CDTrendLayerMBS.setRegressionType.

3.8.97 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, data() as Integer, periodic as boolean=false) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a multi-point radial gradient color.

Notes: In this method, the color points are defined as an array of radial distances and colors, in the following format:

distance0, color0, distance1, color1, distanceN, colorN

The distances are specified as a number from 0 - 256 (0 - 100 in hex), in which 0 represents the center of the gradient defining ellipse, and 256 (100 in hex) represents the perimeter of the gradient defining ellipse.

For example, the array (in hex):

000000, FF0000, 000080, FFFF00, 000100, 00FF00

means the center (000000) is red (FF0000), the mid-point (000080 in hex) is yellow (FFFF00), and the perimeter (000100 in hex) is green (00FF00).

Argument	Default	Description
cx	(Mandatory)	The x coordinate of the center of the radial gradient.
cy	(Mandatory)	The y coordinate of the center of the radial gradient.
rx	(Mandatory)	The horizontal radius of the radial gradient defining ellipse.
ry	(Mandatory)	The vertical radius of the radial gradient defining ellipse.
colorArray	(Mandatory)	An array defining the radial distances and colors.
periodic	false	Specifies whether the gradient will repeat itself periodically. If the gradient does not repeat itself, the points that lie outside the gradient defining ellipse will assume the color at the perimeter of the gradient defining ellipse.

Return Value

A 32-bit integer representing the radial gradient color.

See also:

- 3.8.98 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer 233
- 3.8.99 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer 234

3.8.98 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other radialGradientColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.97 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, data() as Integer, periodic as boolean=false) as Integer 232
- 3.8.99 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer 234

3.8.99 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a two-point radial gradient color.

Notes:

Argument	Default	Description
cx	(Mandatory)	The x coordinate of the center of the radial gradient.
cy	(Mandatory)	The y coordinate of the center of the radial gradient.
rx	(Mandatory)	The horizontal radius of the radial gradient defining ellipse.
ry	(Mandatory)	The vertical radius of the radial gradient defining ellipse.
startColor	(Mandatory)	The color at the center of the gradient defining ellipse.
endColor	(Mandatory)	The color at the perimeter of the gradient defining ellipse.
periodic	false	Specifies whether the gradient will repeat itself periodically. If the gradient does not repeat itself, the points that lie outside the gradient defining ellipse will assume the color at the perimeter of the gradient defining ellipse.

Return Value

A 32-bit integer representing the radial gradient color.

See also:

- 3.8.97 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, data() as Integer, periodic as boolean=false) as Integer 232
- 3.8.98 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer 233

3.8.100 redMetalGradient as Integer()

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A constant array of integers to represent a gradient that looks like a red metallic color.

Notes: The array is in a format that can be directly used in BaseChart.gradientColor and DrawArea.gradientColor. Its contents (in hex) is:

```
00 E09898 60 FFF0F0 B0 F0D8D8 100 E09898
```

See Color Specification on how colors are represented in ChartDirector.

3.8.101 RegularSpacingFilter(labelStep as Integer = 1, initialMargin as Integer = 0) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches 1 out of every N elements.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` to select specific elements for formatting as axis labels.

Argument	Default	Description
labelStep	1	Picks 1 out of every "labelStep" number of elements. For example, if this argument is 3, only 1 of every 3 elements will be selected.
initialMargin	0	Adds an offset when determining the elements to be selected. For example, if the labelStep is 3, the selected indexes should be 0, 3, 6, 9, If initialMargin is set to 1, the indexes becomes 1, 4, 7, 10,

Return Value

An integer filter id representing the filter.

3.8.102 removeDynamicLayer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Removes the dynamic layer if any.

3.8.103 RGB(r as Integer, g as Integer, b as Integer, a as Integer = 0) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a RGB color.

3.8.104 SelectItemFilter(item as Integer) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches the specified item.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` to select a specific element for special formatting.

Argument	Default	Description
item	(Mandatory)	The index of the specified item.

Return Value

An integer filter id representing the filter.

3.8.105 setAMPM(am as string, pm as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the names to be used to denote morning and afternoon.

Notes: The default is to use "am" and "pm".

Argument	Default	Description
am	(Mandatory)	The name used to denote morning.
pm	(Mandatory)	The name used to denote afternoon.

3.8.106 setAntiAlias(shapeAntiAlias as Boolean, textAntiAlias as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Controls whether anti-alias is used when drawing lines, shapes and text.

Notes: For anti-aliasing text, ChartDirector supports the following modes.

ConstantValueDescription

NoAntiAlias	0	Disable anti-alias when drawing text
AntiAlias	1	Always use anti-alias when drawing text
AutoAntiAlias	2	Automatically determine if anti-alias should be used for the text. This is the default.

Currently, ChartDirector will anti-alias only large or bold fonts. For small fonts, assuming it is of high quality, anti-alias is unnecessary. It is because high quality fonts are normally designed to be sharp and clear at low resolution. Anti-aliasing will blur the fonts and make them look worse.

However, for complicated fonts (e.g. some fonts with oriental characters), or for lower quality fonts (e.g. some freeware fonts), anti-alias may be necessary. In this case, it may be needed to force anti-aliasing of all fonts using AntiAlias mode.

Argument	Default	Description
shapeAntiAlias	true	A true value enables anti-alias when drawing lines and shapes. A false value disables anti-alias when drawing lines and shapes
textAntiAlias	AutoAntiAlias	The text anti-alias mode, which must be one of AutoAntiAlias, AntiAlias or NoAntiAlias.

3.8.107 setBackground(colorvalue as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setBackground method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.108 setBackground(colorvalue as Integer, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0) 237

3.8.108 setBackground(colorvalue as Integer, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the background color, border color and 3D border effect of the chart.

Notes:

See also:

- 3.8.107 setBackground(colorvalue as color, edgeColor as color, raisedEffect as Integer = 0) 237

Argument	Default	Description
color	(Mandatory)	The background color of the chart.
edgeColor	Transparent	The border color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

3.8.109 setBgImage(img as string, align as Integer = 5)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses the image from the specified file as the background image of the chart.

Notes: ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to BaseChart.setSearchPath on the directory that ChartDirector will search for the file.

Argument	Default	Description
img	(Mandatory)	The image file that is used as the background image of the chart.
align	Center	The alignment of the background image relative to the chart. See Alignment Specification for supported alignment types.

3.8.110 setBorder(colorvalue as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setBorder method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.111 setBorder(colorvalue as Integer) 238

3.8.111 setBorder(colorvalue as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Deprecated. Use SetBackground instead.

See also:

- 3.8.110 setBorder(colorvalue as color) 238

3.8.112 setColor(paletteEntry as Integer, colorvalue as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.113 setColor(paletteEntry as Integer, colorvalue as Integer) 239

3.8.113 setColor(paletteEntry as Integer, colorvalue as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Change the color at the specified position in the palette.

Notes: See Color Specification on how colors are represented in ChartDirector.

Argument	Default	Description
paletteEntry	(Mandatory)	An index to the palette.
color	(Mandatory)	The color to change to.

See also:

- 3.8.112 setColor(paletteEntry as Integer, colorvalue as color) 239

3.8.114 setColors(numbers() as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setColors method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.115 setColors(numbers() as Integer) 239
- 3.8.116 setColors(paletteEntry as Integer, numbers() as color) 240
- 3.8.117 setColors(paletteEntry as Integer, numbers() as Integer) 240

3.8.115 setColors(numbers() as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Change the colors in the palette.

Notes: See Color Specification on how colors are represented in ChartDirector.

Argument	Default	Description
colors	(Mandatory)	An array of colors to change to.

See also:

- 3.8.114 `setColors(numbers() as color)` 239
- 3.8.116 `setColors(paletteEntry as Integer, numbers() as color)` 240
- 3.8.117 `setColors(paletteEntry as Integer, numbers() as Integer)` 240

3.8.116 `setColors(paletteEntry as Integer, numbers() as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setColors` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.8.114 `setColors(numbers() as color)` 239
- 3.8.115 `setColors(numbers() as Integer)` 239
- 3.8.117 `setColors(paletteEntry as Integer, numbers() as Integer)` 240

3.8.117 `setColors(paletteEntry as Integer, numbers() as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Change the colors in the palette, starting from the specified position in the palette.

Notes: See Color Specification on how colors are represented in `ChartDirector`.

Argument	Default	Description
paletteEntry	(Mandatory)	An index to the palette to start changing the colors.
colors	(Mandatory)	An array of colors to change to.

See also:

- 3.8.114 `setColors(numbers() as color)` 239
- 3.8.115 `setColors(numbers() as Integer)` 239
- 3.8.116 `setColors(paletteEntry as Integer, numbers() as color)` 240

3.8.118 setDefaultColors(paletteEntry as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the default colors.

3.8.119 setDefaultFonts(normal as string, bold as string, italic as string, boldItalic as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the defaults for normal, bold, italic and bold-italic fonts.

Example:

```
dim Chart as CDBaseChartMBS // your chart
```

```
#If TargetARM And TargetLinux Then
```

```
// use specific fonts on Linux on Raspberry Pi
```

```
Call Chart.setDefaultFonts("/usr/share/fonts/truetype/piboto/PibotoLt-Regular.ttf", "/usr/share/fonts/truetype/piboto/Pi
```

```
#EndIf
```

Notes: See Font Specification for details on various font attributes.

Argument	Default	Description
normal	(Mandatory)	The default normal font. This is the same as the first font in the font table.
bold	""	The default bold font. This is the same as the second font in the font table. An empty string means the default is unchanged.
italic	""	The default italic font. This is the same as the third font in the font table. An empty string means the default is unchanged.
boldItalic	""	The default bold-italic font. This is the same as the fourth font in the font table. An empty string means the default is unchanged.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

3.8.120 setDropShadow(ColorValue as color, OffsetX as Integer = 5, OffsetY as Integer = &h7ffffff, blurRadius as Integer = 5)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setDropShadow method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.121 `setDropShadow(ColorValue as Integer = &hAAAAAA, OffsetX as Integer = 5, OffsetY as Integer = &h7ffffff, blurRadius as Integer = 5)` 242

3.8.121 `setDropShadow(ColorValue as Integer = &hAAAAAA, OffsetX as Integer = 5, OffsetY as Integer = &h7ffffff, blurRadius as Integer = 5)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a drop shadow to the chart.

Notes: The drop shadow effect is created using a single color representation of the non-transparent part of the chart, offsetted by an amount, and put under the chart. The drop shadow can be blurred to create a soft drop shadow effect.

Note that adding a drop shadow will increase the width and height of the chart image so as to accommodate the drop shadow.

Because the drop shadow is located exterior to the original chart, it uses an exterior background color different from the background color of the original chart. The exterior background color is by default white, and can be configured with `CDBaseChartMBS.setRoundedFrame`.

Arguments:

Argument	Default	Description
color	AAAAAA	The color of the drop shadow.
offsetX	5	The x offset of the drop shadow.
offsetY	7ffffff	The y offset of the drop shadow. 7ffffff means it is the same as the x offset.
blurRadius	5	The blur radius of the drop shadow.

See also:

- 3.8.120 `setDropShadow(ColorValue as color, OffsetX as Integer = 5, OffsetY as Integer = &h7ffffff, blurRadius as Integer = 5)` 241

3.8.122 `SetFontSearchPath(path as folderitem)`

Plugin Version: 14.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the font search path.

Example:

```
dim FontFolder as folderitem = SpecialFolder.Desktop.Child("fonts")
CDBaseChartMBS.SetFontSearchPath FontFolder
```

Notes: This method can be used to configure the font search path. You may set your own font search path, or add additional search path before or after the default search path. The usage us like:

```
CDBaseChartMBS.setFontSearchPath("myPath1;myPath2;%PATH%;myPath3;myPath4");
```

In the above %PATH% (case sensitive) represents the default search path. This method must be called before the ChartDirector font system is used. It is suggested it be called before any ChartDirector methods. Once ChartDirector tries to look for the fonts (eg. to get font metrics so as to layout a chart), the search path cannot be changed without restarting the process.

e.g. if you use ubuntu, you can install the ttf-mscorefonts-installer package and call this method with "/usr/share/fonts/truetype/msttcorefonts" as the path. No backslash on the end of a path, please.
See also:

- 3.8.123 SetFontSearchPath(path as string)

243

3.8.123 SetFontSearchPath(path as string)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the font search path.

Example:

```
if TargetLinux then
CDBaseChartMBS.SetFontSearchPath "/usr/share/fonts/truetype;/usr/share/fonts/truetype/msttcorefonts"
else
// on Mac and Windows we use system fonts.
end if
```

Notes: This method can be used to configure the font search path. You may set your own font search path, or add additional search path before or after the default search path. The usage us like:

```
CDBaseChartMBS.setFontSearchPath("myPath1;myPath2;%PATH%;myPath3;myPath4");
```

In the above %PATH% (case sensitive) represents the default search path. This method must be called before the ChartDirector font system is used. It is suggested it be called before any ChartDirector methods. Once ChartDirector tries to look for the fonts (eg. to get font metrics so as to layout a chart), the search path cannot be changed without restarting the process.

e.g. if you use ubuntu, you can install the `ttf-mscorefonts-installer` package and call this method with `"/usr/share/fonts/truetype/msttcorefonts"` as the path. No backslash on the end of a path, please.

See also:

- 3.8.122 `SetFontSearchPath(path as folderitem)` 242

3.8.124 `setFontTable(index as Integer, font as string)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets an entry in the font table to the specified font name.

Notes: The first 4 fonts in the font table have special significance. They are the defaults for normal, bold, italic and bold-italic fonts.

See Font Specification for details on various font attributes.

Argument	Default	Description
<code>index</code>	(Mandatory)	An index to the font table, starting from 0.
<code>font</code>	(Mandatory)	The font name to be put into the font table.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

3.8.125 `setLicenseCode(n as string, enddate as Integer, v1 as Integer, v2 as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Registers the chartdirector plugin and library.

3.8.126 `setMonthNames(names() as string)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the names of the months for date/time formatting purposes.

Notes: The default is to use the first 3 characters of the English month names (Jan, Feb, Mar ...).

Argument	Default	Description
names	(Mandatory)	An array of 12 text strings to be used as the month names.

3.8.127 setNumberFormat(thousandSeparator as string = "textasciitilde ", decimalPointChar as string = ".", signChar as string = "_")

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the characters used for thousand separator, decimal point, and negative sign.

Notes:

Argument	Default	Description
thousandSeparator	'	
textasciitilde ' textasciitilde ' to mean no thousand separator.	The thousand separator. Use ' textasciitilde ' to mean no thousand separator.	
decimalPointChar	.'	The decimal point character.
signChar	'_'	The negative sign character.

3.8.128 setOutputOptions(options as string)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets output format options for the next chart output. Currently, only SVG, SVGZ and PDF output formats support output options.

Example:

```
dim c as CDBaseChartMBS // your chart object
```

```
// 2x for higher DPI displays
c.setOutputOptions("bmpscale=2")
```

Notes: An output option can be a flag (such as "compress") or an attribute-value pair (such as "width=800"). Multiple output options can be joined using semicolons as delimiters.

Raster Image Options (PNG, JPG, GIF, BMP or desktop display output)

Raster Image Option	Type	Description
bmpScale	Attribute	Render the chart with a scale factor. For example, "bmpScale=1.5" means the chart size will be 1.5 times of the original. The supported bmpScale is 0.1 to 4.
dpi	Attribute	Specify the factor for conversion from pixel to physical unit. e.g. 196 for double size.

SVG Options

SVG Option	Type	Description
compress	Flag	Compressed the SVG, that is, output SVGZ.
bitmap	Flag	Render the chart as a bitmap and output the bitmap as SVG.
noxmldecl	Flag	Do not include the xml declaration line " <code><?xml version="1.0" ... ></code> " in the SVG.
nodoctype	Flag	Do not include the document type declaration line " <code><!DOCTYPE svg PUBLIC ... ></code> " in the SVG.
width	Flag / Attribute	Specifies the width attribute of the SVG. By default, ChartDirector will not include the width or height attribute in the SVG output. In this case, the SVG is variable in size and would assume the size of its container. For example, if the SVG is inside a <code><DIV></code> block in a web page, it will assume the size of the DIV block. If the "width" option is used as a flag, ChartDirector will include the width attribute in the SVG and set it to the chart width. If the "width" option is used as an attribute (such as "width=800"), ChartDirector will include the width attribute in the SVG and set it to the specified value. The specified value should be some text that is valid as SVG width. Examples are "100" and "75%".
height	Flag / Attribute	Specifies the height attribute of the SVG. See the description on "width" above on how to use it.

PDF Options

3.8.129 setResource(id as string, data as MemoryBlock)

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns a resource ID to a memory block to allow it to be referenced using a resource path.

Notes: Many ChartDirector features, such as `BaseChart.setBgImage` and the `<img*>` tag in CDML, expect a file or resource path for loading an image. If the image happens to be in memory, such as if the image is retrieved from a database, `setResource` can be used to assign a resource ID to the memory. It can then be referenced using "`@/res_id`", in which `res_id` is the resource ID.

The assigned resource ID is only valid for the `DrawArea` object of which this method is called. Use `setResourceGlobal` to assign a resource ID that is valid for all ChartDirector objects.

This method stores only a pointer to the memory. It does not copy the memory. You must ensure the memory contains valid content for as long as the resource is being used.

While ChartDirector does not copy the data, the MBS Plugin will put the resources in a dictionary to make sure they stay available till the chart is destroyed and avoid a crash.

See also:

- 3.8.130 `setResource(id as string, data as string)`

PDF Option	Type	Description
bitmap	Flag	Render the chart as a bitmap and output the bitmap as PDF.
width	Attribute	The width of the chart in the PDF in pixel unit. By default, ChartDirector will use the pixel width of the chart as the width of the chart in PDF. The "width" attribute can be used to specify an alternative value. The value must be a number.
height	Attribute	The width of the chart in the PDF in pixel unit. See the description on "width" above for how to use it.
pagewidth	Attribute	The page width in pixel unit. By default, ChartDirector will set the page width to the same width as the chart. The "pagewidth" attribute can be used to specify an alternative value. The value must be a number.
pageheight	Attribute	The page height in pixel unit. By default, ChartDirector will set the page height to the same height as the chart. The "pageheight" attribute can be used to specify an alternative value. The value must be a number.
leftx	Attribute	The x coordinate of the left side of the chart within the page in pixel unit. By default, ChartDirector will center the chart in the page. The "leftx" attribute can be used to specify an alternative horizontal position. The coordinate must be a number.
topy	Attribute	The y coordinate of the top side of the chart within the page in pixel unit. By default, ChartDirector will center the chart in the page. The "topy" attribute can be used to specify an alternative vertical position. The coordinate must be a number.
dpi	Attribute	Specify the factor for conversion from pixel to physical unit. The PDF viewer will convert the pixel unit into physical unit (eg. inches) so that it can be layout on paper or other physical media. The default conversion factor for the chart is 96 pixels per inch. The "dpi" attribute can be used to specify an alternative value. The value must be a number.

Argument	Default	Description
options	(Mandatory)	A list of options delimited by semicolons.

- 3.8.131 setResource(id as string, DataArea as CDDrawAreaMBS)

248

3.8.130 setResource(id as string, data as string)

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns a resource ID to a memory block to allow it to be referenced using a resource path.

Notes: Many ChartDirector features, such as BaseChart.setBgImage and the <*img*>tag in CDML, expect a file or resource path for loading an image. If the image happens to be in memory, such as if the image is retrieved from a database, setResource can be used to assign a resource ID to the memory. It can then be referenced using "@/res_id", in which res_id is the resource ID.

The assigned resource ID is only valid for the DrawArea object of which this method is called. Use setRe-

Argument	Default	Description
id	(Mandatory)	The resource ID to be used to reference the memory image.
data	(Mandatory)	The memory that the image occupies.

sourceGlobal to assign a resource ID that is valid for all ChartDirector objects.

This method stores only a pointer to the memory. It does not copy the memory. You must ensure the memory contains valid content for as long as the resource is being used.

Argument	Default	Description
id	(Mandatory)	The resource ID to be used to reference the memory image.
data	(Mandatory)	The memory that the image occupies.

While ChartDirector does not copy the data, the MBS Plugin will put the resources in a dictionary to make sure they stay available till the chart is destroyed and avoid a crash.

See also:

- 3.8.129 setResource(id as string, data as MemoryBlock) 246
- 3.8.131 setResource(id as string, DataArea as CDDrawAreaMBS) 248

3.8.131 setResource(id as string, DataArea as CDDrawAreaMBS)

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns a resource ID to a DrawArea object to allow it to be referenced using a resource path.
Notes: This method is the same DrawArea.setResource, except that it assigns the resource ID to a CD-BaseChartMBS object instead of a memory image. This allows the CDBaseChartMBS object to be reference as an image resource using "@/res_id".

Argument	Default	Description
id	(Mandatory)	The resource ID to be used to reference the DrawArea object.
drawArea	(Mandatory)	The CDDrawAreaMBS object to be used as a resource.

While ChartDirector does not copy the draw area, the MBS Plugin will put the resources in a dictionary to make sure they stay available till the chart is destroyed and avoid a crash.

See also:

- 3.8.129 setResource(id as string, data as MemoryBlock) 246
- 3.8.130 setResource(id as string, data as string) 247

3.8.132 setResourceGlobal(id as string, data as MemoryBlock)

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns a resource ID to a memory block to allow it to be referenced using a resource path.

Notes: Many ChartDirector features, such as BaseChart.setBgImage and the <*img*>tag in CDML, expect a file or resource path for loading an image. If the image happens to be in memory, such as if the image is retrieved from a database, setResource can be used to assign a resource ID to the memory. It can then be referenced using "@/res_id", in which res_id is the resource ID.

This method stores only a pointer to the memory. It does not copy the memory. You must ensure the memory contains valid content for as long as the resource is being used.

Argument	Default	Description
id	(Mandatory)	The resource ID to be used to reference the memory image.
data	(Mandatory)	The memory that the image occupies.

While ChartDirector does not copy the data, the MBS Plugin will put the resources in a dictionary to make sure they stay available till the chart is destroyed and avoid a crash.

See also:

- 3.8.133 setResourceGlobal(id as string, data as string) 249
- 3.8.134 setResourceGlobal(id as string, DrawArea as CDDrawAreaMBS) 250

3.8.133 setResourceGlobal(id as string, data as string)

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns a resource ID to a memory block to allow it to be referenced using a resource path.

Notes: Many ChartDirector features, such as BaseChart.setBgImage and the <*img*>tag in CDML, expect a file or resource path for loading an image. If the image happens to be in memory, such as if the image is retrieved from a database, setResource can be used to assign a resource ID to the memory. It can then be referenced using "@/res_id", in which res_id is the resource ID.

This method stores only a pointer to the memory. It does not copy the memory. You must ensure the memory contains valid content for as long as the resource is being used.

Argument	Default	Description
id	(Mandatory)	The resource ID to be used to reference the memory image.
data	(Mandatory)	The memory that the image occupies.

While ChartDirector does not copy the data, the MBS Plugin will put the resources in a dictionary to make

sure they stay available till the chart is destroyed and avoid a crash.

See also:

- 3.8.132 `setResourceGlobal(id as string, data as MemoryBlock)` 249
- 3.8.134 `setResourceGlobal(id as string, DrawArea as CDDrawAreaMBS)` 250

3.8.134 `setResourceGlobal(id as string, DrawArea as CDDrawAreaMBS)`

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns a resource ID to a DrawArea object to allow it to be referenced using a resource path.
Notes: This method is the same DrawArea.setResource, except that it assigns the resource ID to a CD-BaseChartMBS object instead of a memory image. This allows the CD-BaseChartMBS object to be reference as an image resource using "@/res_id".

Argument	Default	Description
id	(Mandatory)	The resource ID to be used to reference the DrawArea object.
drawArea	(Mandatory)	The CDDrawAreaMBS object to be used as a resource.

While ChartDirector does not copy the draw area, the MBS Plugin will put the resources in a dictionary to make sure they stay available till the chart is destroyed and avoid a crash.

See also:

- 3.8.132 `setResourceGlobal(id as string, data as MemoryBlock)` 249
- 3.8.133 `setResourceGlobal(id as string, data as string)` 249

3.8.135 `setRoundedFrame(extColor as color, r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setRoundedFrame method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.136 `setRoundedFrame(extColor as Integer = &hFFFFFF, r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)` 250

3.8.136 `setRoundedFrame(extColor as Integer = &hFFFFFF, r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the border style of the chart to rounded corners.

Notes: The underlying drawing surface for a chart is always rectangular. When rounded corners are used, part of the drawing surface (the regions external to the rounded corners) will be outside the chart border. The `extColor` argument specifies the color to be used for the external regions. Typically, it is set to the same color as the background the container that will be hosting the chart.

For example, in a web page, the `extColor` may be set to the same color as the web page background.

Argument	Default	Description
<code>extColor</code>	FFFFFF	The exterior background color.
<code>r1</code>	10	The radius of the top-left rounded corner in pixels.
<code>r2</code>	-1	The radius of the top-right rounded corner in pixels. The default value of -1 means it is the same as the radius of the top-left corner.
<code>r3</code>	-1	The radius of the bottom-right rounded corner in pixels. The default value of -1 means it is the same as the radius of the top-left corner.
<code>r4</code>	-1	The radius of the bottom-left rounded corner in pixels. The default value of -1 means it is the same as the radius of the top-left corner.

See also:

- 3.8.135 `setRoundedFrame(extColor as color, r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)` 250

3.8.137 `setSearchPath(path as string)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the file system search path for loading image files.

Notes: The plugin uses `folderitems` for most file operations, so this method is not needed for most operations.

Several `ChartDirector` operations involve loading image files. Examples are wallpapers (`BaseChart.setWallpaper`), background images (`BaseChart.setBgImage` and `PlotArea.setBackground2`), user-defined symbols (`DataSet.setDataSymbol2`) or for embedding images in text using `ChartDirector Mark Up Language`.

Argument	Default	Description
<code>path</code>	(Mandatory)	A list of directories, separated with the path separator of your operating system (";" for Windows, ":" for Linux/UNIX).

3.8.138 `setSize(width as Integer, height as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the chart.

Notes:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.

3.8.139 `setThickFrame(thickness as Integer, frameColor as Integer = -1, outerEdgeColor as Integer = -1, innerEdgeColor as Integer = -1)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets a thick frame around the chart.

Notes: The thick frame can have 3 colors for the frame itself, the outer edge and the inner edge.

In some applications, the frame color may be similar to the external background color. An common example is a "silver" frame (which is actually a light grey gradient) against a white background. In this case, the frame effect may look less than optimal. An outer edge of different color (such as darker grey) can often help to highlight the frame. Similarly, if the frame color is similar to the chart background, an inner edge of different color may help to improve the frame effect.

Argument	Default	Description
thickness	(Mandatory)	The frame thickness.
frameColor	-1	The frame color. The default value of -1 means the frame color is the same as the outer edge color.
outerEdgeColor	-1	The outer edge color. The default value of -1 means the outer edge color is the same as the current edge color, which can be set in the chart constructor or in <code>BaseChart.setBackground</code> .
innerEdgeColor	-1	The inner edge color. The default value of -1 means the inner edge color is the same as the frame color.

3.8.140 `setTransparentColor(c as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setTransparentColor` method, but uses color instead of integer data type for passing color values.

See also:

- 3.8.141 `setTransparentColor(c as Integer)`

3.8.141 setTransparentColor(c as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Specifies a certain color to mean transparent when creating the image output, or to include alpha transparency channel in the output.

Notes: Alpha transparency: In addition to red, green and blue levels, there is a transparency level associated with each pixel, which can range from completely transparent to completely opaque. The data associated with the transparency information is called the alpha channel.

Single color transparency: The image itself has no alpha channel, but a certain color is used to mean completely transparent. For internal drawing, ChartDirector always use alpha transparency. However, when outputting the image as an image file, ChartDirector by default will remove the alpha channel to reduce image size. It is because many image displaying software do not support alpha transparency. For example, the IE browser only supports single color transparency but not alpha transparency.

If you want to use single color transparency in the output, you may specify the transparent color as the argument to the setTransparentColor method. Note that only GIF and PNG can support single color transparency. JPEG, BMP and WBMP cannot support transparency at all.

If you do want to keep the alpha channel in final output, you may pass -1 as the argument to setTransparentColor. Note that the only image format that can support alpha transparency is PNG.

One important thing to note is that the IE browser (and possibly many image displaying software) only supports single color transparency for palette based images with up to 256 colors, but not for true color images. For this reason, if single color transparency is used, ChartDirector will automatically reduce the image to 256 colors if it has more than 256 colors. This may result in lost of image quality, especially if the image contains gradient colors.

Therefore, due to the limitations of the current generations of image displaying software, for highest image quality, sometimes it may be beneficial to not using transparency in image output, but to set the image background color to the same color as the container background.

Argument	Default	Description
c	(Mandatory)	The color that is designated as the transparent color. If -1 is used, the full alpha transparency channel will be included in the final output.

See also:

- 3.8.140 setTransparentColor(c as color)

3.8.142 setTransparentColors(paletteEntry as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color palette to transparent colors.

3.8.143 setWallpaper(img as folderitem)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses the image from the specified file as the wallpaper to repetitively fill the background of the chart.

Notes: ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to BaseChart.setSearchPath on the directory that ChartDirector will search for the file.

Argument	Default	Description
img	(Mandatory)	The image file that is used as the background wallpaper of the chart.

3.8.144 setWeekDayNames(names() as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the names of the week days for date/time formatting purposes.

Notes: The default is to use the first 3 characters of the English week day names (Sun, Mon, Tue, ...).

Argument	Default	Description
names	(Mandatory)	An array of 7 text strings to be used as the week day names.

3.8.145 setWhiteOnBlackColors(paletteEntry as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color palette to white on black colors.

3.8.146 silverColor(angle as Integer = 90) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a silver color, most commonly used as a background color.

Notes: This method is a short cut to the CDBaseChartMBS.metalColor method, using grey (CCCCCC in hex) as the base color.

Argument	Default	Description
angle	90	The direction for brightness modulation, specified as a clockwise angle in degrees, with 0 being the upward pointing direction.

Return Value

A 32-bit integer representing the silver color.

3.8.147 silverGradient as Integer()

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A constant array of integers to represent a gradient that looks like a silver color.

Notes: The array is in a format that can be directly used in BaseChart.gradientColor2 and DrawArea.gradientColor2. Its contents (in hex) is:

```
00 C8C8C8 60 F8F8F8 B0 E0E0E0 100 C8C8C8
```

See Color Specification on how colors are represented in ChartDirector.

3.8.148 softLighting(direction as Integer = 8, raisedEffect as Integer = 4) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A special shading effect that looks like gradient coloring.

Notes: This effect is best explained by viewing the examples.

ExampleLocation of lighting effect

Soft Multi-Bar Chart	The bars are shaded using soft lighting effect, with light direction from Top, and raised effect of 4 pixels.
Soft Bar Shading	The bars are shaded using soft lighting effect, with light direction from Left, and raised effect of 4 pixels.
Spline Line Chart	The title is shaded using soft lighting effect, with light direction from Right, and raised effect of 4 pixels.

The `softLighting` method returns an integer representing this effect. The integer can be used as the third argument to `Box.setBackground` to apply the effect it objects derived from `Box` (such as labels and titles represented by `TextBox`). It may also be used as the second argument to `Layer.setBorderColor` for `BarLayer` objects to apply the effect to bars.

Argument	Default	Description
<code>direction</code>	Top	The direction of the lighting, which must be one of the predefined constants <code>Top</code> , <code>Bottom</code> , <code>Right</code> or <code>Left</code> .
<code>raisedEffect</code>	4	With soft lighting effect, the object will appear to have some 3D depth. The <code>raisedEffect</code> argument controls the amount of 3D depth in pixels.

Return Value

An integer representing the soft lighting effect.

3.8.149 `StarShape(slide as Integer) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the shape id that represents a star shape.

Notes: Please refer to Shape Specification for samples and more information on using shapes in `ChartDirector`.

Argument	Default	Description
<code>side</code>	(Mandatory)	The number of points the polygon has.

Return Value

An integer shape id representing the star shape.

3.8.150 `StartOfDayFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches date/times that represent the start of a new day in a date/time series.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` to select specific dates/times for formatting as axis labels.

In a date/time series, an element is considered that start of a new day if it is of a different day than the previous element. It does not need to be at exactly the starting instance of the day.

For the first element of the date/time series, because there is no previous element to compare with, it will be considered as the start of a new day if it is "near" the exact starting instance of the current day, in which "near" is defined using the `initialMargin` argument, expressed as a ratio (0 to 1) of the day duration.

Argument	Default	Description
<code>labelStep</code>	1	For dates/times that matches the start of day criteria, picks only 1 out of every "labelStep" number of elements. For example, if this argument is 3, only 1 of every 3 elements that at at the start of day will be selected.
<code>initialMargin</code>	0.05	If the first label is "near" the exact starting instance of the current day to within the ratio specified in the initial margin, it will be considered to have matched the start of day criteria.

Return Value

An integer filter id representing the filter.

3.8.151 StartOfHourFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches date/times that represent the start of a new hour in a date/time series.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` to select specific dates/times for formatting as axis labels.

In a date/time series, an element is considered that start of a new hour if it is of a different hour than the previous element. It does not need to be at exactly the starting instance of the hour.

For the first element of the date/time series, because there is no previous element to compare with, it will be considered as the start of a new hour if it is "near" the exact starting instance of the current hour, in which "near" is defined using the `initialMargin` argument, expressed as a ratio (0 to 1) of the hour duration.

Argument	Default	Description
<code>labelStep</code>	1	For dates/times that matches the start of hour criteria, picks only 1 out of every "labelStep" number of elements. For example, if this argument is 3, only 1 of every 3 elements that at at the start of hour will be selected.
<code>initialMargin</code>	0.05	If the first label is "near" the exact starting instance of the current hour to within the ratio specified in the initial margin, it will be considered to have matched the start of hour criteria.

Return Value

An integer filter id representing the filter.

3.8.152 StartOfMinuteFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches date/times that represent the start of a new minute in a date/time series.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` to select specific dates/times for formatting as axis labels.

In a date/time series, an element is considered that start of a new minute if it is of a different minute than the previous element. It does not need to be at exactly the starting instance of the minute.

For the first element of the date/time series, because there is no previous element to compare with, it will be considered as the start of a new minute if it is "near" the exact starting instance of the current minute, in which "near" is defined using the `initialMargin` argument, expressed as a ratio (0 to 1) of the minute duration.

Argument	Default	Description
<code>labelStep</code>	1	For dates/times that matches the start of minute criteria, picks only 1 out of every "labelStep" number of elements. For example, if this argument is 3, only 1 of every 3 elements that at at the start of minute will be selected.
<code>initialMargin</code>	0.05	If the first label is "near" the exact starting instance of the current minute to within the ratio specified in the initial margin, it will be considered to have matched the start of minute criteria.

Returns an integer filter id representing the filter.

3.8.153 StartOfMonthFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches date/times that represent the start of a new month in a date/time series.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` to select specific dates/times for formatting as axis labels.

In a date/time series, an element is considered that start of a new month if it is of a different month than the previous element. It does not need to be at exactly the starting instance of the month.

For the first element of the date/time series, because there is no previous element to compare with, it will be considered as the start of a new month if it is "near" the exact starting instance of the current month, in which "near" is defined using the initialMargin argument, expressed as a ratio (0 to 1) of the month duration.

Argument	Default	Description
labelStep	1	For dates/times that matches the start of month criteria, picks only 1 out of every "labelStep" number of elements. For example, if this argument is 3, only 1 of every 3 elements that at at the start of month will be selected.
initialMargin	0.05	If the first label is "near" the exact starting instance of the current month to within the ratio specified in the initial margin, it will be considered to have matched the start of month criteria.

Return Value

An integer filter id representing the filter.

3.8.154 StartOfSecondFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches date/times that represent the start of a new second in a date/time series.

Notes: This method is typically used in Axis.setMultiFormat and Axis.setMultiFormat2 to select specific dates/times for formatting as axis labels.

In a date/time series, an element is considered that start of a new second if it is of a different second than the previous element. It does not need to be at exactly the starting instance of the second.

For the first element of the date/time series, because there is no previous element to compare with, it will be considered as the start of a new second if it is "near" the exact starting instance of the current second, in which "near" is defined using the initialMargin argument, expressed as a ratio (0 to 1) of the second duration.

Argument	Default	Description
labelStep	1	For dates/times that matches the start of second criteria, picks only 1 out of every "labelStep" number of elements. For example, if this argument is 3, only 1 of every 3 elements that at at the start of second will be selected.
initialMargin	0.05	If the first label is "near" the exact starting instance of the current second to within the ratio specified in the initial margin, it will be considered to have matched the start of second criteria.

Returns an integer filter id representing the filter.

3.8.155 StartOfWeekFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches date/times that represent the start of a new week in a date/time series.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` to select specific dates/times for formatting as axis labels.

In a date/time series, an element is considered that start of a new week if it is of a different week than the previous element. It does not need to be at exactly the starting instance of the week.

For the first element of the date/time series, because there is no previous element to compare with, it will be considered as the start of a new week if it is "near" the exact starting instance of the current week, in which "near" is defined using the `initialMargin` argument, expressed as a ratio (0 to 1) of the week duration.

Argument	Default	Description
<code>labelStep</code>	1	For dates/times that matches the start of week criteria, picks only 1 out of every "labelStep" number of elements. For example, if this argument is 3, only 1 of every 3 elements that at at the start of week will be selected.
<code>initialMargin</code>	0.05	If the first label is "near" the exact starting instance of the current week to within the ratio specified in the initial margin, it will be considered to have matched the start of week criteria.

Return Value

An integer filter id representing the filter.

3.8.156 StartOfYearFilter(labelStep as Integer = 1, initialMargin as Double = 0.05) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a data filter that matches date/times that represent the start of a new year in a date/time series.

Notes: This method is typically used in `Axis.setMultiFormat` and `Axis.setMultiFormat2` to select specific dates/times for formatting as axis labels.

In a date/time series, an element is considered that start of a new year if it is of a different year than the

previous element. It does not need to be at exactly the starting instance of the year.

For the first element of the date/time series, because there is no previous element to compare with, it will be considered as the start of a new year if it is "near" the exact starting instance of the current year, in which "near" is defined using the initialMargin argument, expressed as a ratio (0 to 1) of the year duration.

Argument	Default	Description
labelStep	1	For dates/times that matches the start of year criteria, picks only 1 out of every "labelStep" number of elements. For example, if this argument is 3, only 1 of every 3 elements that at at the start of year will be selected.
initialMargin	0.05	If the first label is "near" the exact starting instance of the current year to within the ratio specified in the initial margin, it will be considered to have matched the start of year criteria.

Return Value

An integer filter id representing the filter.

3.8.157 testFont(font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, byref buffer as string) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A diagnostic function to perform a font loading test.

Example:

```
dim buffer as string
call CDBaseChartMBS.testFont("arial.ttf", 0, 12, 12, 0, buffer)
MsgBox buffer
```

Notes: From experience, the most common issue for font loading is unable to access server side fonts using anonymous user account for a web application, probably due to security restrictions. This diagnostic function can return the cause of problem to aid trouble-shooting.

Other uses of this function is to trace out where does ChartDirector search for the fonts, and the substitution font in case the request font is not available.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

Argument	Default	Description
font	(Mandatory)	The font name. See Font Specification for details on various font attributes.
fontIndex	(Mandatory)	The font index if the font name refers to a font collection. An index of 0 means the first font.
fontHeight	(Mandatory)	The font height in points. This parameter will not affect font loading if the exact font exists, but will affect which substitution font to use if the font does not exist.
fontWidth	(Mandatory)	The font width in points. This parameter will not affect font loading if the exact font exists, but will affect which substitution font to use if the font does not exist.
angle	(Mandatory)	The rotation angle of the text. The angle is measured in degrees in clockwise direction. This parameter will not affect font loading if the exact font exists, but will affect which substitution font to use if the font does not exist.
buffer	(Mandatory)	A string to hold the result of the font loading test.

3.8.158 transparentPalette as Integer()

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the default transparent palette.

3.8.159 whiteOnBlackPalette as Integer()

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the default white on black palette.

3.8.160 xySize(x as Integer, y as Integer) as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Encode width and height into a single number to be used as size.

Notes: This is intended to be used in certain ChartDirector API that that supports the encoded value.

Argument	Default	Description
width	(Mandatory)	An integer between 0 and 30000.
height	(Mandatory)	An integer between 0 and 30000.

Returns a number that encodes the width and height.

3.8.161 Properties

3.8.162 Handle as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal object reference.

Notes: (Read only property)

3.8.163 Resources as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal resource dictionary.

Notes: Stores references for registered resources.

Just for debugging.

(Read only property)

3.8.164 ScaleFactor as Double

Plugin Version: 16.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Scale factor for charts.

Example:

```
dim c as CDBaseChartMBS // your chart object
```

```
// 2x for higher DPI displays
c.setOutputOptions("bmpscale=2")
' better than c.ScaleFactor = 2
```

Notes: When the plugin internally passes a pixel number to ChartDirector library, we multiply by this factor.

When numbers come back we divide by factor.

Default is 1.

You may better do use setOutputOptions function to set scale factor.

(Read and Write property)

3.8.165 Constants

Constants

Constant	Value	Description
kCircleLayout	1	A constant equals to 1 to represent the circular label layout method for pie/donut charts. Circle Layout This constant is used in PieChart.setLabelLayout.
kKeepAspectRatio	3	A constant equals to 3 to represent can be both horizontal and vertical and the zoom level will be kept the same in both directions. This constant is used in CChartViewer.setZoomDirection and QChartViewer.setZoomDirection .
kNewShape	-1342177279	A constant equals to &h4ffffff in hexadecimal to represent a shape separator in custom shape definition. This constant is used in custom shape definition in DrawArea.polyShape, DataSet.setDataSymbol4 and PolarLayer.setDataSymbol4.
kSideLayout	0	A constant equals to 0 to represent the side label layout method for pie/donut charts. Side Layout This constant is used in PieChart.setLabelLayout.

Aggregate Modes

Constant	Value	Description
kAggregateAvg	1	A constant equals to 1 to represent using the average as the aggregated value in CdataArrayMBS.aggregate.
kAggregateCount	9	A constant equals to 9 to represent using the item count as the aggregated value in CdataArrayMBS.aggregate.
kAggregateFirst	7	A constant equals to 7 to represent using the first value as the aggregated value in CdataArrayMBS.aggregate.
kAggregateLast	8	A constant equals to 8 to represent using the last value as the aggregated value in CdataArrayMBS.aggregate.
kAggregateMax	5	A constant equals to 5 to represent using the maximum value as the aggregated value in CdataArrayMBS.aggregate.
kAggregateMed	4	A constant equals to 4 to represent using the median value as the aggregated value in CdataArrayMBS.aggregate.
kAggregateMin	3	A constant equals to 3 to represent using the minimum value as the aggregated value in CdataArrayMBS.aggregate.
kAggregatePercentile	6	A constant equals to 6 to represent using the percentile value as the aggregated value in CdataArrayMBS.aggregate.
kAggregateStdDev	2	A constant equals to 2 to represent using the standard deviation as the aggregated value in CdataArrayMBS.aggregate.
kAggregateSum	0	A constant equals to 0 to represent using the sum as the aggregated value in CdataArrayMBS.aggregate.

Line Modes

Constant	Value	Description
kAltDashLine	&h0A050505	A constant equals to 0A050505 (in hex) to represent a alternating long/short dash line pattern for use in dash colors. See Color Specification on how colors are represented in ChartDirector.
kDashLine	&h0505	A constant equals to 00000505 (in hex) to represent a dash line pattern for use in dash colors. See Color Specification on how colors are represented in ChartDirector.
kDotDashLine	&h05050205	A constant equals to 05050205 (in hex) to represent a dot-dash line pattern for use in dash colors. See Color Specification on how colors are represented in ChartDirector.
kDotLine	&h0202	A constant equals to 00000202 (in hex) to represent a dot-line pattern for use in dash colors. See Color Specification on how colors are represented in ChartDirector.

Scale Modes

Constant	Value	Description
kAngularAxisScale	1	A constant equals to 1 to represent that the size is measured using the angular-axis scale. This constant is used in PolarLayer.setSymbolScale.
kEndPoints	3	A constant equals to 3 to represent that the vector lengths and directions are measured by specifying the end points.
kPixelScale	0	A constant equals to 0 to represent that the size is measured in pixels. This constant is used in LineLayer.setSymbolScale, PolarLayer.setSymbolScale, XYChart.addVectorLayer, PolarChart.addVectorLayer, VectorLayer.setVector and PolarVectorLayer.setVector.
kRadialAxisScale	2	A constant equals to 2 to represent that the size is measured using the radial-axis scale. This constant is used in PolarLayer.setSymbolScale, PolarChart.addVectorLayer and PolarVectorLayer.setVector.
kXAxisScale	1	A constant equals to 1 to represent that the size is measured using the x-axis scale. This constant is used in LineLayer.setSymbolScale, XYChart.addVectorLayer and VectorLayer.setVector.
kYAxisScale	2	A constant equals to 2 to represent that the size is measured using y-axis scale. This constant is used in LineLayer.setSymbolScale, XYChart.addVectorLayer and VectorLayer.setVector.

Anti Alias Modes

Constant	Value	Description
kAntiAlias	1	Anti-Aliasing This constant is used in BaseChart.setAntiAlias and DrawArea.setAntiAlias.
kAutoAntiAlias	2	Auto Anti-Aliasing This constant is used in BaseChart.setAntiAlias and DrawArea.setAntiAlias.
kClearType	3	Clear Type
kCompatAntiAlias	6	Compat Anti-Aliasing This constant is used in BaseChart.setAntiAlias and DrawArea.setAntiAlias.
kNoAntiAlias	0	No Anti-Aliasing This constant is used in BaseChart.setAntiAlias and DrawArea.setAntiAlias.

Pointers

Constant	Value	Description
kArrowPointer	2	A constant equals to 2 to represent the arrow style meter pointer. This constant is used in MeterPointer.setShape.
kArrowPointer2	3	A constant equals to 3 to represent the alternative arrow style meter pointer. This constant is used in MeterPointer.setShape.
kDiamondPointer	0	A constant equals to 0 to represent the diamond style meter pointer. This constant is used in MeterPointer.setShape.
kLinePointer	4	A constant equals to 4 to represent the line style meter pointer. This constant is used in MeterPointer.setShape.
kLinePointer2	7	A constant equals to 7 to represent the line style meter pointer. This constant is used in MeterPointer.setShape.
kPencilPointer	5	A constant equals to 5 to represent the pencil style meter pointer. This constant is used in MeterPointer.setShape.
kTriangularPointer	1	A constant equals to 1 to represent the triangular style meter pointer. This constant is used in MeterPointer.setShape.
kTriangularPointer2	6	A constant equals to 6 to represent the new triangular style meter pointer.

Special Values

Constant	Value	Description
kAutoGrid	-2	A constant equals to -2 to represent that the number of columns in the legend box with grid layout is automatically determine. This constant is used in BaseChart.addLegend and LegendBox.setCols.

Special Colors

Constant	Value	Description
kBackgroundColor	&hFFFFFF000	A constant equals to FFFF0000 (in hex) to represent the background color. See Color Specification on how colors are represented in ChartDirector.
kDataColor	&hFFFFFF008	A constant equals to FFFF0008 (in hex) to represent the starting index of automatic data color. The array is in a format that can be directly used in BaseChart.setColors and DrawArea.setColorTable.
kLineColor	&hFFFFFF001	See Color Specification on how colors are represented in ChartDirector. A constant equals to FFFF0001 (in hex) to represent the default line color.
kPalette	&hFFFFFF000	See Color Specification on how colors are represented in ChartDirector. A constant equals to FFFF0000 (in hex) to represent the starting index of the color palette.
kSameAsMainColor	&hFFFFFF007	See Color Specification on how colors are represented in ChartDirector. A constant equals to FFFF0007 (in hex) to represent the current main color.
kTextColor	&hFFFFFF002	See Color Specification on how colors are represented in ChartDirector. A constant equals to FFFF0002 (in hex) to represent the default text color.
kTransparent	&hFF000000	See Color Specification on how colors are represented in ChartDirector. A constant equals to FF000000 (in hex) to represent the transparent color.

Filter Modes

Constant	Value	Description
kBesselFilter	13	A constant equals to 13 to represent a Bessel graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kBlackmanFilter	12	A constant equals to 12 to represent a Blackman graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kBoxFilter	0	A constant equals to 0 to represent a Box graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kBSplineFilter	3	A constant equals to 3 to represent a B-spline graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kCatromFilter	5	A constant equals to 5 to represent a Catrom graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kGaussianFilter	9	A constant equals to 9 to represent a Gaussian graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kHammingFilter	11	A constant equals to 11 to represent a Hamming graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kHanningFilter	10	A constant equals to 10 to represent a Hanning graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kHermiteFilter	4	A constant equals to 4 to represent a Hermite graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kLanczosFilter	8	A constant equals to 8 to represent a Lanczos graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kLinearFilter	1	A constant equals to 1 to represent a Linear graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kMitchellFilter	6	A constant equals to 6 to represent a Mitchell graphical re-sampling filter. Please refer to Re-Sampling Filters for more information graphical filters in ChartDirector.
kQuadraticFilter	2	
kSincFilter	7	

Image Formats

Constant	Value	Description
kBMP	4	The BMP format. This constant is used in BaseChart.makeChart.
kGIF	1	The GIF format. This constant is used in BaseChart.makeChart.
kJPG	2	The JPEG format. This constant is used in BaseChart.makeChart.
kPDF	7	The PDF format.
kPNG	0	The PNG format. This constant is used in BaseChart.makeChart.
kQIMG	9	The QT Image format.
kSVG	5	The SVG format.
kSVGZ	6	The compressed SVG format.
kWMP	3	The WAP bitmap format. This constant is used in BaseChart.makeChart.

Alignment

Constant	Value	Description
kBottom	2	A constant equals to 2 to represent the bottom position. See Alignment Specification for supported alignment types.
kBottomCenter	2	A constant equals to 2 to represent the bottom center position. See Alignment Specification for supported alignment types.
kBottomLeft	1	A constant equals to 1 to represent the bottom left position. See Alignment Specification for supported alignment types.
kBottomLeft2	12	A constant equals to 12 to represent the alternative exterior bottom left position. See Alignment Specification for supported alignment types.
kBottomRight	3	A constant equals to 3 to represent the bottom right position. See Alignment Specification for supported alignment types.
kBottomRight2	13	A constant equals to 13 to represent the alternative exterior bottom right position. See Alignment Specification for supported alignment types.
kCenter	5	A constant equals to 5 to represent the center position. See Alignment Specification for supported alignment types.
kLeft	4	A constant equals to 4 to represent the left position. See Alignment Specification for supported alignment types.
kRight	6	A constant equals to 6 to represent the right position. See Alignment Specification for supported alignment types.
kTop	8	A constant equals to 8 to represent the top position. See Alignment Specification for supported alignment types.
kTopCenter	8	A constant equals to 8 to represent the top center position. See Alignment Specification for supported alignment types.
kTopLeft	7	A constant equals to 7 to represent the top left position. See Alignment Specification for supported alignment types.
kTopLeft2	10	A constant equals to 10 to represent the alternative exterior top left position. See Alignment Specification for supported alignment types.
kTopRight	9	A constant equals to 9 to represent the top right position. See Alignment Specification for supported alignment types.
kTopRight2	11	A constant equals to 11 to represent the alternative exterior top right position. See Alignment Specification for supported alignment types.

Z-Level

Constant	Value	Description
kChartBackZ	&h100	A constant equals to 100 (in hex) to represent the z-order of the back surface of the chart. This constant is used in Box.setZOrder, Line.setZOrder and MeterPointer.setZOrder.
kChartFrontZ	&hffff	A constant equals to ffff (in hex) to represent the z-order of the front surface of the chart. This constant is used in Box.setZOrder, Line.setZOrder and MeterPointer.setZOrder.
kGridLinesZ	&h2000	A constant equals to 2000 (in hex) to represent the z-order of the grid lines of the chart. This constant is used in Box.setZOrder, Line.setZOrder and MeterPointer.setZOrder.
kPlotAreaZ	&h1000	A constant equals to 1000 (in hex) to represent the z-order of the plot area back surface of the chart. This constant is used in Box.setZOrder, Line.setZOrder and MeterPointer.setZOrder.

Filter Tags

Constant	Value	Description
kChartDir_AllPassFilterTag	7	All pass filter
kChartDir_NonePassFilterTag	8	None pass
kChartDir_RegularSpacingFilterTag	6	Regular Spacing
kChartDir_SelectItemFilterTag	9	Select Item
kChartDir_StartOfDayFilterTag	2	Start of Day
kChartDir_StartOfHourFilterTag	1	Start of Hour
kChartDir_StartOfMinuteFilterTag	10	Start of minute
kChartDir_StartOfMonthFilterTag	4	Start of month
kChartDir_StartOfSecondFilterTag	11	Start of second
kChartDir_StartOfWeekFilterTag	3	Start of week
kChartDir_StartOfYearFilterTag	5	Start of year

Symbols

Constant	Value	Description
kChartDir_ArrowSymbol	18	Arrow
kChartDir_CustomSymbol	14	Custom
kChartDir_Polygon2Symbol	12	Polygon 2
kChartDir_PolygonSymbol	11	Polygon
kChartDir_StarSymbol	13	Star
kCircleSymbol	7	Circle
kCross2Symbol	9	Cross 2
kCrossSymbol	8	Cross
kDiamondSymbol	2	Diamond
kInvertedTriangleSymbol	6	Inverted Triangle
kLeftTriangleSymbol	5	Left Triangle
kNoSymbol	0	No symbol
kRightTriangleSymbol	4	Right Triangle
kSquareSymbol	1	Square
kTriangleSymbol	3	Triangle

Shapes

Constant	Value	Description
kCircleShape	7	Circle Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.
kCircleShapeNoShading	10	Circle without shading. Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.
kDiamondShape	2	Diamond Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.
kGlassSphere2Shape	16	Glas Sphere 2
kGlassSphereShape	15	Glas Sphere
kInvertedTriangleShape	6	Inverted Triangle Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.
kLeftTriangleShape	5	Left Triangle Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.
kNoShape	0	No shape
kRightTriangleShape	4	Triangle Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.
kSolidSphereShape	17	Solid shape
kSquareShape	1	Square shape Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.
kTriangleShape	3	Triangle Please refer to Shape Specification for samples and more information on using shapes in ChartDirector.

Shadings

Constant	Value	Description
kConcaveShading	4	Concave shading
kDefaultShading	0	Default shading
kFlatShading	1	Flat shading
kGlobalGradientShading	3	Global gradient shading
kLocalGradientShading	2	Local gradient shading
kRadialShading	7	Radial shading
kRingShading	8	Ring shading
kRoundedEdgeNoGlareShading	5	Rounded edge and no glare
kRoundedEdgeShading	6	Rounded Edge

Regression Modes

Constant	Value	Description
kConstrainedLinearRegression	0	A constant equals to 0 to represent that constrained linear regression be used to draw a trend line.
kExponentialRegression	-1	A constant equals to -1 to represent that exponential regression be used to draw a trend line.
kLinearRegression	1	A constant equals to 1 to represent that linear regression be used to draw a trend line.
kLogarithmicRegression	-2	A constant equals to -2 to represent that logarithmic regression be used to draw a trend line. This constant is used in <code>CDTrendLayerMBS.setRegressionType</code> .

Data Combine Methods

Constant	Value	Description
kDepth	2	A constant equals to 2 to represent the "Depth" data representation method for multiple data sets. This constant is used in <code>XYChart.addBarLayer</code> , <code>XYChart.addAreaLayer</code> , and <code>Layer.setDataCombineMethod</code> .
kOverlay	0	A constant equals to 0 to represent the "Overlay" data representation method for multiple data sets. This constant is used in <code>XYChart.addBarLayer</code> , <code>XYChart.addAreaLayer</code> , and <code>Layer.setDataCombineMethod</code> .
kPercentage	4	A constant equals to 4 to represent the "Percentage" data representation method for multiple data sets. This constant is used in <code>XYChart.addBarLayer</code> , <code>XYChart.addAreaLayer</code> and <code>Layer.setDataCombineMethod</code> .
kSide	3	A constant equals to 3 to represent the "Side" data representation method for multiple data sets. This constant is used in <code>XYChart.addBarLayer</code> , <code>XYChart.addAreaLayer</code> and <code>Layer.setDataCombineMethod</code> .
kStack	1	A constant equals to 1 to represent the "Stack" data representation method for multiple data sets. This constant is used in <code>XYChart.addBarLayer</code> , <code>XYChart.addAreaLayer</code> and <code>Layer.setDataCombineMethod</code> .

Directions

Constant	Value	Description
kDirectionHorizontal	0	A constant equals to 0 to represent that the zoom and/or scroll orientation is horizontal. This constant is used in CChartViewer.setZoomDirection and CChartViewer.setScrollDirection.
kDirectionHorizontalVertical	2	A constant equals to 2 to represent that the zoom and/or scroll orientation can be both horizontal and vertical. This constant is used in CChartViewer.setZoomDirection and CChartViewer.setScrollDirection.
kDirectionVertical	1	A constant equals to 1 to represent that the zoom and/or scroll orientation is vertical. This constant is used in CChartViewer.setZoomDirection and CChartViewer.setScrollDirection.

Dithering Modes

Constant	Value	Description
kErrorDiffusion	2	Error Diffusion This constant is used in DrawArea.setDitherMethod.
kOrderedDither	1	Ordered Dither This constant is used in DrawArea.setDitherMethod.
kQuantize	0	Quantize This constant is used in DrawArea.setDitherMethod.

Palette Modes

Constant	Value	Description
kForcePalette	1	Force palette. This constant is used in DrawArea.setPaletteMode.
kNoPalette	2	No palette. This constant is used in DrawArea.setPaletteMode.
kTryPalette	0	Try palette. This constant is used in DrawArea.setPaletteMode.

HLOC

Constant	Value	Description
kHLOCDefault	0	A constant equals to 0 to represent that the HLOC symbols should be drawn using the same color. This constant is used in <code>XYChart.addHLOCLayer</code> and <code>HLOCLayer.setColorMethod</code> .
kHLOCOpenClose	1	A constant equals to 1 to represent that the HLOC symbols should be drawn using two alternative colors based on whether the closing price is higher than the opening price. This constant is used in <code>XYChart.addHLOCLayer</code> and <code>HLOCLayer.setColorMethod</code> .
kHLOCUpDown	2	A constant equals to 2 to represent that the HLOC symbols should be drawn using two alternative colors based on whether the closing price is higher than the previous closing price. This constant is used in <code>XYChart.addHLOCLayer</code> and <code>HLOCLayer.setColorMethod</code> .

Monotonic Modes

Constant	Value	Description
kMonotonicAuto	4	A constant equals to 4 to represent that the system will automatically determine whether to constrained a spline curve to not overshooting or undershooting in the x-axis direction and/or the y-axis direction.
kMonotonicNone	0	A constant equals to 0 to represent that a spline curve is not constraint to flow in any direction.
kMonotonicX	1	A constant equals to 1 to represent that a spline curve is constrained to not overshooting or undershooting in the x-axis direction.
kMonotonicXY	3	A constant equals to 3 to represent that a spline curve is constrained to not overshooting or undershooting in both the x-axis direction and the y-axis direction.
kMonotonicY	2	A constant equals to 2 to represent that a spline curve is constrained to not overshooting or undershooting in the y-axis direction.

Glare Modes

Constant	Value	Description
kNoGlare	1	A constant equals to 1 to represent disabling the glare in <code>CD-BaseChartMBS.glassEffect</code> shading style.
kNormalGlare	3	A constant equals to 3 to represent using normal glare strength in <code>CD-BaseChartMBS.glassEffect</code> shading style.
kReducedGlare	2	A constant equals to 2 to represent using reduced glare strength in <code>CD-BaseChartMBS.glassEffect</code> shading style.

Legend Modes

Constant	Value	Description
kNoLegend	2	A constant equals to 2 to represent that no legend keys should be added to the legend box. This constant is used in Layer.setLegendOrder.
kNormalLegend	0	A constant equals to 0 to represent that the legend keys order should follow the creation order of the data sets. This constant is used in Layer.setLegendOrder.
kReverseLegend	1	A constant equals to 1 to represent that the legend keys order is the reverse of the creation order of the data sets. This constant is used in Layer.setLegendOrder.

Surface Shading Style

Constant	Value	Description
kRectangularFrame	4	
kRectangularShading	2	Rectangular sharing
kSmoothShading	0	Smooth
kTriangularFrame	3	
kTriangularShading	1	Triangular

Layout Methods

Constant	Value	Description
kTreeMapBinaryByMid	5	Same as TreeMapBinaryBySize except that the nodes will be partitioned so that the count of nodes are as equal as possible.
kTreeMapBinaryBySize	4	Without changing the ordering of the nodes, the nodes will be partitioned into two groups so that their sizes are as equal as possible. The two groups will then become two rectangular region. If the width of the plot area is not smaller than its height, the two groups will be placed at the left and right sides of plot area, otherwise they will be placed at the top and bottom sides of the plot area. The splitting then recursively apply to each group to partition them, until each subgroup contains only one node and can no longer be partitioned.
kTreeMapNoLayout	6	No layout.
kTreeMapSliceAndDice	1	This method is intended for multi-level tree maps. Assume the layout direction is TopLeft, the first level nodes will flow from left to right. The second level nodes will flow from top to bottom. If there are more levels, they will alternate between left/right and top/bottom flow directions. The layoutDirection argument can be used to specify other layout directions.
kTreeMapSquarify	2	The nodes will be layout so that they are as close to a square as possible. This method may sort the nodes, so the ordering of the nodes on the chart may be different from the ordering in the data array.
kTreeMapStrip	3	The nodes will flow according to the layoutDirection argument. The default is TopLeft, which means the nodes will flow from left to right and top to bottom like text. The number of nodes in each row will be such that on average, they are as close to squares as possible.

At Origin

Constant	Value	Description
kXAxisAtOrigin	1	A constant equals to 1 to represent that the x-axis should intersect with the zero point of the y-axis if it exists on the chart. This constant is used in <code>XYChart.setAxisAtOrigin</code> .
kXYAxisAtOrigin	3	A constant equals to 3 to represent that the x-axis and y-axis should intersect at the origin if it exists on the chart. This constant is used in <code>XYChart.setAxisAtOrigin</code> .
kYAxisAtOrigin	2	A constant equals to 2 to represent that the y-axis should intersect with the zero point of the x-axis if it exists on the chart. This constant is used in <code>XYChart.setAxisAtOrigin</code> .

Symmetric Modes

Constant	Value	Description
kXAxisSymmetric	1	A constant equals to 1 to represent that the x-axis should be symmetrical about the origin. This constant is used in <code>XYChart.setAxisAtOrigin</code> .
kXAxisSymmetricIfNeeded	2	A constant equals to 2 to represent that the x-axis should be symmetrical about the origin if the data contain both positive and negative values. This constant is used in <code>XYChart.setAxisAtOrigin</code> .
kXYAxisSymmetric	16	A constant equals to 16 to represent that the the x-axis and y-axis should be symmetrical about the origin. This constant is used in <code>XYChart.setAxisAtOrigin</code> .
kXYAxisSymmetricIfNeeded	32	A constant equals to 32 to represent that the x-axis and y-axis should be symmetrical about the origin if the data contain both positive and negative values. This constant is used in <code>XYChart.setAxisAtOrigin</code> .
kYAxisSymmetric	4	A constant equals to 4 to represent that the y-axis should be symmetrical about the origin. This constant is used in <code>XYChart.setAxisAtOrigin</code> .
kYAxisSymmetricIfNeeded	8	A constant equals to 8 to represent that the y-axis should be symmetrical about the origin if the data contain both positive and negative values. This constant is used in <code>XYChart.setAxisAtOrigin</code> .

3.9 class CDBaseMeterMBS

3.9.1 class CDBaseMeterMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The BaseMeter class represents a generic meter.

Notes: Subclass of the CDBaseChartMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [ChartDirector Meters and Gauges](#)

3.9.2 Methods

3.9.3 addColorScale(colorStops() as Double, startPos as Integer = -2147483647, startWidth as Integer = -2147483647, endPos as Integer = -2147483647, endWidth as Integer = -2147483647, edgeColor as Integer = -1)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a color scale to the meter.

Notes: A color scale is similar to a meter scale. Instead of using labels to denote the values, a color scale uses different colors for different values. The colors can vary continuously or in discrete steps. The width of the color scale can also vary based on the value. There can be multiple more than one color scale in a meter.

A color scale is defined with an array of numbers. For a continuous color scale, each pair of numbers represents a value and its associated color. For example, to define a continuous color scale in which 0 is blue (0000FF), 50 is yellow (FFFF00) and 100 is red (FF0000), the numbers should be:

```
0.0, &h0000ff, 50.0, &hffff00, 100.0, &hff0000
```

For a step color scale, the number of colors would be one less than the number of values. For example, to define a step color scale in which 0 to 50 is red (FF0000), and 50 to 100 is green (00FF00), the numbers will be:

```
0.0, &hff0000, 50.0, &h00ff00, 100.0
```

See also:

- 3.9.4 addColorScale(colorStops() as Double, startPos as Integer, startWidth as Integer, endPos as Integer, endWidth as Integer, edgeColor as color) 280

Argument	Default	Description
colorStops	(Mandatory)	An array of numbers alternating between values and colors. If the number count is even, the array will define a continuous color scale, otherwise it will define a step color scale.
startPos	<code>~&h7ffffff</code>	The position of the starting point of the color scale. For an angular meter, the position refers to the radius, and the default is the meter scale radius (see <code>AngularMeter.setMeter</code>). For a linear meter, the position refers to the x or y coordinate depending on whether the meter is vertical or horizontal, and the default is the leftX or topY coordinate of the meter scale (see <code>LinearMeter.setMeter</code>).
startWidth	<code>~&h7ffffff</code>	The width at the starting point of the color scale. A positive value means the width is in the direction of increasing "position", while a negative value means decreasing "position". Please refer to the description of <code>startPos</code> on the meaning of "position" for various types of meters. The default is the length of the major tick (see <code>BaseMeter.setTickLength</code>).
endPos	<code>~&h7ffffff</code>	The position of the ending point of the color scale. Please refer to the description of <code>startPos</code> above on how the position parameter is interpreted. The default is for the ending position to be equal to the starting position.
endWidth	<code>~&h7ffffff</code>	The width at the ending point of the color scale. Please refer to the description of <code>startWidth</code> above on how the width parameter is interpreted. The default is for the ending width to be equal to the starting width.
edgeColor	-1	The edge color of the color scale. The default is to have no edge.

3.9.4 `addColorScale(colorStops()` as Double, `startPos` as Integer, `startWidth` as Integer, `endPos` as Integer, `endWidth` as Integer, `edgeColor` as color)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a color scale to the meter.

Notes: A color scale is similar to a meter scale. Instead of using labels to denote the values, a color scale uses different colors for different values. The colors can vary continuously or in discrete steps. The width of the color scale can also vary based on the value. There can be multiple more than one color scale in a meter.

A color scale is defined with an array of numbers. For a continuous color scale, each pair of numbers represents a value and its associated color. For example, to define a continuous color scale in which 0 is blue (0000FF), 50 is yellow (FFFF00) and 100 is red (FF0000), the numbers should be:

```
0.0, &h0000ff, 50.0, &hffff00, 100.0, &hff0000
```

For a step color scale, the number of colors would be one less than the number of values. For example, to define a step color scale in which 0 to 50 is red (FF0000), and 50 to 100 is green (00FF00), the numbers will be:

```
0.0, &hff0000, 50.0, &h00ff00, 100.0
```

Argument	Default	Description
colorStops	(Mandatory)	An array of numbers alternating between values and colors. If the number count is even, the array will define a continuous color scale, otherwise it will define a step color scale.
startPos	-&h7ffffff	The position of the starting point of the color scale. For an angular meter, the position refers to the radius, and the default is the meter scale radius (see <code>AngularMeter.setMeter</code>). For a linear meter, the position refers to the x or y coordinate depending on whether the meter is vertical or horizontal, and the default is the leftX or topY coordinate of the meter scale (see <code>LinearMeter.setMeter</code>).
startWidth	-&h7ffffff	The width at the starting point of the color scale. A positive value means the width is in the direction of increasing "position", while a negative value means decreasing "position". Please refer to the description of <code>startPos</code> on the meaning of "position" for various types of meters. The default is the length of the major tick (see <code>BaseMeter.setTickLength</code>).
endPos	-&h7ffffff	The position of the ending point of the color scale. Please refer to the description of <code>startPos</code> above on how the position parameter is interpreted. The default is for the ending position to be equal to the starting position.
endWidth	-&h7ffffff	The width at the ending point of the color scale. Please refer to the description of <code>startWidth</code> above on how the width parameter is interpreted. The default is for the ending width to be equal to the starting width.
edgeColor	-1	The edge color of the color scale. The default is to have no edge.

See also:

- 3.9.3 `addColorScale(colorStops()` as Double, `startPos` as Integer = -2147483647, `startWidth` as Integer = -2147483647, `endPos` as Integer = -2147483647, `endWidth` as Integer = -2147483647, `edgeColor` as Integer = -1) 279

3.9.5 `addLabel(v as Double, label as string)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a label to a specific position on the meter scale.

Notes: By default, all labels will be associated major ticks on the scale. To associate a label with a minor tick instead, use `'-'` as the first character of the label. To associate a label with a micro tick instead, use `':'` as the first character of the label. To draw the label without any tick at all, use `'` as the first character of the label.

Leading `'-'`, `'`

`textasciitilde` `'` or `':'` characters are tick specification characters and will not appear on the labels. They just specify the the tick style to be associated with the labels. If you want have a label that actually begins these characters, add `'\'` as the first character as the escape character.

In some cases, it may be desirable to skip some labels. If you want to draw a major, minor or micro tick

without any label, use ' ' (a space character), '-' or ':' as the only character in the label text. If you want leave a label position empty without a tick or a label, use an empty string.

Argument	Default	Description
pos	(Mandatory)	The position to add the label to in meter scale unit.
label	(Mandatory)	The label to add to the meter scale.

3.9.6 addPointer(value as Double, fillColor as color, edgeColor as color) as CDMeterPointerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addPointer method, but uses color instead of integer data type for passing color values.

See also:

- 3.9.7 addPointer(value as Double, fillColor as Integer = &hfff0001, edgeColor as Integer = -1) as CDMeterPointerMBS 282

3.9.7 addPointer(value as Double, fillColor as Integer = &hfff0001, edgeColor as Integer = -1) as CDMeterPointerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a pointer to the meter.

Notes:

Argument	Default	Description
value	(Mandatory)	The value that the pointer will point to.
fillColor	LineColor	The fill color of the pointer.
edgeColor	-1	The edge color of the pointer. The default value of -1 means the edge color is the same as the fill color.

See also:

- 3.9.6 addPointer(value as Double, fillColor as color, edgeColor as color) as CDMeterPointerMBS 282

3.9.8 getCoor(v as Double) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the geometric coordinates given the data value.

Notes: For an angular meter, this method returns the angular position of the pointer for the given data

value in degrees. The angle is measure in the clockwise direction, with 0 being the upward pointing direction.

For a linear meter, this method returns the linear pixel offset of the pointer for the given data value.

Argument	Default	Description
v	(Mandatory)	The input data value.

Return Value

The geometric representation of the data value, which is an angle in degrees for an angular meter, and a linear pixel offset for a linear meter.

3.9.9 getLabel(v as Double) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the label at the specified position on the meter scale.

Notes:

Argument	Default	Description
i	(Mandatory)	The position specified as a value on the meter scale.

Return Value

Returns the label at the specified position, or a null string if there is no label at that position.

3.9.10 getTicks as CDArrayMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the positions of all ticks as values on the meter scale.

Notes: Return Value

All array of numbers representing the positions of the ticks as values on the meter scale.

3.9.11 setLabelFormat(mainLabelFormat as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the format of the labels computed by ChartDirector.

Example:

```
dim c as CDBaseMeterMBS
```

// you can use label formats like this:

```
c.setLabelFormat("<*block,halign=left*><*font=timesbi.ttf,size=12,underline=1*>{ label } <*/font*><*br*>US$
{ value } K ( { percent } %)")
```

// we can reduce that to this:

```
c.setLabelFormat("< { label } { value } { percent } %")
```

// and it shows 3 numbers. With | 1 after the variable name, we define the decimals after dot:

```
c.setLabelFormat("< { label } { value | 1 } { percent | 1 } %")
```

// and

```
c.setLabelFormat("< { label } { value | 1., } { percent | 1., } %")
```

// uses dot for thousands and comma for decimal separator.

Notes: The method is mainly used when `BaseMeter.setScale` is used, in which case the label values are not directly specified but are computed by `ChartDirector`. For example, a format string of "`{ value | 2 }`" will format the values with 2 decimal places.

Please refer to [Parameter Substitution and Formatting](#) on all available format parameters.

Argument	Default	Description
<code>formatString</code>	(Mandatory)	The format string.

3.9.12 `setLabelPos(labelInside as boolean, labelOffset as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the positions of the meter scale labels.

Notes:

Argument	Default	Description
<code>labelInside</code>	(Mandatory)	A true value means that the labels are on the 'inward' side of the meter scale. A false value means the labels are on the 'outward' side of the meter scale.

For an angular meter, the default is for the labels on the 'inward' side. For a linear meter, the default is for the labels on the 'outward' side.

labelOffset 0 The offset, in pixels, of the labels position relative to the standard position. If the major tick is at the same side as the label, the standard starting position of the label is at the end point of the major tick. Otherwise, the standard starting position is on the meter scale.

A positive labelOffset means the labels should move more towards the 'inward' or 'outward' side, depending on the first parameter. A negative labelOffset means moving the labels towards the opposite direction.

3.9.13 setLabelStyle(font as string = "", fontsize as Double = -1, fontcolor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the font style used to for the meter labels.

Notes: See Font Specification for details on various font attributes.

Argument	Default	Description
font	"bold"	The font used to draw the labels.
fontSize	-1	The font size used to draw the labels in points. A value of -1 means the font size is not changed. The default font size is 10 points for angular meters, and 8 points for linear meters.
fontColor	TextColor	The color used to draw the labels.
fontAngle	0	The rotation angle of the labels.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.9.14 setLabelStyle(font as string, fontsize as Double, fontcolor as color, fontAngle as Double = 0) as CDTextBoxMBS 285

3.9.14 setLabelStyle(font as string, fontsize as Double, fontcolor as color, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setLabelStyle` method, but uses color instead of integer data type for passing color values.

See also:

- 3.9.13 `setLabelStyle(font as string = "", fontsize as Double = -1, fontcolor as Integer = &hfff0002, fontAngle as Double = 0)` as `CDTextBoxMBS` 285

3.9.15 `setLineWidth(axisWidth as Integer, majorTickWidth as Integer = 1, minorTickWidth as Integer = 1, microTickWidth as Integer = 1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the line widths of the scale line and the ticks on the meter.

Notes:

Argument	Default	Description
<code>axisWidth</code>	(Mandatory)	The line width for drawing the meter scale line in pixels. For an angular meter, the default is 1 pixel. For a linear meter, the default is 2 pixels.
<code>majorTickWidth</code>	1	The line width of the major ticks in pixels. For a linear meter, if this method is not called, the initial major tick width is set to 2 pixels.
<code>minorTickWidth</code>	1	The line width of the minor ticks in pixels.
<code>microTickWidth</code>	1	The line width of the micro ticks in pixels.

3.9.16 `setMeterColors(axisColor as color, labelColor as color, tickColor as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setMeterColors` method, but uses color instead of integer data type for passing color values.

See also:

- 3.9.17 `setMeterColors(axisColor as Integer, labelColor as Integer = -1, tickColor as Integer = -1)` 286

3.9.17 `setMeterColors(axisColor as Integer, labelColor as Integer = -1, tickColor as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors of the meter scale line, scale label and tick.

Notes:

See also:

- 3.9.16 `setMeterColors(axisColor as color, labelColor as color, tickColor as color)` 286

Argument	Default	Description
axisColor	(Mandatory)	The color for drawing the meter scale line. The default is LineColor.
labelColor	-1	The color for drawing the scale labels. Passing -1 in this argument means the label color is not changed. The default label color is TextColor.
tickColor	-1	The color for drawing the ticks. Passing -1 in this argument means the tick color is not changed. The default tick color is LineColor.

3.9.18 setScale(lowerLimit as Double, upperLimit as Double, labels() as Double, formatstring as string = "")

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the meter to use the given scale with the given numeric labels and tick positions.

Notes: ChartDirector will distribute the labels evenly on the scale. By default, all labels are associated with major ticks. If you want certain positions to show a minor or micro ticks only, use MinorTickOnly or MicroTickOnly as the label values for those positions.

Argument	Default	Description
lowerLimit	(Mandatory)	The lower limit of the meter scale.
upperLimit	(Mandatory)	The upper limit of the meter scale.
labels	(Mandatory)	An array of numbers to be used as the labels on the meter scale.
formatString	""	The format string for formatting the numbers. An empty string means the format will be automatically determined.

See also:

- 3.9.19 setScale(lowerLimit as Double, upperLimit as Double, labels() as string) 287
- 3.9.20 setScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0, microTickInc as Double = 0) 288

3.9.19 setScale(lowerLimit as Double, upperLimit as Double, labels() as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the meter to use the given scale with the given text labels and tick positions.

Notes: ChartDirector will distribute the labels evenly on the scale. By default, all labels are associated with major ticks. These can be modified by using '-', 'textasciitilde' or ':' as the first character. Please refer to BaseMeter.addLabel for details.

See also:

- 3.9.18 setScale(lowerLimit as Double, upperLimit as Double, labels() as Double, formatstring as string = "") 287

Argument	Default	Description
lowerLimit	(Mandatory)	The lower limit of the meter scale.
upperLimit	(Mandatory)	The upper limit of the meter scale.
labels	(Mandatory)	An array of text strings to be used as the labels on the meter scale.

- 3.9.20 `setScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0, microTickInc as Double = 0)` 288

3.9.20 `setScale(lowerLimit as Double, upperLimit as Double, majorTickInc as Double = 0, minorTickInc as Double = 0, microTickInc as Double = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the meter to use the given scale.

Notes:

Argument	Default	Description
lowerLimit	(Mandatory)	The lower limit of the meter scale.
upperLimit	(Mandatory)	The upper limit of the meter scale.
majorTickInc	0	The interval between major ticks. For example, a value of 10 means a major tick every 10 units in the meter scale. Each major tick will have an associated label for the value at the tick. A value of 0 disables major ticks.
minorTickInc	0	The interval between minor ticks. For example, a value of 5 means a minor tick every 5 units in the meter scale. A value of 0 disables minor ticks.
microTickInc	0	The interval between minor ticks. For example, a value of 1 means a minor tick every 1 units in the meter scale. A value of 0 disables micro ticks.

See also:

- 3.9.18 `setScale(lowerLimit as Double, upperLimit as Double, labels() as Double, formatstring as string = "")` 287
- 3.9.19 `setScale(lowerLimit as Double, upperLimit as Double, labels() as string)` 287

3.9.21 `setTickLength(majorLen as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Set the lengths of the ticks.

Notes:

For an angular meter, the default is -10 (10 pixels at the inward direction). For a horizontal linear meter, the default is the same height as the meter scale region in the inward direction. For a vertical linear meter, the default is the same width as the meter scale region in the inward direction.

Argument	Default	Description
majorLen	(Mandatory)	The length of the major ticks in pixels. A positive value means the tick is at the 'outward' direction of the meter. A negative value means the tick is at the 'inward' direction.
minorLen	-7fffff	The length of the minor ticks in pixels. The default is 60% of the length of the major ticks.
microLen	-7fffff	The length of the micro ticks in pixels. The default is 50% of the length of the minor ticks.

See also:

- 3.9.22 `setTickLength(majorLen as Integer, minorLen as Integer)` 289
- 3.9.23 `setTickLength(majorLen as Integer, minorLen as Integer, microLen as Integer)` 289

3.9.22 `setTickLength(majorLen as Integer, minorLen as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Set the lengths of the ticks.

Notes:

Argument	Default	Description
majorLen	(Mandatory)	The length of the major ticks in pixels. A positive value means the tick is at the 'outward' direction of the meter. A negative value means the tick is at the 'inward' direction.

For an angular meter, the default is -10 (10 pixels at the inward direction). For a horizontal linear meter, the default is the same height as the meter scale region in the inward direction. For a vertical linear meter, the default is the same width as the meter scale region in the inward direction.

See also:

- 3.9.21 `setTickLength(majorLen as Integer)` 288
- 3.9.23 `setTickLength(majorLen as Integer, minorLen as Integer, microLen as Integer)` 289

3.9.23 `setTickLength(majorLen as Integer, minorLen as Integer, microLen as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

minorLen	-7ffffff	The length of the minor ticks in pixels. The default is 60% of the length of the major ticks.
microLen	-7ffffff	The length of the micro ticks in pixels. The default is 50% of the length of the minor ticks.

Function: Set the lengths of the ticks.

Notes:

Argument	Default	Description
majorLen	(Mandatory)	The length of the major ticks in pixels. A positive value means the tick is at the 'outward' direction of the meter. A negative value means the tick is at the 'inward' direction.

For an angular meter, the default is -10 (10 pixels at the inward direction). For a horizontal linear meter, the default is the same height as the meter scale region in the inward direction. For a vertical linear meter, the default is the same width as the meter scale region in the inward direction.

minorLen	-7ffffff	The length of the minor ticks in pixels. The default is 60% of the length of the major ticks.
microLen	-7ffffff	The length of the micro ticks in pixels. The default is 50% of the length of the minor ticks.

See also:

- 3.9.21 `setTickLength(majorLen as Integer)` 288
- 3.9.22 `setTickLength(majorLen as Integer, minorLen as Integer)` 289

3.10 class CDBoxMBS

3.10.1 class CDBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Box class represents boxes.

Notes: It is used as the base class for more complex classes (such as the TextBox class).

Subclass of the CDDrawObjMBS class.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.10.2 Methods

3.10.3 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.10.4 getHeight as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the height of the box.

Notes: In some usages, the height of a box may be dynamically determined. An example is the height of an LegendBox, which cannot be known until all data are available. In these cases, the height is undefined until the chart or at least the legend box has been laid out (using BaseChart.layout or BaseChart.layout-Legend), or the chart image has been drawn (eg. using BaseChart.makeChart, BaseChart.makeChart2 or BaseChart.makeChart3).

Return Value

The height of the box in pixels.

3.10.5 getImageCoor(OffsetX as Integer = 0, OffsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the image map coordinates of the box as HTML image map attributes.

Notes: The image map coordinates will be in the following format:

```
shape="rect" cords=" [ x1 ] , [ y1 ] , [ x2 ] , [ y2 ] "
```

where (x1, y1) and (x2, y2) are opposite corners of the box. The format is designed so that it can easily be

incorporated into HTML image maps.

This method should be called only after creating the chart image (eg. using `BaseChart.makeChart`, `BaseChart.makeChart2` or `BaseChart.makeChart3`). The image map cannot be determined without creating the chart image first.

Argument	Default	Description
<code>offsetX</code>	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
<code>offsetY</code>	0	An offset to be added to all y coordinates in the image map. See <code>offsetX</code> above for description.

Return Value

A text string representing the coordinates of the box in HTML image map attribute format.

3.10.6 `getLeftX` as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the left x pixel coordinate of the box.

Notes: In some cases, the left x coordinate of a box may be dynamically determined. An example is the left x coordinate of an `CDLegendBoxMBS` with alignment set to Center. To determine the left x coordinate, the size of the box must be known first. For these cases, the left x coordinate is undefined until the legend box or the entire chart has been laid out (using `CDBaseChartMBS.layout` or `CDBaseChartMBS.layoutLegend`), or the chart image has been drawn (eg. using `CDBaseChartMBS.makeChart`).

Arguments:

None

Return Value

The the left x pixel coordinate of the box.

3.10.7 `getTopY` as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the top y pixel coordinate of the box.

Notes: In some cases, the top y coordinate of a box may be dynamically determined. An example is the top y coordinate of an `CDLegendBoxMBS` with alignment set to Center. To determine the top y coordinate, the size of the box must be known first. For these cases, the top y coordinate is undefined until the legend box or the entire chart has been laid out (using `CDBaseChartMBS.layout` or `CDBaseChartMBS.layoutLegend`),

or the chart image has been drawn (eg. using `CDBaseChartMBS.makeChart`).

Arguments:

None

Return Value

The top y pixel coordinate of the box.

3.10.8 getWidth as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the width of the box.

Notes: In some usages, the width of a box may be dynamically determined. An example is the width of an `LegendBox`, which cannot be known until all data are available. In these cases, the width is undefined until the chart or at least the legend box has been laid out (using `BaseChart.layout` or `BaseChart.layoutLegend`), or the chart image has been drawn (eg. using `BaseChart.makeChart`, `BaseChart.makeChart2` or `BaseChart.makeChart3`).

Return Value

The width of the box in pixels.

3.10.9 setBackground(colorvalue as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setBackground` method, but uses color instead of integer data type for passing color values.

Example:

```
// set edge to be 2 pixel wide line
Dim table As CDMLTableMBS
Dim t As CDTextBoxMBS = table.getStyle
Dim x As Integer = c.flatBorder(2)
t.setBackground(&heeeeee, CDXYChartMBS.kLineColor, x)
```

See also:

- 3.10.10 setBackground(colorvalue as Integer, edgeColor as Integer = -1, raisedEffect as Integer = 0)

3.10.10 setBackground(colorvalue as Integer, edgeColor as Integer = -1, raisedEffect as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the background color, border color and 3D border effect of the box.

Notes:

Argument	Default	Description
color	(Mandatory)	The background color of the box.
edgeColor	-1	The border color of the box.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument is also used to support Chart::glassEffect and Chart::softLighting effects.

See also:

- 3.10.9 setBackground(colorvalue as color, edgeColor as color, raisedEffect as Integer = 0) 293

3.10.11 setPos(x as Integer, y as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the coordinates of the top-left corner of the box.

Example:

```
// Create a XYChart object of size 250 x 250 pixels
dim c as new CDXYChartMBS(250, 250)
```

```
// Add a bar chart layer using the given data
dim t as CDTextBoxMBS = c.addBarLayer(data)
t.setPos(0, -10) // move 10 up
```

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of the left of the box.
y	(Mandatory)	The y coordinate of the top of the box.

3.10.12 setRoundedCorners(r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the border style to use rounded corners.

Notes:

Argument	Default	Description
r1	10	The radius of the top-left rounded corner in pixels.
r2	-1	The radius of the top-right rounded corner in pixels. The default value of -1 means it is the same as the radius of the top-left corner.
r3	-1	The radius of the bottom-right rounded corner in pixels. The default value of -1 means it is the same as the radius of the top-left corner.
r4	-1	The radius of the bottom-left rounded corner in pixels. The default value of -1 means it is the same as the radius of the top-left corner.

3.10.13 setSize(w as Integer, h as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the width and height of the box.

Notes:

Argument	Default	Description
w	(Mandatory)	The width of the box in pixels.
h	(Mandatory)	The height of the box in pixels.

3.11 class CDBoxWhiskerLayerMBS

3.11.1 class CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The BoxWhiskerLayer class represents box-whisker layers.

Notes: Subclass of the CDBaseBoxLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.11.2 Methods

3.11.3 setBoxColor(item as Integer, boxColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setBoxColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.11.4 setBoxColor(item as Integer, boxColor as Integer) 296

3.11.4 setBoxColor(item as Integer, boxColor as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color for a single box.

Notes:

Argument	Default	Description
item	(Mandatory)	The index of the box to change color. The first box is 0, while the Nth box is N - 1.
boxColor	(Mandatory)	The color to use to fill the box.

See also:

- 3.11.3 setBoxColor(item as Integer, boxColor as color) 296

3.11.5 setBoxColors(colors() as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

3.11. CLASS CDBOXWHISKERLAYERMBS 297

Function: Same as the other setBoxColors method, but uses color instead of integer data type for passing color values.

See also:

- 3.11.6 setBoxColors(colors() as color, names() as string) 297
- 3.11.7 setBoxColors(colors() as Integer) 297
- 3.11.8 setBoxColors(colors() as Integer, names() as string) 298

3.11.6 setBoxColors(colors() as color, names() as string)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setBoxColors method, but uses color instead of integer data type for passing color values.

See also:

- 3.11.5 setBoxColors(colors() as color) 296
- 3.11.7 setBoxColors(colors() as Integer) 297
- 3.11.8 setBoxColors(colors() as Integer, names() as string) 298

3.11.7 setBoxColors(colors() as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors for all boxes.

Notes: The method will set the BoxWhiskerLayer into multi-color mode. The colors of each box will be obtained from the given colors array. If there are insufficient colors in the array, the remaining boxes will be auto-colored.

If the names argument is given, the names will appear in the legend box if one is configured for the chart.

Argument	Default	Description
colors	(Mandatory)	An array of colors to be used as the color of the boxes.
names	[Empty_Array]	An array of names to be used in the legend box, if one is configured for the chart.

See also:

- 3.11.5 setBoxColors(colors() as color) 296
- 3.11.6 setBoxColors(colors() as color, names() as string) 297
- 3.11.8 setBoxColors(colors() as Integer, names() as string) 298

3.11.8 `setBoxColors(colors() as Integer, names() as string)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors for all boxes.

Notes: The method will set the `BoxWhiskerLayer` into multi-color mode. The colors of each box will be obtained from the given colors array. If there are insufficient colors in the array, the remaining boxes will be auto-colored.

If the names argument is given, the names will appear in the legend box if one is configured for the chart.

Argument	Default	Description
colors	(Mandatory)	An array of colors to be used as the color of the boxes.
names	[Empty_Array]	An array of names to be used in the legend box, if one is configured for the chart.

See also:

- 3.11.5 `setBoxColors(colors() as color)` 296
- 3.11.6 `setBoxColors(colors() as color, names() as string)` 297
- 3.11.7 `setBoxColors(colors() as Integer)` 297

3.11.9 `setWhiskerBrightness(whiskerBrightness as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the whisker colors as darken or brightened fill colors of the boxes for `BoxWhiskerLayer` in multi-color mode.

Notes: In multi-color mode, the boxes of a `BoxWhiskerLayer` can have different fill colors. Instead of specifying a single whisker color for all boxes, `ChartDirector` supports deriving the whisker colors by darkening or brightening the corresponding fill colors.

Argument	Default	Description
whiskerBrightness	(Mandatory)	The brightness factor. A value less than 1 means darkening. A value larger than 1 means brightening. A zero value means black.

3.12 class CDCandleStickLayerMBS

3.12.1 class CDCandleStickLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The CandleStickLayer class represents candlestick layers.

Notes: Subclass of the CDBaseBoxLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Xojo Developer Magazine

- [18.5, page 64: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.12.2 Methods

3.12.3 setColors(upFillColor as color, upLineColor as color, downFillColor as color, downLineColor as color)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the colors for candlesticks.

Notes: The standard way to color a candlestick is to color it based on whether it is on an "up" or "down" trading session, in which "up" is defined as the closing price higher than or equal to the opening price, otherwise it is "down".

Argument	Default	Description
upFillColor	(Mandatory)	The fill color for "up" trading sessions.
upLineColor	(Mandatory)	The line color for "up" trading sessions.
downFillColor	(Mandatory)	The fill color for "down" trading sessions.
downLineColor	(Mandatory)	The line color for "down" trading sessions.

See also:

- [3.12.4 setColors\(upFillColor as Integer, upLineColor as Integer, downFillColor as Integer, downLineColor as Integer\)](#) 299

3.12.4 setColors(upFillColor as Integer, upLineColor as Integer, downFillColor as Integer, downLineColor as Integer)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures the colors for candlesticks.

Notes: The standard way to color a candlestick is to color it based on whether it is on an "up" or "down" trading session, in which "up" is defined as the closing price higher than or equal to the opening price, otherwise it is "down".

Argument	Default	Description
upFillColor	(Mandatory)	The fill color for "up" trading sessions.
upLineColor	(Mandatory)	The line color for "up" trading sessions.
downFillColor	(Mandatory)	The fill color for "down" trading sessions.
downLineColor	(Mandatory)	The line color for "down" trading sessions.

See also:

- 3.12.3 `setColors(upFillColor as color, upLineColor as color, downFillColor as color, downLineColor as color)` 299

3.12.5 `setExtraColors(upDownFillColor as color, upDownLineColor as color, downDownFillColor as color, downDownLineColor as color, leadValue as Double = -1.7E308)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures additional colors for candlesticks.

Notes: The standard way to color a candlestick is to color it using the method as mentioned in `CandleStickLayer.setColors`, in which "up" is defined as the closing price higher than or equal to the opening price.

In some applications, it may be desirable to vary the candlestick colors based on an alternative definition of "up" and "down", in which "up" is defined as the closing price higher than or equal to that of the previous trading session.

With these two definitions of "up" and "down", a candlestick can have 4 states. It can be "up/up" ("up" as according to both the standard and alternative definitions), "up/down" ("up" according to the standard definition, "down" according to the alternative definition), "down/up" or "down/down".

The `setExtraColors` method can be used to configure the "up/down" and "down/down" colors. If this method is used, the colors configured with `CandleStickLayer.setColors` will be considered as the "up/up" and "down/up" colors.

See also:

- 3.12.6 `setExtraColors(upDownFillColor as Integer, upDownLineColor as Integer, downDownFillColor as Integer, downDownLineColor as Integer, leadValue as Double = -1.7E308)` 301

Argument	Default	Description
upDownFillColor	(Mandatory)	The fill color for "up/down" days.
upDownLineColor	(Mandatory)	The line color for "up/down" days.
downDownFillColor	(Mandatory)	The fill color for "down/down" days.
downDownLineColor	(Mandatory)	The line color for "down/down" days.
leadValue	[-Infinity]	The closing price before the first trading session, which is used to determine if the first trading session is "up" or "down" according to the alternative definition.

3.12.6 setExtraColors(upDownFillColor as Integer, upDownLineColor as Integer, downDownFillColor as Integer, downDownLineColor as Integer, leadValue as Double = -1.7E308)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures additional colors for candlesticks.

Notes: The standard way to color a candlestick is to color it using the method as mentioned in `CandleStickLayer.setColors`, in which "up" is defined as the closing price higher than or equal to the opening price.

In some applications, it may be desirable to vary the candlestick colors based on an alternative definition of "up" and "down", in which "up" is defined as the closing price higher than or equal to that of the previous trading session.

With these two definitions of "up" and "down", a candlestick can have 4 states. It can be "up/up" ("up" as according to both the standard and alternative definitions), "up/down" ("up" according to the standard definition, "down" according to the alternative definition), "down/up" or "down/down".

The `setExtraColors` method can be used to configure the "up/down" and "down/down" colors. If this method is used, the colors configured with `CandleStickLayer.setColors` will be considered as the "up/up" and "down/up" colors.

Argument	Default	Description
upDownFillColor	(Mandatory)	The fill color for "up/down" days.
upDownLineColor	(Mandatory)	The line color for "up/down" days.
downDownFillColor	(Mandatory)	The fill color for "down/down" days.
downDownLineColor	(Mandatory)	The line color for "down/down" days.
leadValue	[-Infinity]	The closing price before the first trading session, which is used to determine if the first trading session is "up" or "down" according to the alternative definition.

See also:

- 3.12.5 `setExtraColors(upDownFillColor as color, upDownLineColor as color, downDownFillColor as color, downDownLineColor as color, leadValue as Double = -1.7E308)` 300

3.13 class CDCColorAxisMBS

3.13.1 class CDCColorAxisMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The ColorAxis class represents color axes.

Notes: It is a subclass of Axis.

A color axis is similar to an x or y axis. However, instead of mapping data values to positions, a color axis maps data values to colors. Color axes are used in CDContourLayerMBS and in CDSurfaceChartMBS objects.

Similar to an x or y axis, a color axis is visually represented as a thin bar that can be horizontal or vertical, and has a scale along its length. The scale can be set explicitly or can be determined by auto-scaling.

If you are using the color axis on a surface chart, please note that by default, the color axis scale is synchronized to the z axis. You need to use CDAxisMBS.syncAxis to cancel the synchronization before you can independently set the color axis scale.

In addition to the scale, a color axis is colored with a continuous color gradient or discrete color steps along its length. This maps the values on the axis with colors.

For an x or y axis, the major and minor ticks correspond to major and minor grid lines on the chart. For a color axis, these ticks correspond to major and minor contour lines on the chart.

Like an x or y axis, you can add marks to the color axis using Axis.addMark. They will become mark contour lines on the chart.

The color axis, if visible, acts as a color legend. The axis stem is a thin rectangle 15 pixels in width, configurable with CDAxisMBS.setWidth. A bounding box can be added to surround the color axis using CDCColorAxisMBS.setBoundingBox.

A color axis works normally (determines color mapping and contour levels) even if it is not displayed.

Subclass of the CDAxisMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Xojo Plugins, version 21.3pr1](#)

Xojo Developer Magazine

- [7.5, page 34: Easy Charts and Graphs Part 2, Using the ChartDirector Plugin](#)
- [7.4, page 33: Easy Charts and Graphs, Using the ChartDirector Plugin](#)

3.13.2 Methods

3.13.3 `getBoxHeight` as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the height of the color axis inclusive of the bounding box.

Notes: This method should be called only after axis layout (after calling `CDXYChartMBS.layoutAxes`, `CDBaseChartMBS.layout` or `CDXYChartMBS.packPlotArea`).

3.13.4 `getBoxWidth` as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the width of the color axis inclusive of the bounding box.

Notes: This method should be called only after axis layout (after calling `CDXYChartMBS.layoutAxes`, `CDBaseChartMBS.layout` or `CDXYChartMBS.packPlotArea`).

3.13.5 `getColor(z as Double)` as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the color given the data value.

Notes: You must call `CDXYChartMBS.layoutAxes`, `CDBaseChartMBS.layout` or `CDXYChartMBS.packPlotArea` first before calling this method. `ChartDirector` needs to perform auto-scaling and layout the axis before it can compute the color.

Arguments:

Argument	Default	Description
<code>z</code>	(Mandatory)	The data value.

Return Value

The color that corresponds to the data value.

3.13.6 getColorScale as Double()

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the color stops defining the axis scale and the associated colors.

Notes: Returns an array of numbers representing the color stops.

Please refer to ColorAxis.setColorScale on how to interpret this array.

3.13.7 setAxisBorder(edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setAxisBorder method, but uses color instead of integer data type for passing color values.

Notes:

Argument	Default	Description
edgeColor	(Mandatory)	The edge color of the axis stem.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument can also be used to specify glassEffect, softLighting, :cylinderEffect or flatBorder effects.

See also:

- 3.13.8 setAxisBorder(edgeColor as Integer, raisedEffect as Integer = 0) 304

3.13.8 setAxisBorder(edgeColor as Integer, raisedEffect as Integer = 0)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the border color and 3D border effect of the axis stem.

Notes:

Argument	Default	Description
edgeColor	(Mandatory)	The edge color of the axis stem.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument can also be used to specify glassEffect, softLighting, :cylinderEffect or flatBorder effects.

See also:

- 3.13.7 setAxisBorder(edgeColor as color, raisedEffect as Integer = 0) 304

3.13.9 setAxisPos(x as Integer, y as Integer, alignment as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position and alignment of the color axis.

3.13.10 setBoundingBox(fillColor as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setBoundingBox method, but uses color instead of integer data type for passing color values.

See also:

- 3.13.11 setBoundingBox(fillColor as Integer, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0) 305

3.13.11 setBoundingBox(fillColor as Integer, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the background color, border color and 3D border effect of the bounding box.

Notes: Arguments:

Argument	Default	Description
fillColor	(Mandatory)	The background color of the bounding box.
edgeColor	kTransparent	The border color of the bounding box.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument is also used to support CDBaseChartMBS.glassEffect, CDBaseChartMBS.softLighting and CDBaseChartMBS.cylinderEffect effects.

See also:

- 3.13.10 setBoundingBox(fillColor as color, edgeColor as color, raisedEffect as Integer = 0) 305

3.13.12 setBoxMargin(leftMargin as Integer, rightMargin as Integer, topMargin as Integer, bottomMargin as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the margins of the bounding box in pixels.

Notes: The margins of a bounding box refer to the distances between the borders of the bounding box to the color axis inside.

Arguments:

Argument	Default	Description
leftMargin	(Mandatory)	The left margin in pixels.
rightMargin	(Mandatory)	The right margin in pixels.
topMargin	(Mandatory)	The top margin in pixels.
bottomMargin	(Mandatory)	The bottom margin in pixels.

See also:

- 3.13.13 `setBoxMargin(m as Integer)` 306

3.13.13 `setBoxMargin(m as Integer)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets all margins (left, right, top, and bottom) of the bounding box to the same value.

Notes: The margins of a bounding box refer to the distances between the borders of the bounding box to the color axis inside.

Arguments:

Argument	Default	Description
m	(Mandatory)	The left, right, top and bottom margins in pixels.

See also:

- 3.13.12 `setBoxMargin(leftMargin as Integer, rightMargin as Integer, topMargin as Integer, bottomMargin as Integer)` 305

3.13.14 `setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as color, overflowColor as color)`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the continuous color gradient or discrete color steps for the color axis.

Notes: Arguments:

Argument	Default	Description
isContinuous	true	True to set to continuous color gradient. False to set to discrete color steps.
colors	[Empty_Array]	An array of colors used to define the continuous color gradient. If an empty array is used, the colors are automatically determined.

ChartDirector will interpolate between the colors to create the continuous gradient. If discrete color steps are used, ChartDirector will take samples along the continuous color gradient. The number of samples is determined by the number of ticks on the axis, and may not equal the number of colors in the colors array.

overflowColor	-1	The color to use if a value exceeds the maximum value of the axis scale. -1 means the overflowColor is the same color at the maximum value.
underflowColor	-1	The color to use if a value falls below the minimum value of the axis scale. -1 means the underflowColor is the same color at the minimum value.

See also:

- 3.13.15 setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as Integer, overflowColor as Integer = -1) 307
- 3.13.16 setColorGradient(isContinuous as boolean, Colors() as Integer, underflowColor as Integer = -1, overflowColor as Integer = -1) 308
- 3.13.17 setColorGradient(isContinuous as boolean=true) 309

3.13.15 setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as Integer, overflowColor as Integer = -1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the continuous color gradient or discrete color steps for the color axis.

Notes: Arguments:

Argument	Default	Description
isContinuous	true	True to set to continuous color gradient. False to set to discrete color steps.
colors	[Empty_Array]	An array of colors used to define the continuous color gradient. If an empty array is used, the colors are automatically determined.

ChartDirector will interpolate between the colors to create the continuous gradient. If discrete color steps are used, ChartDirector will take samples along the continuous color gradient. The number of samples is determined by the number of ticks on the axis, and may not equal the number of colors in the colors array.

See also:

overflowColor	-1	The color to use if a value exceeds the maximum value of the axis scale. -1 means the overflowColor is the same color at the maximum value.
underflowColor	-1	The color to use if a value falls below the minimum value of the axis scale. -1 means the underflowColor is the same color at the minimum value.

- 3.13.14 setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as color, overflowColor as color) 306
- 3.13.16 setColorGradient(isContinuous as boolean, Colors() as Integer, underflowColor as Integer = -1, overflowColor as Integer = -1) 308
- 3.13.17 setColorGradient(isContinuous as boolean=true) 309

3.13.16 setColorGradient(isContinuous as boolean, Colors() as Integer, underflowColor as Integer = -1, overflowColor as Integer = -1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the continuous color gradient or discrete color steps for the color axis.

Notes: Arguments:

Argument	Default	Description
isContinuous	true	True to set to continuous color gradient. False to set to discrete color steps.
colors	[Empty_Array]	An array of colors used to define the continuous color gradient. If an empty array is used, the colors are automatically determined.

ChartDirector will interpolate between the colors to create the continuous gradient. If discrete color steps are used, ChartDirector will take samples along the continuous color gradient. The number of samples is determined by the number of ticks on the axis, and may not equal the number of colors in the colors array.

overflowColor	-1	The color to use if a value exceeds the maximum value of the axis scale. -1 means the overflowColor is the same color at the maximum value.
underflowColor	-1	The color to use if a value falls below the minimum value of the axis scale. -1 means the underflowColor is the same color at the minimum value.

See also:

- 3.13.14 setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as color, overflowColor as color) 306
- 3.13.15 setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as Integer, overflowColor as Integer = -1) 307
- 3.13.17 setColorGradient(isContinuous as boolean=true) 309

3.13.17 setColorGradient(isContinuous as boolean=true)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the continuous color gradient or discrete color steps for the color axis.

Notes: Arguments:

Argument	Default	Description
isContinuous	true	True to set to continuous color gradient. False to set to discrete color steps.
colors	[Empty_Array]	An array of colors used to define the continuous color gradient. If an empty array is used, the colors are automatically determined.

ChartDirector will interpolate between the colors to create the continuous gradient. If discrete color steps are used, ChartDirector will take samples along the continuous color gradient. The number of samples is determined by the number of ticks on the axis, and may not equal the number of colors in the colors array.

overflowColor	-1	The color to use if a value exceeds the maximum value of the axis scale. -1 means the overflowColor is the same color at the maximum value.
underflowColor	-1	The color to use if a value falls below the minimum value of the axis scale. -1 means the underflowColor is the same color at the minimum value.

See also:

- 3.13.14 setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as color, overflowColor as color) 306
- 3.13.15 setColorGradient(isContinuous as boolean, Colors() as color, underflowColor as Integer, overflowColor as Integer = -1) 307
- 3.13.16 setColorGradient(isContinuous as boolean, Colors() as Integer, underflowColor as Integer = -1, overflowColor as Integer = -1) 308

3.13.18 setColorScale(colorStops() as Double, underflowColor as Integer = -1, overflowColor as Integer = -1)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis scale and the associated colors.

Notes: The axis scale and the associated colors are defined with an array of numbers. For a color scale with continuous color gradient, each pair of numbers represents a value and its associated color. For example, for a continuous color scale in which 0 is blue (0000FF), 50 is yellow (FFFF00) and 100 is red (FF0000), the numbers should be:

0.0, &h0000ff, 50.0, &hffff00, 100.0, &hff0000

For a color scale with discrete color steps, the number of colors would be one less than the number of values. For example, to define a step color scale in which 0 to 50 is red (FF0000), and 50 to 100 is green (00FF00), the numbers will be:

0.0, &hff000, 50.0, &h00ff00, 100.0

Argument	Default	Description
colorStops	(Mandatory)	An array of numbers alternating between values and colors. If the number count is even, the array will define a continuous color scale, otherwise it will define a step color scale.
underflowColor	-1	The color to use if a value falls below the minimum value of the axis scale. -1 means the underflowColor is the same color at the minimum value.
overflowColor	-1	The color to use if a value exceeds the maximum value of the axis scale. -1 means the overflowColor is the same color at the maximum value.

3.13.19 setCompactAxis(b as boolean=true)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets whether to compact the axis or not.

Notes: For a color axis, the axis can assume any length. Typically, an initial length is specified so that ChartDirector can determine how many labels can be fitted on the axis, and auto-scale the axis accordingly.

After the labels are determined, ChartDirector can compact the axis by setting its length to the minimum length compatible with the major tick density (see CDAxisMBS.setTickDensity).

For example, suppose ChartDirector auto-scaling has set the axis labels to be [0, 1, 2, 3, 4, 5], and the major tick density is 20 pixels, ChartDirector will set the axis length to 100 pixels, irrespective of its original length.

If this method is never called, the default is not to compact the axis.

Arguments:

Argument	Default	Description
b	true	A true value means to compact the axis. A false value means not to compact the axis.

3.13.20 setLevels(maxLevels as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the maximum number of contour intervals on the axis.

Notes: If discrete color steps are used, the number of contour intervals is the number of color steps on the axis.

For a general color axis, the number of contour intervals is equal to the number of ticks minus 1. For example, an axis with 6 ticks at [0, 2, 4, 6, 8, 10] has 6 contour lines and 5 contour intervals.

The main usage of this method is to limit the number of contour lines and/or color steps on the contour layer.

If this method is never called, the default maximum number of contour intervals is 16.

Arguments:

Argument	Default	Description
maxLevels	(Mandatory)	An integer representing the maximum number of contour intervals.

3.13.21 setRoundedCorners(r1 as Integer = 10, r2 as Integer = -1, r3 as Integer = -1, r4 as Integer = -1)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the border style of the bounding box to rounded corners.

Notes: Arguments:

Argument	Default	Description
r1	10	The radius of the top-left rounded corner in pixels.
r2	-1	The radius of the top-right rounded corner in pixels. -1 means it is the same as the radius of the top-left corner.
r3	-1	The radius of the bottom-right rounded corner in pixels. -1 means it is the same as the radius of the top-left corner.
r4	-1	The radius of the bottom-left rounded corner in pixels. -1 means it is the same as the radius of the top-left corner.

3.14 class CDContourLayerMBS

3.14.1 class CDContourLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The ContourLayer class represents contour layers

Notes: CDContourLayerMBS is a subclass of CDLayerMBS.

CDContourLayerMBS objects are created by using CDXYChartMBS.addContourLayer.

The data for the CDContourLayerMBS are the z values at some points in the xy plane. Both gridded and scattered data points are supported.

Using the data, the ContourLayer computes the z values of all pixels in the xy plane. The computation can be based on spline or bilinear/line surface fitting, configurable using CDContourLayerMBS.setSmoothInterpolation.

The ContourLayer then colors the pixels based on the z values, and draws contour lines to join positions with the same z values. The color mapping and contour levels are controlled by the ColorAxis object of the ContourLayer, obtainable using CDContourLayerMBS.colorAxis.

If you need to change colors for the contour layer, please use ColorAxis.setColorGradient:

Call `myContourLayer.colorAxis().setColorGradient(True, arrayOfColors)`

ChartDirector will then use the array of colors, and interpolate among them, to fill the contour layer. For example, if the array of colors contain 3 colors red, green and blue, ChartDirector will interpolate from red to green for half of the z-levels, and green to blue for the other half.

Subclass of the CDLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Xojo Developer Magazine

- [7.4, page 33: Easy Charts and Graphs, Using the ChartDirector Plugin](#)

3.14.2 Methods

3.14.3 addCustomContour(z as double, contourColor as Integer, contourWidth as Integer, contourLabel as string, font as string = "normal", fontsize as double = 12, fontColor as integer = &hfff0002) as CDMarkMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a custom contour to the layer.

Notes: You can also add a custom contour by using `Axis.addMark` on the `ColorAxis`, in which case a label will also be added to the color axis. The `addCustomContour` method adds the label to the contour without affecting the color axis.

Argument	Default	Description
<code>z</code>	(Mandatory)	The <code>z</code> value of the contour.
<code>contourColor</code>	(Mandatory)	The contour line color.
<code>contourWidth</code>	(Mandatory)	The contour line width in pixels.
<code>contourLabel</code>	(Mandatory)	The text of the contour label.
<code>font</code>	"normal"	The font used to draw the label.
<code>fontSize</code>	8	The font size used to draw the label.
<code>fontColor</code>	<code>TextColor</code>	The color used to draw the labels.

3.14.4 colorAxis as CDCColorAxisMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the `ColorAxis` object representing the color axis (color legend).

3.14.5 getCrossSection(x0 as Integer, y0 as Integer, x1 as Integer, y1 as Integer) as Double()

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Get the cross section between two points on a contour layer.

Notes: Note: This method should be used only after `ChartDirector` has output the chart image, or after `BaseChart.layout` has been called.

This method will return an array of numbers representing the `z` values of evenly spaced samples along a straight line joining the two end-points. The end-points are specified in pixel coordinates. If the input are data values, they can be converted to pixel coordinates using `XYChart.getXCoor` and `XYChart.getYCoor`.

Return Value

Argument	Default	Description
x0	(Mandatory)	The x pixel coordinate of the first end-point.
y0	(Mandatory)	The y pixel coordinate of the first end-point.
x1	(Mandatory)	The x pixel coordinate of the second end-point.
y1	(Mandatory)	The y pixel coordinate of the second end-point.

An array of numbers representing the z values of evenly spaced samples along a straight line joining the two end-points. The array can contain NoValue if.

3.14.6 getZAtPixel(x as Integer, y as Integer) as Double

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the z value at the specify (x, y) pixel coordinate.

Notes: This method should be used only after ChartDirector has output the chart image, or after BaseChart.layout has been called.

Argument	Default	Description
x	(Mandatory)	The x pixel coordinate.
y	(Mandatory)	The y pixel coordinate.

Return Value

The z value at the specified (x, y) pixel coordinate, or NoValue if the point lie outside the contour region.

3.14.7 getZAtValue(x as double, y as double) as Double

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the z value at the specify (x, y) data values.

Notes: This method should be used only after ChartDirector has output the chart image, or after BaseChart.layout has been called.

Argument	Default	Description
x	(Mandatory)	The x data value.
y	(Mandatory)	The y data value.

Return Value

The z value at the specified (x, y) data coordinate, or NoValue if the point lie outside the contour region.

3.14.8 setColorAxis(x as Integer, y as Integer, alignment as Integer, length as Integer, orientation as Integer) as CDCColorAxisMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position, length and orientation of the color axis (color legend).

Notes: Arguments:

Argument	Default	Description
x	(Mandatory)	The x-coordinate of the reference point used to position the color axis.
y	(Mandatory)	The y-coordinate of the reference point used to position the color axis.
alignment	(Mandatory)	The alignment of the color axis with respect to the reference point. For example, a value of kTopLeft means the reference point is the top-left corner of the color axis. See Alignment Specification for supported alignment types.
length	(Mandatory)	The length of the color axis in pixels.
orientation	(Mandatory)	The orientation of the color axis. A value of kTop/kBottom means the axis is horizontal, and the axis labels are at top/bottom side of the axis. A value of kLeft/kRight means the axis is vertical, and the axis labels are at the left/right side of the axis.

3.14.9 setContourColor(contourColor as color, minorContourColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setContourColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.14.10 setContourColor(contourColor as Integer, minorContourColor as Integer = -1) 315

3.14.10 setContourColor(contourColor as Integer, minorContourColor as Integer = -1)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets contour line colors.

Notes:

Argument	Default	Description
contourColor	(Mandatory)	The major contour line color. If this method is never called, the default contour line color is semi-transparent black (�).
minorContourColor	-1	The minor contour line color. -1 means the minor contour line color is the same as the major contour line color.

See also:

- 3.14.9 setContourColor(contourColor as color, minorContourColor as color)

315

3.14.11 setContourLabelFormat(formatString as string)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data label format.

Notes: By default, the data label format is " { label } ". Please refer to Parameter Substitution and Formatting on available parameters and how to format them.

Argument	Default	Description
formatString	(Mandatory)	The format string.

3.14.12 setContourLabelSpacing(labelSpacing as Integer, minContourLen as Integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the spacing between contour labels.

Notes: ChartDirector will try to place labels at relative straight segments along the contour while ensuring the labels are not too close together. The default target label spacing is 300 pixels.

If the contour is short relative to the label length, all or a large part of the contour may become the label, and the contour line may become hard to see. By default, ChartDirector will not put a label on a contour if the contour length is less than 3 times that of the label length. This allows the contour on either side of the label to be as least as long as the label.

This method can be used to adjust what is the target label spacing and contour length for positioning labels.

Argument	Default	Description
labelSpacing	(Mandatory)	The target label spacing in pixels. The actual spacing may be different as the labels can only be placed on relatively straight segments along the contour.
minContourLen	(Mandatory)	The minimum contour length in pixels below which there will be no contour label.

3.14.13 setContourLabelStyle(font as string = "normal", fontsize as double = 12, fontColor as integer = &hfff0002) as CDTextBoxMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables contour labels and sets their styles.

Notes:

Argument	Default	Description
font	"normal"	The font used to draw the labels.
fontSize	8	The font size used to draw the labels.
fontColor	TextColor	The color used to draw the labels.

Returns a TextBox object representing the prototype of the contour labels. This may be used to fine-tune the appearance of the contour labels.

3.14.14 setContourWidth(contourWidth as Integer, minorContourWidth as Integer = -1)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets contour line widths.

Notes: Arguments:

Argument	Default	Description
contourWidth	(Mandatory)	The major contour line width in pixels. If this method is never called, the default contour line width is 1 pixel.
minorContourWidth	-1	The minor contour line width. -1 means the minor contour line width is the same as the major contour line width.

3.14.15 setExactContour(contour as boolean = true)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets whether to use exact contours or boundary contours.

Notes: In exact contour mode, the contour at $z = N$ refers to all points at $z = N$. For example, the contour at $z = 10$ refers to all points at $z = 10$. These points normally constitute lines. However, in case there is an exactly flat region at $z = 10$, the entire region will be the contour and will be colored using the contour color. In other words, instead of lines, a contour can become a region.

In boundary contour mode, the contour at $z = N$ refers to the boundary between $z < N$ and $z \geq N$, except for the highest contour in which case the contour is the boundary between $z \leq N$ and $z > N$. With this method, if there is a flat region, the contour will occur at the boundary of the region.

If this method is never called, the default is to use boundary contour mode.

Argument	Default	Description
contour	true	A true value specifies exact contour mode for standard contours, otherwise boundary contour mode will be used.
markContour	[Same_As_Contour]	A true value specifies exact contour mode for mark contours. A false value specifies boundary contour mode for mark contours. A mark contour is an extra contour added using <code>Axis.addMark</code> on the color axis (obtained using <code>ContourLayer.colorAxis</code>). If this parameter is not specified, the default is to use the same contour mode as standard contours.

See also:

- 3.14.16 `setExactContour(contour as boolean, markContour as boolean)` 318

3.14.16 `setExactContour(contour as boolean, markContour as boolean)`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets whether to use exact contours or boundary contours.

Notes: In exact contour mode, the contour at $z = N$ refers to all points at $z = N$. For example, the contour at $z = 10$ refers to all points at $z = 10$. These points normally constitute lines. However, in case there is an exactly flat region at $z = 10$, the entire region will be the contour and will be colored using the contour color. In other words, instead of lines, a contour can become a region.

In boundary contour mode, the contour at $z = N$ refers to the boundary between $z < N$ and $z \geq N$, except for the highest contour in which case the contour is the boundary between $z \leq N$ and $z > N$. With this method, if there is a flat region, the contour will occur at the boundary of the region.

If this method is never called, the default is to use boundary contour mode.

Argument	Default	Description
contour	true	A true value specifies exact contour mode for standard contours, otherwise boundary contour mode will be used.
markContour	[Same_As_Contour]	A true value specifies exact contour mode for mark contours. A false value specifies boundary contour mode for mark contours. A mark contour is an extra contour added using <code>Axis.addMark</code> on the color axis (obtained using <code>ContourLayer.colorAxis</code>). If this parameter is not specified, the default is to use the same contour mode as standard contours.

See also:

- 3.14.15 `setExactContour(contour as boolean = true)` 317

3.14.17 setSmoothInterpolation(b as boolean)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets whether to use spline or bilinear/linear surface interpolation.

Notes: The data points for the contour layer are samples of the z values at certain gridded or scattered (x, y) points. However, to draw the contour and to color the layer, it is necessarily to know the z values at all pixels in the xy plane. ChartDirector uses surface interpolation to compute the z values at all pixels from the given data points.

In spline surface interpolation, ChartDirector will compute a smooth surface that passes through all data points. In bilinear/linear surface interpolation, ChartDirector will use the data points to partition the (x, y) plane into rectangular or triangular regions, and will fit bilinear or linear surfaces to the regions.

Spline surface interpolation generally produces smoother contours and is the default.

Arguments:

Argument	Default	Description
b	(Mandatory)	A true value means to use spline surface interpolation. A false value means to use bilinear/linear surface interpolation.

3.14.18 setZBounds(minZ as Double, maxZ as Double)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clips the z values to the given bounds.

Notes: In spline surface interpolation (see `ContourLayer.setSmoothInterpolation`), it is possible that the interpolated surface contains regions that are higher than the highest data point, or lower than the lowest data point.

As an example, consider a linear sequence of points with z values [10, 20, 30, 40, 40, 30, 20, 10]. The first 4 points represent an upward slope, while the last 4 points represent a downward slope. If these points are to be joined with a smooth spline, the peak will naturally fall somewhere between the middle two points and exceed 40. So there will be a region in the spline higher than the highest data point.

In some applications, it may be desirable to limit the z values to certain bounds. For example, if the data are about rainfall, it is not desirable if the interpolated surface falls below 0, as there cannot be negative rainfall. The `setZBounds` method can be used to clip the interpolated surface to the desired bounds.

Argument	Default	Description
minZ	DataBound	The minimum z value to clip to. Use NoValue if there is no minimum z bound. The special constant DataBound means that the minimum z value is the minimum data value.
maxZ	DataBound	The maximum z value to clip to. Use NoValue if there is no maximum z bound. The special constant DataBound means that the maximum z value is the maximum data value.

3.14.19 setZData(zData() as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the z data used for the contour chart.

Notes: Arguments:

Argument	Default	Description
zData	(Mandatory)	An array of numbers as the z data for the contour chart.

3.15 class CDDataAcceleratorMBS

3.15.1 class CDDataAcceleratorMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The DataAccelerator class is for speeding up processing of huge data series.

Notes: With the DataAccelerator, it is possible to plot a chart with 1 billion data points with programmable track cursor support. The chart can smoothly zoom out to display all data points and zoom in 100,000,000X to see individual data points. At the same time, the chart can update in real time with thousands of data points per second.

Even if DataAccelerator is not used, ChartDirector can often achieve adequate performance for up to a few million data points depending on chart complexity. That means you do not need to use DataAccelerator if you have less than 100000 data points.

Each DataAccelerator can support one x data series shared by multiple y data series. Multi-threading can be used to process the series in parallel. You can use multiple DataAccelerator objects if you have multiple x data series.

The DataAccelerator needs around 12 bytes of RAM per data value. Since the application must store the data in RAM before passing it to DataAccelerator, it needs at least 8 bytes per double precision number. That means the application would consume around 20 bytes per value, or 20G of RAM for 1 billion values.

In the current version of ChartDirector, only XY chart line layers can support DataAccelerator with the CDXYChartMBS.addLineLayer3 method. Area charts can be created from line layers with CDXYChartMBS.addInterLineLayer.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.2](#)
- [MBS Xojo Plugins, version 21.2pr5](#)
- [ChartDirector 7 update](#)

3.15.2 Methods

3.15.3 addDataSeries(id as string, yData as CDArrayMBS)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds a data series.

Notes: The yData is an array for storing the y data values, and yDataLen is the number of valid values in the array.

To conserve memory, `DataAccelerator` will only keep a reference to the array and not copy it. The caller must ensure the array remains valid for as long as the `DataAccelerator` is in use.

For real time chart usage, the array acts as a buffer to store real time data. Initially the number of valid values can be 0. As new data come in, the values can be stored in the array and `CDDataAcceleratorMBS.extendLength` can be called to update the number of valid values.

This method is specially designed to allow multiple data series to be updated concurrently with multiple threads.

Argument	Default	Description
<code>id</code>	(Mandatory)	Name of the data series. Each data series in a <code>CDDataAcceleratorMBS</code> object must have a unique name.
<code>yData</code>	(Mandatory)	The array to store the data values.

See also:

- 3.15.4 `addDataSeries(id as string, yData() as double)` 322

3.15.4 `addDataSeries(id as string, yData() as double)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds a data series.

Notes: The `yData` is an array for storing the y data values, and `yDataLen` is the number of valid values in the array.

To conserve memory, `DataAccelerator` will only keep a reference to the array and not copy it. The caller must ensure the array remains valid for as long as the `DataAccelerator` is in use.

For real time chart usage, the array acts as a buffer to store real time data. Initially the number of valid values can be 0. As new data come in, the values can be stored in the array and `CDDataAcceleratorMBS.extendLength` can be called to update the number of valid values.

This method is specially designed to allow multiple data series to be updated concurrently with multiple threads.

See also:

- 3.15.3 `addDataSeries(id as string, yData as CDDArrayMBS)` 321

Argument	Default	Description
id	(Mandatory)	Name of the data series. Each data series in a CDDataAcceleratorMBS object must have a unique name.
yData	(Mandatory)	The array to store the data values.

3.15.5 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Creates a DataAccelerator object.

Notes: The xData is an array for storing the x data values, and xDataLen is number of valid values in the array.

To conserve memory, DataAccelerator will only keep a reference to the array and not copy it. The caller must ensure the array remains valid for as long as the DataAccelerator is in use.

For real time chart usage, the array acts as a buffer to store real time data. Initially the number of valid values can be 0. As new data come in, the values can be stored in the array and DataAccelerator.extendLength can be called to update the number of valid values.

Argument	Default	Description
xData	(Mandatory)	The array to store the x values.
xDataLen	(Mandatory)	The number of valid values.

See also:

- 3.15.6 Constructor(xData as CDArrayMBS) 323
- 3.15.7 Constructor(xData() as double) 324

3.15.6 Constructor(xData as CDArrayMBS)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Creates a DataAccelerator object.

Notes: The xData is an array for storing the x data values, and xDataLen is number of valid values in the array.

To conserve memory, DataAccelerator will only keep a reference to the array and not copy it. The caller must ensure the array remains valid for as long as the DataAccelerator is in use.

For real time chart usage, the array acts as a buffer to store real time data. Initially the number of valid values can be 0. As new data come in, the values can be stored in the array and `DataAccelerator.extendLength` can be called to update the number of valid values.

Argument	Default	Description
<code>xData</code>	(Mandatory)	The array to store the x values.
<code>xDataLen</code>	(Mandatory)	The number of valid values.

See also:

- 3.15.5 Constructor 323
- 3.15.7 Constructor(`xData()` as double) 324

3.15.7 Constructor(`xData()` as double)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Creates a `DataAccelerator` object.

Notes: The `xData` is an array for storing the x data values, and `xDataLen` is number of valid values in the array.

To conserve memory, `DataAccelerator` will only keep a reference to the array and not copy it. The caller must ensure the array remains valid for as long as the `DataAccelerator` is in use.

For real time chart usage, the array acts as a buffer to store real time data. Initially the number of valid values can be 0. As new data come in, the values can be stored in the array and `DataAccelerator.extendLength` can be called to update the number of valid values.

Argument	Default	Description
<code>xData</code>	(Mandatory)	The array to store the x values.
<code>xDataLen</code>	(Mandatory)	The number of valid values.

See also:

- 3.15.5 Constructor 323
- 3.15.6 Constructor(`xData` as `CArrayMBS`) 323

3.15.8 `extendLength(len as Integer)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Extends the number of valid values in the arrays.

Notes: This method is usually used by real time charts. After storing the real time data into the data arrays, the `extendLength` method should be used to notify the `DataAccelerator` of the new data. You must ensure all data arrays in the `DataAccelerator` are updated to contain valid data before calling this method.

You can only increase the number of valid values, but not decrease it.

Argument	Default	Description
<code>length</code>	(Mandatory)	The number of valid values in the arrays.

3.15.9 `setSubsetRange(xStart as double, xEnd as double, resolution as Integer = 0)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Set the data range used for chart plotting.

Notes: For zoomable and scrollable chart, depending on the zoom level, only part of the data may be visible. The `setSubsetRange` method should be used to set the visible x range so that the `DataAccelerator` can supply the chart with the necessary data only.

The resolution argument is a hint of the number of pixels available in the x direction for plotting the data. This argument avoids the `DataAccelerator` from providing too much data to the chart. For example, if only 600 pixels are available in the x direction, it is not necessary to provide 10 million points to the chart, as they will overlap and block each others and the vast majority would not be visible. The `DataAccelerator` employs a special algorithm that can reduce the data to match the resolution without noticeably affecting the shape of the plotted chart.

Argument	Default	Description
<code>xStart</code>	(Mandatory)	The x value at the start of the visible range.
<code>xEnd</code>	(Mandatory)	The x value at the end of the visible range.
<code>resolution</code>	0	The approximate number of pixels available in the x direction for plotting the chart. In the most common case that the x-axis is horizontal, this is the plot area width. If this argument is not provided, <code>ChartDirector</code> will automatically choose the resolution to use.

3.16 class CDDatasetMBS

3.16.1 class CDDatasetMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The DataSet class represents data sets.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Xojo Plugins, version 18.5pr7](#)
- [MBS Real Studio Plugins, version 12.4pr3](#)
- [MBS Real Studio Plugins, version 12.3pr9](#)

Xojo Developer Magazine

- [18.5, page 77: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.16.2 Methods

3.16.3 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.16.4 getDataColor as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the main color used to draw the data set.

Notes: Returns the main color used to draw the data set.

3.16.5 getDataName as string

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the name of the data set.

Notes: Return the name of the data set.

3.16.6 getLegendIcon as string

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the CDML representation of the legend icon for the data set.

Notes: The CDML representation can be used in any ChartDirector API that supports CDML. Note that the CDML representation is only valid in the BaseChart object that contains the data set. It is not allowed to use the CDML representation obtained from one chart in a different chart.

The legend icon is the same icon that would appear in the LegendBox. If you modify the icon using methods of the LegendBox object (such as using LegendBox.setKeyBorder to set the legend icon border), the modification will also apply to the icon returned by this method.

Returns the CDML representation of the legend icon for the data set.

3.16.7 getPosition(i as Integer) as Double

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the positional value of a data point.

Notes: The positional value is the value used to position the data point. Usually, the positional value is equal to the data value. However, for chart types in which the data sets are combined using the Stack or Percentage method, the positional value can be different from the data value. For example, for stacked charts, the positional value of a data point is equal to the accumulative value of the corresponding data points from data sets added before the target data set.

Argument	Default	Description
i	(Mandatory)	The data point index of the data point. The first data point is 0; the nth data point is (n - 1).

Returns the positional value of the data point.

3.16.8 getYAxis as CDAxisMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y-axis to use when drawing the data set.

Notes: Returns an Axis object representing the y-axis to use when drawing the data set.

3.16.9 `getValue(i as Integer) as Double`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the value of a data point.

Notes:

Argument	Default	Description
<code>i</code>	(Mandatory)	The data point index of the data point. The first data point is 0; the nth data point is (n - 1).

Returns the value of the data point.

3.16.10 `setData(data() as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data for this dataset.

3.16.11 `setDataColor(dataColor as color, edgeColor as color, shadowColor as color, shadowEdgeColor as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setDataColor` method, but uses color instead of integer data type for passing color values.

See also:

- 3.16.12 `setDataColor(dataColor as Integer, edgeColor as Integer = -1, shadowColor as Integer = -1, shadowEdgeColor as Integer = -1)` 328

3.16.12 `setDataColor(dataColor as Integer, edgeColor as Integer = -1, shadowColor as Integer = -1, shadowEdgeColor as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors used to draw the data set.

Notes: The usage of the colors depend on the layer type. For example, in a bar layer, the `dataColor` and `edgeColor` are used as the fill and border color of the bar. In a line layer, the `dataColor` is used as the color of the line, while the `edgeColor` is used as the border of the line for 3D lines, and is ignored for 2D lines..

See also:

Argument	Default	Description
dataColor	(Mandatory)	The main color used to draw the data set.
edgeColor	-1	The color used to draw the edges or borders of the data set, if any. -1 means that the edges are drawn using the default border color of the layer (defined using Layer.setBorderColor).
shadowColor	-1	The color to use to draw 3D shadows of the data set, if any. -1 means the shadow color will be a "darker" version of the dataColor, created by reducing the RGB intensities of the dataColor in half.
shadowEdgeColor	-1	The color to use to draw edges of the 3D shadows of the data set, if any. -1 means the shadow color will be a "darker" version of the edgeColor, created by reducing the RGB intensities of the edgeColor in half.

- 3.16.11 setDataColor(dataColor as color, edgeColor as color, shadowColor as color, shadowEdgeColor as color) 328

3.16.13 setDataLabelFormat(formatString as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data label format for the data set.

Notes: To set the label format for all data sets in a layer, use Layer.setDataLabelFormat.

For details description of data label format, please refer to Layer.setDataLabelFormat.

Argument	Default	Description
formatString	(Mandatory)	The format string.

3.16.14 setDataLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = 0, fontangle as Double = 0) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables data labels and sets their styles.

Notes: To set the data label styles for all data sets in a layer, use Layer.setDataLabelStyle.

For details description of data labels, please refer to :Layer.setDataLabelStyle.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See also:

Argument	Default	Description
font	""	The font used to draw the labels.
fontSize	8	The font size used to draw the labels.
fontColor	TextColor	The color used to draw the labels.
fontAngle	0	The rotation angle of the labels.

- 3.16.15 setDataLabelStyle(font as string, fontsize as Double, fontcolor as color, fontangle as Double = 0) as CDTextBoxMBS 330

3.16.15 setDataLabelStyle(font as string, fontsize as Double, fontcolor as color, fontangle as Double = 0) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setDataLabelStyle method, but uses color instead of integer data type for passing color values.

See also:

- 3.16.14 setDataLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = 0, fontangle as Double = 0) as CDTextBoxMBS 329

3.16.16 setDataName(name as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the name of the data set.

Notes: The name will be used in the legend box, if one is available for the chart.

Argument	Default	Description
name	(Mandatory)	The name of the data set.

3.16.17 setDataSymbol(drawobj as CDDrawAreaMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a DrawArea object as the graphics symbol to plot the data points.

Notes: In the current version of ChartDirector, data symbols are supported only in LineLayer, SplineLayer, StepLineLayer and ScatterLayer. To use data symbols in other layer types, add a ScatterLayer on top of that layer.

See also:

3.16. CLASS CDDATASET MBS 331

Argument	Default	Description
obj	(Mandatory)	A DrawArea object to be used as the symbol.

- 3.16.18 setDataSymbol(file as folderitem) 331
- 3.16.19 setDataSymbol(ImageFilePath as string) 332
- 3.16.20 setDataSymbol(pic as Picture) 333
- 3.16.21 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 333
- 3.16.22 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgeColor as color) 334
- 3.16.23 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 335
- 3.16.24 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 335

3.16.18 setDataSymbol(file as folderitem)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load an image from a file and use it as the graphics symbol to plot the data points.

Notes: ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to BaseChart.setSearchPath on the directory that ChartDirector will search for the file.

In the current version of ChartDirector, data symbols are supported only in LineLayer, SplineLayer, StepLineLayer and ScatterLayer. To use data symbols in other layer types, add a ScatterLayer on top of that layer.

Argument	Default	Description
image	(Mandatory)	The filename of the image file. The image type is determined based on file extension, which must be png, jpg/jpeg, gif or wbmp/wmp.

See also:

- 3.16.17 setDataSymbol(drawobj as CDDrawAreaMBS) 330
- 3.16.19 setDataSymbol(ImageFilePath as string) 332
- 3.16.20 setDataSymbol(pic as Picture) 333

- 3.16.21 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 333
- 3.16.22 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgeColor as color) 334
- 3.16.23 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 335
- 3.16.24 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 335

3.16.19 setDataSymbol(ImageFilePath as string)

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load an image from a file and use it as the graphics symbol to plot the data points.

Notes: ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to BaseChart.setSearchPath on the directory that ChartDirector will search for the file.

In the current version of ChartDirector, data symbols are supported only in LineLayer, SplineLayer, StepLineLayer and ScatterLayer. To use data symbols in other layer types, add a ScatterLayer on top of that layer.

Argument	Default	Description
image	(Mandatory)	The filename of the image file. The image type is determined based on file extension, which must be png, jpg/jpeg, gif or wbmp/wmp.

See also:

- 3.16.17 setDataSymbol(drawobj as CDDrawAreaMBS) 330
- 3.16.18 setDataSymbol(file as folderitem) 331
- 3.16.20 setDataSymbol(pic as Picture) 333
- 3.16.21 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 333
- 3.16.22 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgeColor as color) 334
- 3.16.23 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 335
- 3.16.24 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 335

3.16.20 setDataSymbol(pic as Picture)

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a picture object as the graphics symbol to plot the data points.

Notes: In the current version of ChartDirector, data symbols are supported only in LineLayer, SplineLayer, StepLineLayer and ScatterLayer. To use data symbols in other layer types, add a ScatterLayer on top of that layer.

Argument	Default	Description
obj	(Mandatory)	A picture object to be used as the symbol.

See also:

- 3.16.17 setDataSymbol(drawobj as CDDrawAreaMBS) 330
- 3.16.18 setDataSymbol(file as folderitem) 331
- 3.16.19 setDataSymbol(ImageFilePath as string) 332
- 3.16.21 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 333
- 3.16.22 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgeColor as color) 334
- 3.16.23 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 335
- 3.16.24 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 335

3.16.21 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a custom polygon as the graphics symbol to plot the data points.

Notes: In the current version of ChartDirector, data symbols are supported only in LineLayer, SplineLayer, StepLineLayer and ScatterLayer. To use data symbols in other layer types, add a ScatterLayer on top of that layer.

See also:

- 3.16.17 setDataSymbol(drawobj as CDDrawAreaMBS) 330
- 3.16.18 setDataSymbol(file as folderitem) 331
- 3.16.19 setDataSymbol(ImageFilePath as string) 332

Argument	Default	Description
polygon	(Mandatory)	An array of integers representing the coordinates the polygon vertices. See Shape Specification on how the custom shape is defined.
size	11	The nominal width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means the color of the data set will be used.
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means the edge color of the data set will be used.

- 3.16.20 setDataSymbol(pic as Picture) 333
- 3.16.22 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgeColor as color) 334
- 3.16.23 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 335
- 3.16.24 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 335

3.16.22 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setDataSymbol method, but uses color instead of integer data type for passing color values.

See also:

- 3.16.17 setDataSymbol(drawobj as CDDrawAreaMBS) 330
- 3.16.18 setDataSymbol(file as folderitem) 331
- 3.16.19 setDataSymbol(ImageFilePath as string) 332
- 3.16.20 setDataSymbol(pic as Picture) 333
- 3.16.21 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 333
- 3.16.23 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 335
- 3.16.24 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 335

3.16.23 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses one of the built-in symbols as the graphics symbol to plot the data points.

Notes: In the current version of ChartDirector, data symbols are supported only in LineLayer, SplineLayer, StepLineLayer and ScatterLayer. To use data symbols in other layer types, add a ScatterLayer on top of that layer.

Argument	Default	Description
symbol	(Mandatory)	One of the predefined shape constants representing the symbol shape. See Shape Specification for the available built-in shapes.
size	5	The width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means the color of the data set will be used.
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means the edge color of the data set will be used.
lineWidth	1	The line width used for drawing the symbols.

See also:

- 3.16.17 setDataSymbol(drawobj as CDDrawAreaMBS) 330
- 3.16.18 setDataSymbol(file as folderitem) 331
- 3.16.19 setDataSymbol(ImageFilePath as string) 332
- 3.16.20 setDataSymbol(pic as Picture) 333
- 3.16.21 setDataSymbol(polygon() as Integer, size as Integer = 11, fillcolor as Integer = -1, edgeColor as Integer = -1) 333
- 3.16.22 setDataSymbol(polygon() as Integer, size as Integer, fillcolor as color, edgeColor as color) 334
- 3.16.24 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 335

3.16.24 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setDataSymbol method, but uses color instead of integer data type for passing color values.

See also:

- 3.16.17 setDataSymbol(drawobj as CDDrawAreaMBS) 330

- 3.16.18 setDataSymbol(file as folderitem) 331
- 3.16.19 setDataSymbol(ImageFilePath as string) 332
- 3.16.20 setDataSymbol(pic as Picture) 333
- 3.16.21 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 333
- 3.16.22 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgeColor as color) 334
- 3.16.23 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 335

3.16.25 setLineWidth(w as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the line width of lines when drawing the data set.

Notes: This method only applies to layers that employ lines to represent data (e.g. line layer).

If this method is not called, the default line width for the layer will be used (set using Layer.setLineWidth).

Argument	Default	Description
w	(Mandatory)	The width of the line in pixels.

3.16.26 setSymbolOffset(offsetX as Integer, offsetY as Integer)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Offset the symbols in the x and y directions in pixel unit.

Notes:

Argument	Default	Description
xOffset	(Mandatory)	The x offset in pixels. A positive value mean shifting to the right.
yOffset	(Mandatory)	The y offset in pixels. A positive value mean shifting to the bottom.

3.16.27 setUseYAxis(axis as CDAxisMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Determine if the primary or secondary y-axis should be used when drawing the data set.

Notes: To set the y-axis to use for all data sets within a layer, use Layer.setUseYAxis2.

Note: DataSet.setUseYAxis is a more general method that can support more than 2 y-axes.

Argument	Default	Description
b	true	A true value means the secondary y-axis will be used. A false value means the primary y-axis will be used.

See also:

- 3.16.28 setUseYAxis(b as boolean=true) 337

3.16.28 setUseYAxis(b as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Determine the y-axis to use when drawing the data set.

Notes: To set the y-axis to use for all data sets within a layer, use Layer.setUseYAxis.

Argument	Default	Description
a	(Mandatory)	The y-axis to use when drawing the data set.

See also:

- 3.16.27 setUseYAxis(axis as CDAxisMBS) 336

3.17 class CDDiscreteHeatMapLayerMBS

3.17.1 class CDDiscreteHeatMapLayerMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The DiscreteHeatMapLayer class represents discrete heat map layers. DiscreteHeatMapLayer is a subclass of Layer.

Notes: DiscreteHeatMapLayer objects are created by using XYChart.addDiscreteHeatMapLayer.

A discrete heat map consists of rectangular cells on a rectangular grid. The data values of the cells determine their colors. The data value to color mapping is controlled by the ColorAxis object of the DiscreteHeatMapLayer, obtainable using DiscreteHeatMapLayer.colorAxis.

Subclass of the CDLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.2](#)
- [MBS Xojo Plugins, version 21.2pr5](#)
- [ChartDirector 7 update](#)

Xojo Developer Magazine

- [20.1, page 44: Cool Charts and Heatmaps, Using Monkeybread Software's ChartDirector Plugin by Stefanie Juchmes](#)

3.17.2 Methods

3.17.3 colorAxis as CDColorAxisMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the ColorAxis object representing the color axis (color legend).

3.17.4 setCellGap(gap as integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the gap between cells.

Notes: By default, the cells will touch each others with no gap in between.

Argument	Default	Description
gap	(Mandatory)	The gap between cells in pixels.

3.17.5 setColorAxis(x as integer, y as integer, alignment as integer, length as integer, orientation as integer) as CDCColorAxisMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the position, length and orientation of the color axis (color legend).

Notes:

Argument	Default	Description
x	(Mandatory)	The x-coordinate of the reference point used to position the color axis.
y	(Mandatory)	The y-coordinate of the reference point used to position the color axis.
alignment	(Mandatory)	The alignment of the color axis with respect to the reference point. For example, a value of TopLeft means the reference point is the top- left corner of the color axis. See Alignment Specification for supported alignment types.
length	(Mandatory)	The length of the color axis in pixels.
orientation	(Mandatory)	The orientation of the color axis. A value of Top/Bottom means the axis is horizontal, and the axis labels are at top/bottom side of the axis. A value of Left/Right means the axis is vertical, and the axis labels are at the left/right side of the axis.

3.17.6 setDirectColoring(b as boolean = true)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets whether the data values are the colors themselves or if a CDCColorAxisMBS will be used to map the data values to colors.

Notes: By default, a ColorAxis will be used to map the data values to colors. The setDirectColoring method allows directly specifying the colors of the cells without going through the CDCColorAxisMBS.

Argument	Default	Description
b	true	A true value means the data values of the cells are their colors. A false value means a CDCColorAxisMBS will be used to map the data values to colors.

3.18 class CDDrawAreaMBS

3.18.1 class CDDrawAreaMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The DrawArea class represents drawing surfaces, with methods for performing graphics primitives (such as drawing lines, shapes and texts) and the surfaces.

Notes: All ChartDirector chart objects contains an internal DrawArea object for drawing the charts. This DrawArea object is accessible via BaseChart.getDrawArea or BaseChart.makeChart. This allows developers to apply custom graphics operations on the charts.

ChartDirector also supports creating standalone DrawArea objects by calling the DrawArea.DrawArea constructor. One common application is to use ChartDirector as a general purpose graphics library (e.g. adding text annotations to existing images, creating GIF buttons on the fly, etc).

Blog Entries

- [Several ways for picture to PDF in MBS Plugins](#)
- [News from the MBS Xojo Plugins Version 21.2](#)
- [MBS Xojo Plugins, version 18.5pr7](#)
- [MBS Xojo Plugins, version 18.5pr6](#)
- [A chart similar to the one on Apple Watch](#)
- [MBS Real Studio Plugins, version 12.3pr9](#)

Xojo Developer Magazine

- [21.6, page 90: From 0 to 100, Creating Gauges with MBS and ChartDirector by Stefanie Juchmes](#)
- [21.6, page 85: From 0 to 100, Creating Gauges with MBS and ChartDirector by Stefanie Juchmes](#)
- [18.5, page 74: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.18.2 Methods

3.18.3 adjustBrightness(c as Integer, brightness as Double) as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a color that is a darkened or brightened version of the given color.

Notes: A brightness less than 1 means the color is darkened, while a brightness greater than 1 means the color is brightened. For example, a brightness of 0.5 means the color is half as bright as the original color. If the original color is red, the color will become dark red. Conversely, a brightness of 2 means the color is

twice as bright as the original color. If the original color is red, the color will become light red.

Arguments:

Argument	Default	Description
c	(Mandatory)	The given color.
brightness	(Mandatory)	A non-negative number represent the factor to darken or brighten the color.

Return Value

A 32-bit integer representing the darkened or brightened color.

3.18.4 `affineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `affineTransform` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.18.5 `affineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)` 341

3.18.5 `affineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Perform affine transformation on the drawing surface.

Notes: An affine transformation is an operation where every pixel is copied to another pixel according to the formula.

$xDest = a * xSrc + b * ySrc + cyDest = d * xSrc + e * ySrc + f$
 where $(xSrc, ySrc)$ is a source pixel, and $(xDest, yDest)$ is where it should go to.

Many graphics operation, such as translation, rotation, and resizing, can be considered as a special case of affine transformation.

See also:

Argument	Default	Description
a	(Mandatory)	The parameter 'a' in the coordinate transformation formula " $x_{Dest} = a * x_{Src} + b * y_{Src} + c$ ".
b	(Mandatory)	The parameter 'b' in the coordinate transformation formula " $x_{Dest} = a * x_{Src} + b * y_{Src} + c$ ".
c	(Mandatory)	The parameter 'c' in the coordinate transformation formula " $x_{Dest} = a * x_{Src} + b * y_{Src} + c$ ".
d	(Mandatory)	The parameter 'd' in the coordinate transformation formula " $y_{Dest} = d * x_{Src} + e * y_{Src} + f$ ".
e	(Mandatory)	The parameter 'e' in the coordinate transformation formula " $y_{Dest} = d * x_{Src} + e * y_{Src} + f$ ".
f	(Mandatory)	The parameter 'f' in the coordinate transformation formula " $y_{Dest} = d * x_{Src} + e * y_{Src} + f$ ".
bgColor	FFFFFF	The background color used to fill destination pixels that are not mapped to any source pixels.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

- 3.18.4 `affineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)` 341

3.18.6 `angleGradientColor(cx as double, cy as double, a1 as double, a2 as double, r1 as double, r2 as double, c() as integer) as integer`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an angle gradient color.

Notes: An angle gradient color is a color that depends on the compass bearing of the pixel relative to the center point. The north (upward) direction is zero degree, and the angle is the clockwise angle from the north direction.

ChartDirector supports "smooth angle gradient" and "step angle gradient". For a smooth angle gradient, the color changes smoothly from one color stop to the next color stop. For a step angle gradient, the color remains unchanged between the color stops, but change abruptly once it reaches the next color stop.

The color stops are defined as an array of offsets and colors. For a smooth angle gradient, the offsets and colors should be in the following format:

```
offset0, color0, offset1, color1, .... offsetN, colorN
```

The first offset (offset0) should be 0, which represents the start angle of the gradient, and color0 is the color at that offset. The last offset (offsetN) can be any number not greater than 100000. It represents the end angle of the gradient, and colorN represents the color at the that offset. The other offsets and colors

represent the angles and colors of the color stops in between.

For a step angle gradient, the offsets and colors should be in the following format:

offset0, color0, offset1, color1, ..., offsetP, colorP, offsetN

Note that for a step color gradient, there is one more offset than colors. If there are 10 color steps, it will have 10 colors and 11 offsets. The offsets represent angles and are interpreted the same as in smooth angle gradient. The colors are applied between the offsets. For example, color0 is the color between offset0 and offset1, and color1 is the color between offset1 and offset2.

Argument	Default	Description
cx	(Mandatory)	The x coordinate of the center point.
cy	(Mandatory)	The y coordinate of the center point.
a1	(Mandatory)	The start angle of the gradient.
a2	(Mandatory)	The end angle of the gradient.
r1	(Mandatory)	The minimum radius for which the gradient is applicable.
r2	(Mandatory)	The maximum radius for which the gradient is applicable.
colorArray	(Mandatory)	An array defining the offsets and colors of the color stops along the gradient.

Returns a 32-bit integer representing the angle gradient color.

3.18.7 arc(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, c as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a circular or elliptical arc.

Notes:

Argument	Default	Description
cx	(Mandatory)	The x coordinate of the center of the circle or ellipse.
cy	(Mandatory)	The y coordinate of the center of the circle or ellipse.
rx	(Mandatory)	The horizontal radius of the circle or ellipse.
ry	(Mandatory)	The vertical radius of the circle or ellipse.
a1	(Mandatory)	The start angle of the arc in degrees. The angle is measured clockwise, with 0 degree being the upward pointing direction.
a2	(Mandatory)	The end angle of the arc in degrees. The angle is measured clockwise, with 0 degree being the upward pointing direction.
c	(Mandatory)	The color of the arc.

3.18.8 `circle(cx as Integer, cy as Integer, rx as Integer, ry as Integer, edgeColor as color, fillColor as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other circle method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.9 `circle(cx as Integer, cy as Integer, rx as Integer, ry as Integer, edgeColor as Integer, fillColor as Integer)` 344

3.18.9 `circle(cx as Integer, cy as Integer, rx as Integer, ry as Integer, edgeColor as Integer, fillColor as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a circle or an ellipse.

Notes:

Argument	Default	Description
<code>cx</code>	(Mandatory)	The x coordinate of the center of the circle or ellipse.
<code>cy</code>	(Mandatory)	The y coordinate of the center of the circle or ellipse.
<code>rx</code>	(Mandatory)	The horizontal radius of the circle or ellipse.
<code>ry</code>	(Mandatory)	The vertical radius of the circle or ellipse.
<code>edgeColor</code>	(Mandatory)	The border color. To disable border, set the <code>edgeColor</code> the same as the <code>fillColor</code> .
<code>fillColor</code>	(Mandatory)	The fill color. To disable filling, set the <code>fillColor</code> to <code>Transparent</code> .

See also:

- 3.18.8 `circle(cx as Integer, cy as Integer, rx as Integer, ry as Integer, edgeColor as color, fillColor as color)` 344

3.18.10 `clone(d as CDDrawAreaMBS, x as Integer, y as Integer, align as Integer, newWidth as Integer = -1, newHeight as Integer = -1, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copy the current DrawArea object to another DrawArea object, with optional resizing.

Notes:

3.18.11 Constructor

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Argument	Default	Description
d	(Mandatory)	The destination DrawArea object where the current DrawArea is copied to.
x	(Mandatory)	The x coordinate of a reference point in the destination DrawArea object.
y	(Mandatory)	The y coordinate of a reference point in the destination DrawArea object.
align	(Mandatory)	The alignment of the current DrawArea relative to the reference point. See Alignment Specification for supported alignment types.
newWidth	-1	The new width to which the current DrawArea will be resized to before copying. The default value -1 means no resizing.
newHeight	-1	The new height to which the current DrawArea will be resized to before copying. The default value -1 means no resizing.
filter	LinearFilter	The filter to use for re-sampling. (Only applies if there is resizing.)
blur	1	The blur factor to use for re-sampling. (Only applies if there is resizing.)

Function: Creates a DrawArea object.

3.18.12 cylinder(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, depthX as Integer, depthY as Integer, edgeColor as color, fillColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other cylinder method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.13 cylinder(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, depthX as Integer, depthY as Integer, edgeColor as Integer, fillColor as Integer) 345

3.18.13 cylinder(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, depthX as Integer, depthY as Integer, edgeColor as Integer, fillColor as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a cylinder surface as the area spanned by moving an arc.

Notes:

See also:

- 3.18.12 cylinder(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, depthX as Integer, depthY as Integer, edgeColor as color, fillColor as color) 345

Argument	Default	Description
cx	(Mandatory)	The x coordinate of the center of the circle or ellipse for the arc.
cy	(Mandatory)	The y coordinate of the center of the circle or ellipse for the arc.
rx	(Mandatory)	The horizontal radius of the circle or ellipse.
ry	(Mandatory)	The vertical radius of the circle or ellipse.
a1	(Mandatory)	The start angle of the arc. The angle is measured clockwise, with 0 degree being the upward pointing direction.
a2	(Mandatory)	The end angle of the arc. The angle is measured clockwise, with 0 degree being the upward pointing direction.
depthX	(Mandatory)	The x displacement representing the motion of the arc to span the cylinder.
depthY	(Mandatory)	The y displacement representing the motion of the arc to span the cylinder.
edgeColor	(Mandatory)	The border color. To disable border, set the edgeColor the same as the fillColor.
fillColor	(Mandatory)	The fill color. To disable filling, set the fillColor to Transparent.

3.18.14 dashLineColor(colorvalue as color, patternCode as Integer = &h0505) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other dashLineColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.15 dashLineColor(colorvalue as Integer, patternCode as Integer = &h0505) as Integer 346

3.18.15 dashLineColor(colorvalue as Integer, patternCode as Integer = &h0505) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a dash line color.

Notes: A dash line color is a dynamic color that switches on and off periodically. When it is used to draw a line, the line will appear as a dash line.

The style of the dash line is defined by a pattern code, which is a 4-byte integer. A value of PPQRRSS (in hex) means the first PP pixels are turned on, followed by QQ pixels turned off, followed by RR pixels turned on, followed by SS pixels turned off, and then restart from PP again.

ChartDirector comes from several predefined constants for common dash line patterns.

ConstantValue (in Hex)Dash Line Style

Return Value

DashLine	00000505
DotLine	00000202
DotDashLine	05050205
AltDashLine	0A050505

Argument	Default	Description
color	(Mandatory)	The color to draw the dash line.
patternCode	DashLine	A 4-byte integer representing the style of the dash line.

A 32-bit integer representing the dash line color.

See also:

- 3.18.14 dashLineColor(colorvalue as color, patternCode as Integer = &h0505) as Integer 346

3.18.16 enableVectorOutput

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Enables true vector graphics output.

Notes: By default, when creating the output image, ChartDirector draws directly onto an output buffer representing the bitmap of the image. For example, for a chart 800 x 600 pixels in size, the output buffer may represent a 800 x 600 bitmap. The output buffer size is unchanged no matter how many items are drawn onto it. Even if the output contains 1 million elements (eg. 1 million symbols), the size of the output buffer is still the same.

On the other hand, a true vector output is indefinitely scalable and can be considered as having infinite resolution. To produce a true vector output, it is necessary to remember the graphics operations for every element in the output buffer. The output buffer size is therefore proportional to the number of elements to draw.

This method tells ChartDirector that it needs to remember the graphics operations to prepare for true vector output. If true vector output is needed, this method should be called immediately after creating the DrawArea object.

If this method is not called, and a vector graphics output format is used (such as SVG), instead of a true vector output, ChartDirector will output a raster image using the vector graphics format. (Most vector graphics formats support embedded raster images.)

3.18.17 fill(x as Integer, y as Integer, colorvalue as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other fill method, but uses color instead of integer data type for passing color values.
See also:

- 3.18.18 fill(x as Integer, y as Integer, colorvalue as color, borderColor as color) 348
- 3.18.19 fill(x as Integer, y as Integer, colorvalue as Integer) 348
- 3.18.20 fill(x as Integer, y as Integer, colorvalue as Integer, borderColor as Integer) 349

3.18.18 fill(x as Integer, y as Integer, colorvalue as color, borderColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other fill method, but uses color instead of integer data type for passing color values.
See also:

- 3.18.17 fill(x as Integer, y as Integer, colorvalue as color) 348
- 3.18.19 fill(x as Integer, y as Integer, colorvalue as Integer) 348
- 3.18.20 fill(x as Integer, y as Integer, colorvalue as Integer, borderColor as Integer) 349

3.18.19 fill(x as Integer, y as Integer, colorvalue as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Flood fill a region using the specified color.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate one pixel inside the region to be filled.
y	(Mandatory)	The y coordinate one pixel inside the region to be filled.
color	(Mandatory)	The color used to fill the region.

See also:

- 3.18.17 fill(x as Integer, y as Integer, colorvalue as color) 348
- 3.18.18 fill(x as Integer, y as Integer, colorvalue as color, borderColor as color) 348
- 3.18.20 fill(x as Integer, y as Integer, colorvalue as Integer, borderColor as Integer) 349

3.18.20 fill(x as Integer, y as Integer, colorvalue as Integer, borderColor as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill a region using the specified color, where the region is bounded by a given border color.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate one pixel inside the region to be filled.
y	(Mandatory)	The y coordinate one pixel inside the region to be filled.
color	(Mandatory)	The color used to fill the region.
borderColor	(Mandatory)	The color of the border that bounds the region.

See also:

- 3.18.17 fill(x as Integer, y as Integer, colorvalue as color) 348
- 3.18.18 fill(x as Integer, y as Integer, colorvalue as color, borderColor as color) 348
- 3.18.19 fill(x as Integer, y as Integer, colorvalue as Integer) 348

3.18.21 getARGBColor(c as Integer) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Obtain the ARGB color given a palette color.

Notes: If the given color is already in ARGB format, the same value is returned.

Argument	Default	Description
c	(Mandatory)	The color to be changed to ARGB format.

Return Value

The ARGB color converted from the given color.

3.18.22 getHeight as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the height of the drawing surface.

3.18.23 `getPixel(x as Integer, y as Integer) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the color of a pixel.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of the pixel.
y	(Mandatory)	The y coordinate of the pixel.

Return Value

The color of the pixel.

3.18.24 `getWidth as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the width of the drawing surface.

3.18.25 `gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `gradientColor` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.18.26 `gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 350
- 3.18.27 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer` 351
- 3.18.28 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer` 352

3.18.26 `gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a multi-point linear gradient color.

Notes: This method is for backward compatibility. Use `DrawArea.linearGradientColor2` instead.

Argument	Default	Description
colorArray	(Mandatory)	An array defining the positions and colors of the pixels along the reference gradient line segment.
angle	90	The direction of the reference gradient line segment in degrees, measured clockwise, with 0 degree as the upward pointing direction. The default direction is horizontal from left to right (90 degrees).
scale	1.0	The scaling factor for the reference gradient line segment. By default, the reference gradient line segment is 256 pixels in length. The scaling factor can be use to stretch or compress the gradient line segment.
startX	0	The x coordinate of the starting point of the reference gradient line segment.
startY	0	The y coordinate of the starting point of the reference gradient line segment.

Return Value

A 32-bit integer representing the linear gradient color.

See also:

- 3.18.25 `gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 350
- 3.18.27 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer` 351
- 3.18.28 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer` 352

3.18.27 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `gradientColor` method, but uses `color` instead of `integer` data type for passing color values.

See also:

- 3.18.25 `gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 350
- 3.18.26 `gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 350
- 3.18.28 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer` 352

3.18.28 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a two-point linear gradient color.

Notes: This method is for backward compatibility. Use `DrawArea.linearGradientColor` instead.

Argument	Default	Description
<code>startX</code>	(Mandatory)	The x coordinate of the starting point of the reference gradient line segment.
<code>startY</code>	(Mandatory)	The y coordinate of the starting point of the reference gradient line segment.
<code>endX</code>	(Mandatory)	The x coordinate of the ending point of the reference gradient line segment.
<code>endY</code>	(Mandatory)	The y coordinate of the ending point of the reference gradient line segment.
<code>startColor</code>	(Mandatory)	The color at the starting point of the reference gradient line segment.
<code>endColor</code>	(Mandatory)	The color at the ending point of the reference gradient line segment.

Return Value

A 32-bit integer representing the linear gradient color.

See also:

- 3.18.25 `gradientColor(colors() as color, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 350
- 3.18.26 `gradientColor(colors() as Integer, angle as Double = 90, scale as Double = 1.0, startX as Integer = 0, startY as Integer = 0) as Integer` 350
- 3.18.27 `gradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color) as Integer` 351

3.18.29 `halfColor(c as Integer) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a color that is half the intensity of the given color.

Notes:

Argument	Default	Description
<code>c</code>	(Mandatory)	The given color.

Return Value

A 32-bit integer representing the half intensity color.

3.18.30 hCylinderTransform(yDiameter as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other hCylinderTransform method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.31 hCylinderTransform(yDiameter as Integer, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 353

3.18.31 hCylinderTransform(yDiameter as Integer, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Wraps the drawing surface onto a horizontal cylinder.

Notes:

Argument	Default	Description
yDiameter	(Mandatory)	The diameter of the cylinder in pixels.
bgColor	FFFFFF	The background color used to fill the space left after transformation.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

See also:

- 3.18.30 hCylinderTransform(yDiameter as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0) 353

3.18.32 hFlip

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Flip the drawing surface along the central horizontal line.

3.18.33 hline(x1 as Integer, x2 as Integer, y as Integer, c as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a horizontal line.

Notes:

Argument	Default	Description
x1	(Mandatory)	The x coordinate of the first end-point of the line.
x2	(Mandatory)	The x coordinate of the second end-point of the line.
y	(Mandatory)	The y coordinate of the line.
c	(Mandatory)	The color of the line.

3.18.34 `hTriangleTransform(tWidth as Integer = -1, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Wraps the drawing surface onto a horizontal triangle pointing leftwards.

Notes:

Argument	Default	Description
tWidth	-1	The width of the triangle in pixels.
bgColor	FFFFFF	The background color used to fill the space left after transformation.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

See also:

- 3.18.35 `hTriangleTransform(tWidth as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0)` 354

3.18.35 `hTriangleTransform(tWidth as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `hTriangleTransform` method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.34 `hTriangleTransform(tWidth as Integer = -1, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)` 354

3.18.36 `initDynamicLayer`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Initializes a dynamic layer for drawing text and shapes.

Notes: This method clears the existing dynamic layer, or creates a new one if there is no existing dynamic layer. This method returned a `DrawArea` object that can be used to draw on the dynamic layer. The

dynamic layer can later be removed using `BaseChart.removeDynamicLayer`.

The design of the dynamic layer is for drawing small, rapidly updatable contents for desktop applications. For example, the dynamic layer can be used to implement a cross-hair mouse cursor, with text showing the location of the mouse cursor. To do this, in the mouse move event handler, `BaseChart.initDynamicLayer` can be used to create or clear the dynamic layer. The returned `DrawArea` object can then be used to draw the cross hair cursor (as two straight lines) and the text. When the mouse cursor leaves the chart, `BaseChart.removeDynamicLayer` can be used in the mouse out event handler to remove the cross-hair cursor and the text.

Note that as long as the dynamic layer is not removed with `BaseChart.removeDynamicLayer`, only the returned `DrawArea` object should be used to draw things on the dynamic layer. No other objects should be used to draw on the chart.

Returns a `DrawArea` object that can be used to add text and shapes to the dynamic layer.

3.18.37 `line(x1 as Double, y1 as Double, x2 as Double, y2 as Double, colorValue as color, lineWidth as Integer = 1)`

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draw a straight line.

Notes:

Argument	Default	Description
<code>x1</code>	(Mandatory)	The x coordinate of the first end-point of the line.
<code>y1</code>	(Mandatory)	The y coordinate of the first end-point of the line.
<code>x2</code>	(Mandatory)	The x coordinate of the second end-point of the line.
<code>y2</code>	(Mandatory)	The y coordinate of the second end-point of the line.
<code>c</code>	(Mandatory)	The color of the line.
<code>lineWidth</code>	1	The line width (thickness).

See also:

- 3.18.38 `line(x1 as Double, y1 as Double, x2 as Double, y2 as Double, colorValue as Integer, lineWidth as Integer = 1)` 355

3.18.38 `line(x1 as Double, y1 as Double, x2 as Double, y2 as Double, colorValue as Integer, lineWidth as Integer = 1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draw a straight line.

Notes:

Argument	Default	Description
x1	(Mandatory)	The x coordinate of the first end-point of the line.
y1	(Mandatory)	The y coordinate of the first end-point of the line.
x2	(Mandatory)	The x coordinate of the second end-point of the line.
y2	(Mandatory)	The y coordinate of the second end-point of the line.
c	(Mandatory)	The color of the line.
lineWidth	1	The line width (thickness).

See also:

- 3.18.37 `line(x1 as Double, y1 as Double, x2 as Double, y2 as Double, colorValue as color, lineWidth as Integer = 1)` 355

3.18.39 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `linearGradientColor` method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.40 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer` 356
- 3.18.41 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer` 358
- 3.18.42 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer` 358

3.18.40 `linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a multi-point linear gradient color.

Notes: In this method, the color points are defined as an array of positions and colors along a reference line segment, in the following format:

position0, color0, position1, color1, positionN, colorN

The positions are specified as a number from 0 - 256 (0 - 100 in hex), in which 0 represents the starting point

of the reference line segment, and 256 (100 in hex) represents the ending point of the reference line segment.

For example, the array (in hex):

```
000000, FF0000, 000080, FFFF00, 000100, 00FF00
```

means the starting point (000000) is red (FF0000), the mid-point (000080 in hex) is yellow (FFFF00), and the ending point (000100 in hex) is green (00FF00).

One common usage of multi-point gradient colors is to define colors that have metallic look and feel. Chart-Director comes from several predefined gradient color arrays as follows.

NameValue (in Hex)

```
goldGradient      000000, FFE743, 000060, FFFF00, 0000B0, FFF0B0, 000100, FFE743
silverGradient    000000, C8C8C8, 000060, F8F8F8, 0000B0, E0E0E0, 000100, C8C8C8
redMetalGradient  000000, E09898, 000060, FFF0F0, 0000B0, F0D8D8, 000100, E09898
greenMetalGradient 000000, 98E098, 000060, F0FFF0, 0000B0, D8F0D8, 000100, 98E098
blueMetalGradient 000000, 9898E0, 000060, F0F0FF, 0000B0, D8D8F0, 000100, 9898E0
```

Argument	Default	Description
startX	(Mandatory)	The x coordinate of the starting point of the reference gradient line segment.
startY	(Mandatory)	The y coordinate of the starting point of the reference gradient line segment.
endX	(Mandatory)	The x coordinate of the ending point of the reference gradient line segment.
endY	(Mandatory)	The y coordinate of the ending point of the reference gradient line segment.
colorArray	(Mandatory)	An array defining the positions and colors of the pixels along the reference gradient line segment.
periodic	false	Specifies whether the gradient will repeat itself periodically. If the gradient does not repeat itself, the points that lie beyond the end points of the gradient line segment will assume the colors of the end points.

Return Value

A 32-bit integer representing the linear gradient color.

See also:

- 3.18.39 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer 356
- 3.18.41 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer 358
- 3.18.42 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer 358

3.18.41 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other linearGradientColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.39 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer 356
- 3.18.40 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer 356
- 3.18.42 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer 358

3.18.42 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a two-point linear gradient color.

Notes:

Argument	Default	Description
startX	(Mandatory)	The x coordinate of the starting point of the reference gradient line segment.
startY	(Mandatory)	The y coordinate of the starting point of the reference gradient line segment.
endX	(Mandatory)	The x coordinate of the ending point of the reference gradient line segment.
endY	(Mandatory)	The y coordinate of the ending point of the reference gradient line segment.
startColor	(Mandatory)	The color at the starting point of the reference gradient line segment.
endColor	(Mandatory)	The color at the ending point of the reference gradient line segment.
periodic	false	Specifies whether the gradient will repeat itself periodically. If the gradient does not repeat itself, the points that lie beyond the end points of the gradient line segment will assume the colors of the end points.

Return Value

A 32-bit integer representing the linear gradient color.

See also:

- 3.18.39 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as color, periodic as boolean=false) as Integer 356
- 3.18.40 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, colors() as Integer, periodic as boolean=false) as Integer 356

- 3.18.41 linearGradientColor(startX as Integer, startY as Integer, endX as Integer, endY as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer 358

3.18.43 load(path as string) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load an image file into the current DrawArea.

Notes: This method will overwrite the current DrawArea. The image type is determined based on file extension, which must be png, jpg/jpeg, gif or wbmp/wmp.

Argument	Default	Description
filename	(Mandatory)	The filename of the image to be loaded.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

3.18.44 loadData(data as MemoryBlock, ImgType as Integer = -1) as boolean

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads image from data in MemoryBlock.

Notes: Optionally provide image type.

Returns true on success or false on failure.

See also:

- 3.18.45 loadData(data as string, ImgType as Integer = -1) as boolean 359

3.18.45 loadData(data as string, ImgType as Integer = -1) as boolean

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads image from data in String.

Notes: Optionally provide image type.

Returns true on success or false on failure.

See also:

- 3.18.44 loadData(data as MemoryBlock, ImgType as Integer = -1) as boolean 359

3.18.46 loadGIF(path as string) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load a GIF image into the current DrawArea.

Notes: This method will overwrite the current DrawArea.

Argument	Default	Description
filename	(Mandatory)	The filename of the image to be loaded.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

3.18.47 loadJPG(path as string) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load a JPG image into the current DrawArea.

Notes: This method will overwrite the current DrawArea.

Argument	Default	Description
filename	(Mandatory)	The filename of the image to be loaded.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

3.18.48 loadPNG(path as string) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load a PNG image into the current DrawArea.

Notes: This method will overwrite the current DrawArea.

Argument	Default	Description
filename	(Mandatory)	The filename of the image to be loaded.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

3.18.49 loadWMP(path as string) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load a WAP bitmap image into the current DrawArea.

Notes: This method will overwrite the current DrawArea.

Argument	Default	Description
filename	(Mandatory)	The filename of the image to be loaded.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

3.18.50 merge(d as CDDrawAreaMBS, x as Integer, y as Integer, align as Integer, transparency as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copy another DrawArea to the current DrawArea.

Notes:

Argument	Default	Description
d	(Mandatory)	A DrawArea object representing the source.
x	(Mandatory)	The x coordinate of a reference point in the current DrawArea.
y	(Mandatory)	The y coordinate of a reference point in the current DrawArea.
align	(Mandatory)	The alignment of the source DrawArea relative to the reference point. See Alignment Specification for supported alignment types.
transparency	(Mandatory)	Specify the transparency level when copying the other DrawArea to the current DrawArea. A value of 0 means non-transparent. A value of 255 means totally transparent.

3.18.51 move(xOffset as Double, yOffset as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other move method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.52 move(xOffset as Double, yOffset as Double, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 362

3.18.52 `move(xOffset as Double, yOffset as Double, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Moves the drawing surface.

Notes:

Argument	Default	Description
<code>xOffset</code>	(Mandatory)	The pixel offset to move towards the left.
<code>yOffset</code>	(Mandatory)	The pixel offset to move towards the bottom.
<code>bgColor</code>	FFFFFF	The background color used to fill the space left after moving.
<code>filter</code>	LinearFilter	The filter to use for re-sampling. (Only applies for fractional pixel offsets.)
<code>blur</code>	1	The blur factor to use for re-sampling. (Only applies for fractional pixel offsets.)

See also:

- 3.18.51 `move(xOffset as Double, yOffset as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)` 361

3.18.53 `out(file as folderitem) as boolean`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea to an image file.

Notes: The file format is determined based on file extension, which must be png, jpg/jpeg, gif, wbmp/wmp or bmp.

Argument	Default	Description
<code>filename</code>	(Mandatory)	The filename of the output image file.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

3.18.54 `outBMP as string`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea as a BMP image to memory.

See also:

- 3.18.55 `outBMP(file as folderitem) as boolean`

3.18.55 outBMP(file as folderitem) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea to a BMP image file.

Notes:

Argument	Default	Description
filename	(Mandatory)	The filename of the output image file.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

See also:

- 3.18.54 outBMP as string

362

3.18.56 outGIF as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea as a GIF image to memory.

See also:

- 3.18.57 outGIF(file as folderitem) as boolean

363

3.18.57 outGIF(file as folderitem) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea to a GIF image file.

Notes:

Argument	Default	Description
filename	(Mandatory)	The filename of the output image file.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

See also:

- 3.18.56 outGIF as string

363

3.18.58 outJPG(file as folderitem, quality as Integer = 80) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea to a JPEG image file.

Notes:

Argument	Default	Description
filename	(Mandatory)	The filename of the output image file.
quality	80	The quality of the image.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

See also:

- 3.18.59 outJPG(quality as Integer = 80) as string 364

3.18.59 outJPG(quality as Integer = 80) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea as a JPEG image to memory.

Notes:

Argument	Default	Description
quality	80	The quality of the image.

Return Value

A binary string containing the JPEG image.

See also:

- 3.18.58 outJPG(file as folderitem, quality as Integer = 80) as boolean 363

3.18.60 outPDF as string

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea as a PDF image to memory.

See also:

- 3.18.61 outPDF(file as folderitem) as boolean 364

3.18.61 outPDF(file as folderitem) as boolean

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea to a PDF image file.

Notes: A return value being true indicates no error. A false value indicates the operation is unsuccessful.

See also:

- 3.18. CLASS CDDRAWAREAMBS 365
- 3.18.60 outPDF as string 364

3.18.62 outPicture as picture

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea into a picture.

Notes: Returns nil on any error.

3.18.63 outPNG as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea as a PNG image to memory.

See also:

- 3.18.64 outPNG(file as folderitem) as boolean 365

3.18.64 outPNG(file as folderitem) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea to a PNG image file.

Notes:

Argument	Default	Description
filename	(Mandatory)	The filename of the output image file.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

See also:

- 3.18.63 outPNG as string 365

3.18.65 outSVG(file as folderitem, options as string = "") as boolean

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea to a SVG or SVGZ image file.

Notes: To output true vector graphics in SVG or SVGZ format, please ensure DrawArea.enableVectorOutput is called immediately after creating the DrawArea object. Otherwise the output will be a bitmap image embedded in SVG or SVGZ.

Arguments:

Argument	Default	Description
file	(Mandatory)	The file of the output image file.
options	""	A text string specifying optional parameters for the SVG output. Currently, the only supported text string is "compress", which means to create a SVGZ (compressed SVG) instead of a regular SVG.

See also:

- 3.18.66 outSVG(options as string = "") as string 366

3.18.66 outSVG(options as string = "") as string

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea as a SVG or SVGZ image to memory.

Notes: To output true vector graphics in SVG or SVGZ format, please ensure DrawArea.enableVectorOutput is called immediately after creating the DrawArea object. Otherwise the output will be a bitmap image embedded in SVG or SVGZ.

Arguments:

Argument	Default	Description
options	""	A text string specifying optional parameters for the SVG output. Currently, the only supported text string is "compress", which means to create a SVGZ (compressed SVG) instead of a regular SVG.

Return Value

A memory block containing the SVG or SVGZ image.

See also:

- 3.18.65 outSVG(file as folderitem, options as string = "") as boolean 365

3.18.67 outWMP as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea as a WAP bitmap image to memory.

See also:

- 3.18.68 outWMP(file as folderitem) as boolean 367

3.18.68 outWMP(file as folderitem) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Writes the DrawArea to a WAP bitmap image file.

Notes:

Argument	Default	Description
filename	(Mandatory)	The filename of the output image file.

Return Value

A true value indicates no error. A false value indicates the operation is unsuccessful.

See also:

- 3.18.67 outWMP as string 366

3.18.69 patternColor(colors() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other patternColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.70 patternColor(colors() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer 367
- 3.18.71 patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer 368

3.18.70 patternColor(colors() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a pattern color using an array of colors as the bitmap pattern.

Notes: A pattern color is a dynamic color that changes according to a 2D periodic pattern. When it is used to fill an area, the area will look like being tiled with a wallpaper pattern.

Return Value

A 32-bit integer representing the pattern color.

See also:

- 3.18.69 patternColor(colors() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer 367
- 3.18.71 patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer 368

Argument	Default	Description
colorArray	(Mandatory)	An array of colors representing the colors of the bitmap pixels. The color of the pixel at (x, y) should correspond to index (x + y * width - 1) of the array.
height	(Mandatory)	The height of the bitmap in pixels. (The width is automatically computed as the size of the color array divided by the height.)
startX	0	The x coordinate of a reference point to align with the top-left corner the pattern.
startY	0	The y coordinate of a reference point to align with the top-left corner the pattern.

3.18.71 `patternColor(file as folderitem, startX as Integer = 0, startY as Integer = 0) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a pattern color by loading the pattern from an image file.

Notes: A pattern color is a dynamic color that changes according to a 2D periodic pattern. When it is used to fill an area, the area will look like being tiled with a wallpaper pattern.

ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to `DrawArea.setSearchPath` on the directory that ChartDirector will search for the file.

Argument	Default	Description
filename	(Mandatory)	An image file providing the pattern.
startX	0	The x coordinate of a reference point to align with the top-left corner the pattern.
startY	0	The y coordinate of a reference point to align with the top-left corner the pattern.

Return Value

A 32-bit integer representing the pattern color.

See also:

- 3.18.69 `patternColor(colors() as color, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer` 367
- 3.18.70 `patternColor(colors() as Integer, height as Integer, startX as Integer = 0, startY as Integer = 0) as Integer` 367

3.18.72 Pixel(x as Integer, y as Integer, c as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draw a pixel.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of the pixel.
y	(Mandatory)	The y coordinate of the pixel.
c	(Mandatory)	The color of the pixel.

3.18.73 polygon(x() as Double, y() as Double, edgeColor as color, fillColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other polygon method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.74 polygon(x() as Double, y() as Double, edgeColor as Integer, fillColor as Integer) 369
- 3.18.75 polygon(x() as Integer, y() as Integer, edgeColor as color, fillColor as color) 370
- 3.18.76 polygon(x() as Integer, y() as Integer, edgeColor as Integer, fillColor as Integer) 370

3.18.74 polygon(x() as Double, y() as Double, edgeColor as Integer, fillColor as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a polygon.

Notes:

Argument	Default	Description
x	(Mandatory)	An array of numbers representing the x coordinates of the vertices of a polygon.
y	(Mandatory)	An array of numbers representing the y coordinates of the vertices of a polygon.
edgeColor	(Mandatory)	The border color. To disable border, set the edgeColor the same as the fillColor.
fillColor	(Mandatory)	The fill color. To disable filling, set the fillColor to Transparent.

See also:

- 3.18.73 polygon(x() as Double, y() as Double, edgeColor as color, fillColor as color) 369

- 3.18.75 `polygon(x() as Integer, y() as Integer, edgeColor as color, fillColor as color)` 370
- 3.18.76 `polygon(x() as Integer, y() as Integer, edgeColor as Integer, fillColor as Integer)` 370

3.18.75 `polygon(x() as Integer, y() as Integer, edgeColor as color, fillColor as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `polygon` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.18.73 `polygon(x() as Double, y() as Double, edgeColor as color, fillColor as color)` 369
- 3.18.74 `polygon(x() as Double, y() as Double, edgeColor as Integer, fillColor as Integer)` 369
- 3.18.76 `polygon(x() as Integer, y() as Integer, edgeColor as Integer, fillColor as Integer)` 370

3.18.76 `polygon(x() as Integer, y() as Integer, edgeColor as Integer, fillColor as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a polygon.

Notes:

<code>x</code>	(Mandatory)	An array of numbers representing the x coordinates of the vertices of a polygon.
<code>y</code>	(Mandatory)	An array of numbers representing the y coordinates of the vertices of a polygon.
<code>edgeColor</code>	(Mandatory)	The border color. To disable border, set the <code>edgeColor</code> the same as the <code>fillColor</code> .
<code>fillColor</code>	(Mandatory)	The fill color. To disable filling, set the <code>fillColor</code> to <code>Transparent</code> .

none

See also:

- 3.18.73 `polygon(x() as Double, y() as Double, edgeColor as color, fillColor as color)` 369
- 3.18.74 `polygon(x() as Double, y() as Double, edgeColor as Integer, fillColor as Integer)` 369
- 3.18.75 `polygon(x() as Integer, y() as Integer, edgeColor as color, fillColor as color)` 370

3.18.77 `polyShape(xy() as integer, edgeColor as color, fillColor as color)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a shape.

Notes: A simple shape can be a polygon or an ellipse. A polygon can be specified as an array of numbers

(x0, y0, x1, y1,) representing the coordinates of the polygon vertices. As a polygon must have at least 3 vertices, the array must contain at least 6 numbers. If it contains only 4 numbers, it is will considered as an ellipse, with the numbers (x, y, rx, ry) interpreted as the coordinates of the center and the horizontal and vertical radii.

A complex shape is composed of multiple simple shapes. For example, a donut is a complex shape composed of two concentric circles. The shape is filled using the even-odd rule. (The web should have a lot of explanation on what is the even-odd rule in graphics.) A complex shape can be specified by including multiple simple shapes in the array, using the NewShape constant to separate them.

Argument	Default	Description
xy	(Mandatory)	An array of integers specifying the shape.
edgeColor	(Mandatory)	The border color. To disable border, set the edgeColor the same as the fillColor.
fillColor	(Mandatory)	The fill color. To disable filling, set the fillColor to Transparent.

See also:

- 3.18.78 polyShape(xy() as integer, edgeColor as Integer, fillColor as Integer) 371

3.18.78 polyShape(xy() as integer, edgeColor as Integer, fillColor as Integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a shape.

Notes: A simple shape can be a polygon or an ellipse. A polygon an be specified as an array of numbers (x0, y0, x1, y1,) representing the coordinates of the polygon vertices. As a polygon must have at least 3 vertices, the array must contain at least 6 numbers. If it contains only 4 numbers, it is will considered as an ellipse, with the numbers (x, y, rx, ry) interpreted as the coordinates of the center and the horizontal and vertical radii.

A complex shape is composed of multiple simple shapes. For example, a donut is a complex shape composed of two concentric circles. The shape is filled using the even-odd rule. (The web should have a lot of explanation on what is the even-odd rule in graphics.) A complex shape can be specified by including multiple simple shapes in the array, using the NewShape constant to separate them.

Argument	Default	Description
xy	(Mandatory)	An array of integers specifying the shape.
edgeColor	(Mandatory)	The border color. To disable border, set the edgeColor the same as the fillColor.
fillColor	(Mandatory)	The fill color. To disable filling, set the fillColor to Transparent.

See also:

- 3.18.77 polyShape(xy() as integer, edgeColor as color, fillColor as color) 370

3.18.79 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, data() as Integer, periodic as boolean=false) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a multi-point radial gradient color.

Notes: In this method, the color points are defined as an array of radial distances and colors, in the following format:

distance0, color0, distance1, color1, distanceN, colorN

The distances are specified as a number from 0 - 256 (0 - 100 in hex), in which 0 represents the center of the gradient defining ellipse, and 256 (100 in hex) represents the perimeter of the gradient defining ellipse.

For example, the array (in hex):

000000, FF0000, 000080, FFFF00, 000100, 00FF00

means the center (000000) is red (FF0000), the mid-point (000080 in hex) is yellow (FFFF00), and the perimeter (000100 in hex) is green (00FF00).

Argument	Default	Description
cx	(Mandatory)	The x coordinate of the center of the radial gradient.
cy	(Mandatory)	The y coordinate of the center of the radial gradient.
rx	(Mandatory)	The horizontal radius of the radial gradient defining ellipse.
ry	(Mandatory)	The vertical radius of the radial gradient defining ellipse.
colorArray	(Mandatory)	An array defining the radial distances and colors.
periodic	false	Specifies whether the gradient will repeat itself periodically. If the gradient does not repeat itself, the points that lie outside the gradient defining ellipse will assume the color at the perimeter of the gradient defining ellipse.

Return Value

A 32-bit integer representing the radial gradient color.

See also:

- 3.18.80 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer 372
- 3.18.81 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer 373

3.18.80 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other radialGradientColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.79 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, data() as Integer, periodic as boolean=false) as Integer 372
- 3.18.81 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer 373

3.18.81 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as Integer, endColor as Integer, periodic as boolean=false) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a two-point radial gradient color.

Notes:

Argument	Default	Description
cx	(Mandatory)	The x coordinate of the center of the radial gradient.
cy	(Mandatory)	The y coordinate of the center of the radial gradient.
rx	(Mandatory)	The horizontal radius of the radial gradient defining ellipse.
ry	(Mandatory)	The vertical radius of the radial gradient defining ellipse.
startColor	(Mandatory)	The color at the center of the gradient defining ellipse.
endColor	(Mandatory)	The color at the perimeter of the gradient defining ellipse.
periodic	false	Specifies whether the gradient will repeat itself periodically. If the gradient does not repeat itself, the points that lie outside the gradient defining ellipse will assume the color at the perimeter of the gradient defining ellipse.

Return Value

A 32-bit integer representing the radial gradient color.

See also:

- 3.18.79 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, data() as Integer, periodic as boolean=false) as Integer 372
- 3.18.80 radialGradientColor(cx as Integer, cy as Integer, rx as Integer, ry as Integer, startColor as color, endColor as color, periodic as boolean=false) as Integer 372

3.18.82 rAffineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `rAffineTransform` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.18.83 `rAffineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)` 374

3.18.83 `rAffineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Perform reverse affine transformation on the drawing surface.

Notes: A reverse affine transformation is an operation where every pixel is copied from another pixel according to the formula.

$xSrc = a * xDest + b * yDest + c$
 $ySrc = d * xDest + e * yDest + f$
 where $(xDest, yDest)$ is a destination pixel, and $(xSrc, ySrc)$ is where it should come from.

Many graphics operation, such as translation, rotation, and resizing, can be considered as a special case of reverse affine transformation.

Argument	Default	Description
a	(Mandatory)	The parameter 'a' in the coordinate transformation formula " $xSrc = a * xDest + b * yDest + c$ ".
b	(Mandatory)	The parameter 'b' in the coordinate transformation formula " $xSrc = a * xDest + b * yDest + c$ ".
c	(Mandatory)	The parameter 'c' in the coordinate transformation formula " $xSrc = a * xDest + b * yDest + c$ ".
d	(Mandatory)	The parameter 'd' in the coordinate transformation formula " $ySrc = d * xDest + e * yDest + f$ ".
e	(Mandatory)	The parameter 'e' in the coordinate transformation formula " $ySrc = d * xDest + e * yDest + f$ ".
f	(Mandatory)	The parameter 'f' in the coordinate transformation formula " $ySrc = d * xDest + e * yDest + f$ ".
bgColor	FFFFFF	The background color used to fill destination pixels that are not mapped to any source pixels.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

See also:

- 3.18.82 `rAffineTransform(a as Double, b as Double, c as Double, d as Double, e as Double, f as Double,`

bgColor as color, filter as Integer = 2, blur as Double = 1.0)

373

3.18.84 rect(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, edgeColor as color, fillColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other rect method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.85 rect(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, edgeColor as Integer, fillColor as Integer, raisedEffect as Integer = 0) 375

3.18.85 rect(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, edgeColor as Integer, fillColor as Integer, raisedEffect as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a rectangle.

Notes:

Argument	Default	Description
x1	(Mandatory)	The x coordinate of one corner of the rectangle.
y1	(Mandatory)	The y coordinate of one corner of the rectangle.
x2	(Mandatory)	The x coordinate of the opposite corner of the rectangle.
y2	(Mandatory)	The y coordinate of the opposite corner of the rectangle.
edgeColor	(Mandatory)	The border color. To disable border, set the edgeColor the same as the fillColor.
fillColor	(Mandatory)	The fill color. To disable filling, set the fillColor to Transparent.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument is also used to support CD-BaseChartMBS.glassEffect and CD-BaseChartMBS.softLighting effects.

See also:

- 3.18.84 rect(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer, edgeColor as color, fillColor as color, raisedEffect as Integer = 0) 375

3.18.86 reduceColors(colorCount as Integer, blackAndWhite as boolean=false) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reduces the number of colors in the image.

Notes: The colors in the image will be reduced to at most the number of colors specified, which should be 16 - 256. You may also set the image to black and white. In this case, the color count means number of grey levels.

The colors will be reduced by computing an optimal palette for the image. The image will then be converted using the palette based on the current dithering settings (see `DrawArea.setDitherMethod`).

Argument	Default	Description
<code>colorCount</code>	(Mandatory)	The maximum number of colors that image should have after after reduction. Must be between 16 - 256.
<code>blackAndWhite</code>	false	A true value means the image will be converted to black and white (with grey levels). A false value means the image will not be converted to black and white.

Return Value

The actual number of colors the converted image has.

3.18.87 `removeDynamicLayer(keepOriginal as boolean = false)`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Removes the dynamic layer if any.

3.18.88 `resize(newWidth as Integer, newHeight as Integer, filter as Integer = 1, blur as Double = 1.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Resize the drawing surface.

Notes:

Argument	Default	Description
<code>newWidth</code>	(Mandatory)	The new width of the drawing surface in pixels.
<code>newHeight</code>	(Mandatory)	The new height of the drawing surface in pixels.
<code>filter</code>	LinearFilter	The filter to use for re-sampling.
<code>blur</code>	1	The blur factor to use for re-sampling.

3.18.89 `ring(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, edgeColor as color, fillColor as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other ring method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.90 ring(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, edgeColor as Integer, fillColor as Integer) 377

3.18.90 ring(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, edgeColor as Integer, fillColor as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a ring.

See also:

- 3.18.89 ring(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, edgeColor as color, fillColor as color) 376

3.18.91 ringSector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, a1 as Double, a2 as Double, edgeColor as color, fillColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other ringSector method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.92 ringSector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, a1 as Double, a2 as Double, edgeColor as Integer, fillColor as Integer) 377

3.18.92 ringSector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, a1 as Double, a2 as Double, edgeColor as Integer, fillColor as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a ring sector.

See also:

- 3.18.91 ringSector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, rx2 as Integer, ry2 as Integer, a1 as Double, a2 as Double, edgeColor as color, fillColor as color) 377

3.18.93 rotate(angle as Double, bgColor as color, cx as Double = -1, cy as Double = -1, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other rotate method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.94 rotate(angle as Double, bgColor as Integer = &hFFFFFF, cx as Double = -1, cy as Double = -1, filter as Integer = 2, blur as Double = 1.0) 378

3.18.94 rotate(angle as Double, bgColor as Integer = &hFFFFFF, cx as Double = -1, cy as Double = -1, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Rotate the drawing surface.

Notes:

Argument	Default	Description
angle	(Mandatory)	The rotation angle measured clockwise in degrees.
bgColor	FFFFFF	The background color used to fill the space left after rotation.
cx	-1	The x coordinate of the center of rotation. -1 means rotating about the center of the drawing surface.
cy	-1	The y coordinate of the center of rotation. -1 means rotating about the center of the drawing surface.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

See also:

- 3.18.93 rotate(angle as Double, bgColor as color, cx as Double = -1, cy as Double = -1, filter as Integer = 2, blur as Double = 1.0) 378

3.18.95 sector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, edgeColor as color, fillColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other sector method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.96 sector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, edgeColor as Integer, fillColor as Integer) 379

3.18.96 `sector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, edgeColor as Integer, fillColor as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a circular or elliptical sector.

Notes:

Argument	Default	Description
<code>cx</code>	(Mandatory)	The x coordinate of the center of the circle or ellipse.
<code>cy</code>	(Mandatory)	The y coordinate of the center of the circle or ellipse.
<code>rx</code>	(Mandatory)	The horizontal radius of the circle or ellipse.
<code>ry</code>	(Mandatory)	The vertical radius of the circle or ellipse.
<code>a1</code>	(Mandatory)	The start angle of the sector in degrees. The angle is measured clockwise, with 0 degree being the upward pointing direction.
<code>a2</code>	(Mandatory)	The end angle of the sector in degrees. The angle is measured clockwise, with 0 degree being the upward pointing direction.
<code>edgeColor</code>	(Mandatory)	The border color. To disable border, set the <code>edgeColor</code> the same as the <code>fillColor</code> .
<code>fillColor</code>	(Mandatory)	The fill color. To disable filling, set the <code>fillColor</code> to <code>Transparent</code> .

See also:

- 3.18.95 `sector(cx as Integer, cy as Integer, rx as Integer, ry as Integer, a1 as Double, a2 as Double, edgeColor as color, fillColor as color)` 378

3.18.97 `setAntiAlias(shapeAntiAlias as boolean=true, textAntiAlias as Integer = 2)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Controls whether anti-alias is used when drawing lines, shapes and text.

Notes: For anti-aliasing text, `ChartDirector` supports the following modes.

ConstantValueDescription

<code>NoAntiAlias</code>	0	Disable anti-alias when drawing text
<code>AntiAlias</code>	1	Always use anti-alias when drawing text
<code>AutoAntiAlias</code>	2	Automatically determine if anti-alias should be used for the text. This is the default.

Currently, `ChartDirector` will anti-alias only large or bold fonts. For small fonts, assuming it is of high quality, anti-alias is unnecessary. It is because high quality fonts are normally designed to be sharp and clear at low resolution. Anti-aliasing will blur the fonts and make them look worse.

However, for complicated fonts (e.g. some fonts with oriental characters), or for lower quality fonts (e.g. some freeware fonts), anti-alias may be necessary. In this case, it may be needed to force anti-aliasing of all fonts using AntiAlias mode.

Argument	Default	Description
shapeAntiAlias	true	A true value enables anti-alias when drawing lines and shapes. A false value disables anti-alias when drawing lines and shapes
textAntiAlias	AutoAntiAlias	The text anti-alias mode, which must be one of AutoAntiAlias, AntiAlias or NoAntiAlias.

3.18.98 setAntiAliasText(value as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Set how text is antialiased.

3.18.99 setBgColor(c as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setBgColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.100 setBgColor(c as Integer) 380

3.18.100 setBgColor(c as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the background color of the image.

Notes:

Argument	Default	Description
c	(Mandatory)	The background color of the image.

See also:

- 3.18.99 setBgColor(c as color) 380

3.18.101 setClipRect(left as Integer, top as Integer, right as Integer, bottom as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the clip rectangle.

Notes: When performing graphics operation, only regions inside the clip rectangle will be affected. Regions outside will be unaffected.

Argument	Default	Description
left	(Mandatory)	The x coordinate of the left side of the clip rectangle.
top	(Mandatory)	The y coordinate of the top side of the clip rectangle.
right	(Mandatory)	The x coordinate of the right side of the clip rectangle.
bottom	(Mandatory)	The y coordinate of the bottom side of the clip rectangle.

3.18.102 setColorTable(colors() as color, offset as Integer)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setColorTable method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.103 setColorTable(colors() as Integer, offset as Integer)

3.18.103 setColorTable(colors() as Integer, offset as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Change the colors in the palette starting with the specified offset position.

Notes: See Color Specification on how colors are represented in ChartDirector.

Argument	Default	Description
colors	(Mandatory)	An array of colors to replace the colors in the palette.
offset	(Mandatory)	The position in the palette to start the replacement.

See also:

- 3.18.102 setColorTable(colors() as color, offset as Integer)

381

3.18.104 setDefaultFonts(normal as string, bold as string, italic as string, boldItalic as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the defaults for normal, bold, italic and bold-italic fonts.

Notes: See Font Specification for details on various font attributes.

Argument	Default	Description
normal	(Mandatory)	The default normal font. This is the same as the first font in the font table.
bold	""	The default bold font. This is the same as the second font in the font table. An empty string means the default is unchanged.
italic	""	The default italic font. This is the same as the third font in the font table. An empty string means the default is unchanged.
boldItalic	""	The default bold-italic font. This is the same as the fourth font in the font table. An empty string means the default is unchanged.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

3.18.105 setDitherMethod(value as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the dithering method in case dithering is necessary.

Notes: Dithering is a process of reducing the colors of an image. It is required if an image has more colors than can be supported by the image format. For example, a GIF image can only have 256 colors. If the

actual image contains more than 256 colors, dithering is needed to reduce the colors to less than 256.

The dithering method must be one of the following predefined constants.

ConstantValueDescription

Quantize	0	ChartDirector will first compute an optimal 256-color palette based on colors on the actual image. It then replaces the color of each pixel with the nearest color in the palette. This is the default method and produces the best result in most cases.
OrderedDither	1	Use the ordered dithering algorithm with a 4 x 4 matrix, and with the standard web-safe palette. The web-safe palette is a palette compatible with very old browsers (e.g. Netscape 1.x and 2.x browsers) on 256-color displays.

Web-safe palette is not an optimal palette. In most cases, this method is not as good as the Quantize method. Use this method only if you have to use the web-safe palette.

ErrorDiffusion	2	Similar to OrderedDither but use the Floyd and Steinberg error diffusion algorithm.
----------------	---	---

Argument	Default	Description
m	(Mandatory)	The dithering method to use in case dithering is necessary.

3.18.106 setFontTable(index as Integer, font as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets an entry in the font table to the specified font name.

Notes: The first 4 fonts in the font table have special significance. They are the defaults for normal, bold, italic and bold-italic fonts.

See Font Specification for details on various font attributes.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

Argument	Default	Description
index	(Mandatory)	An index to the font table, starting from 0.
font	(Mandatory)	The font name to be put into the font table.

3.18.107 setInterlace(value as boolean)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the interlace mode when creating the image.

Notes: Note that in many case an interlaced image is less compressible, and may have a large image size. The default is non-interlace.

Note that this method only applies to image formats that support interlacing (GIF and PNG). It is ignored for image formats that does not support interlacing.

Argument	Default	Description
i	(Mandatory)	A true value means the image is interlaced. A false value means the image is non-interlaced.

3.18.108 setOutputOptions(options as string)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets output format options for the next chart output. Currently, only SVG, SVGZ and PDF output formats support output options.

Notes: An output option can be a flag (such as "compress") or an attribute-value pair (such as "width=800"). Multiple output options can be joined using semicolons as delimiters.

SVG Options

PDF Options

The PDF viewer will convert the pixel unit into physical unit (eg. inches) so that it can be layout on paper or other physical media. The default conversion factor for the chart is 96 pixels per inch. The "dpi" attribute can be used to specify an alternative value. The value must be a number.

3.18.109 setPaletteMode(value as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

SVG Option	Type	Description
compress	Flag	Compressed the SVG, that is, output SVGZ.
bitmap	Flag	Render the chart as a bitmap and output the bitmap as SVG.
noxmldecl	Flag	Do not include the xml declaration line "<?xml version="1.0" >" in the SVG.
noddoctype	Flag	Do not include the document type declaration line "<!DOCTYPE svg PUBLIC >" in the SVG.
width	Flag / Attribute	Specifies the width attribute of the SVG. By default, ChartDirector will not include the width or height attribute in the SVG output. In this case, the SVG is variable in size and would assume the size of its container. For example, if the SVG is inside a <DIV>block in a web page, it will assume the size of the DIV block. If the "width" option is used as a flag, ChartDirector will include the width attribute in the SVG and set it to the chart width. If the "width" option is used as an attribute (such as "width=800"), ChartDirector will include the width attribute in the SVG and set it to the specified value. The specified value should be some text that is valid as SVG width. Examples are "100" and "75%".
height	Flag / Attribute	Specifies the height attribute of the SVG. See the description on "width" above on how to use it.

Function: Sets the palette mode to use when writing the image in PNG format.

Notes: The PNG format supports both palette based images and true color images. Palette based images can only have 256 colors, but is smaller in size.

The palette mode must be one of the following predefined constants.

ConstantValueDescription

3.18.110 setResource(id as string, data as MemoryBlock)

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns a resource ID to a memory block to allow it to be referenced using a resource path.

Notes: Many ChartDirector features, such as BaseChart.setBgImage and the <img*>tag in CDML, expect a file or resource path for loading an image. If the image happens to be in memory, such as if the image is retrieved from a database, setResource can be used to assign a resource ID to the memory. It can then be referenced using "@/res_id", in which res_id is the resource ID.

The assigned resource ID is only valid for the DrawArea object of which this method is called. Use CD-BaseChartMBS.setResourceGlobal to assign a resource ID that is valid for all ChartDirector objects.

This method stores only a pointer to the memory. It does not copy the memory. You must ensure the memory contains valid content for as long as the resource is being used.

PDF Option	Type	Description
bitmap	Flag	Render the chart as a bitmap and output the bitmap as PDF.
width	Attribute	The width of the chart in the PDF in pixel unit. By default, ChartDirector will use the pixel width of the chart as the width of the chart in PDF. The "width" attribute can be used to specify an alternative value. The value must be a number.
height	Attribute	The width of the chart in the PDF in pixel unit. See the description on "width" above for how to use it.
pagewidth	Attribute	The page width in pixel unit. By default, ChartDirector will set the page width to the same width as the chart. The "pagewidth" attribute can be used to specify an alternative value. The value must be a number.
pageheight	Attribute	The page height in pixel unit. By default, ChartDirector will set the page height to the same height as the chart. The "pageheight" attribute can be used to specify an alternative value. The value must be a number.
leftx	Attribute	The x coordinate of the left side of the chart within the page in pixel unit. By default, ChartDirector will center the chart in the page. The "leftx" attribute can be used to specify an alternative horizontal position. The coordinate must be a number.
topy	Attribute	The y coordinate of the top side of the chart within the page in pixel unit. By default, ChartDirector will center the chart in the page. The "topy" attribute can be used to specify an alternative vertical position. The coordinate must be a number.
dpi	Attribute	Specify the factor for conversion from pixel to physical unit.

Argument	Default	Description
options	(Mandatory)	A list of options delimited by semicolons.

While ChartDirector does not copy the data, the MBS Plugin will put the resources in a dictionary to make sure they stay available till the chart is destroyed and avoid a crash.

See also:

- 3.18.111 setResource(id as string, data as string) 386
- 3.18.112 setResource(id as string, drawArea as CDDrawAreaMBS) 387

3.18.111 setResource(id as string, data as string)

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns a resource ID to a memory block to allow it to be referenced using a resource path.

Notes: Many ChartDirector features, such as BaseChart.setBgImage and the <img*>tag in CDML, expect a file or resource path for loading an image. If the image happens to be in memory, such as if the image is retrieved from a database, setResource can be used to assign a resource ID to the memory. It can then be referenced using "@/res_id", in which res_id is the resource ID.

TryPalette	0	Use palette mode if the image contains less than 256 colors, otherwise use true color mode. This is the default.
ForcePalette	1	Use palette mode. If the image contains more than 256 colors, reduce it to 256 colors using dithering (see DrawArea.setDitherMethod).
NoPalette	2	Use true color mode.

Argument	Default	Description
p	(Mandatory)	The palette mode for PNG images.

The assigned resource ID is only valid for the DrawArea object of which this method is called. Use CD-BaseChartMBS.setResourceGlobal to assign a resource ID that is valid for all ChartDirector objects.

This method stores only a pointer to the memory. It does not copy the memory. You must ensure the memory contains valid content for as long as the resource is being used.

While ChartDirector does not copy the data, the MBS Plugin will put the resources in a dictionary to make sure they stay available till the chart is destroyed and avoid a crash.

See also:

- 3.18.110 setResource(id as string, data as MemoryBlock) 385
- 3.18.112 setResource(id as string, drawArea as CDDrawAreaMBS) 387

3.18.112 setResource(id as string, drawArea as CDDrawAreaMBS)

Plugin Version: 18.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns a resource ID to a DrawArea object to allow it to be referenced using a resource path.
Notes: This method is the same DrawArea.setResource, except that it assigns the resource ID to a DrawArea object instead of a memory image. This allows the DrawArea object to be reference as an image resource using "@/res_id".

While ChartDirector does not copy the draw area, the MBS Plugin will put the resources in a dictionary to make sure they stay available till the chart is destroyed and avoid a crash.

See also:

- 3.18.110 setResource(id as string, data as MemoryBlock) 385
- 3.18.111 setResource(id as string, data as string) 386

3.18.113 setSearchPath(path as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Argument	Default	Description
id	(Mandatory)	The resource ID to be used to reference the memory image.
img	(Mandatory)	The memory that the image occupies.

Argument	Default	Description
id	(Mandatory)	The resource ID to be used to reference the memory image.
img	(Mandatory)	The memory that the image occupies.

Function: Sets the file system search path for loading image files.

Notes: Several ChartDirector operations involve loading image files. Examples are wallpapers (`BaseChart.setWallpaper`), background images (`BaseChart.setBgImage` and `PlotArea.setBackground2`), user-defined symbols (`DataSet.setDataSymbol2`) or for embedding images in text using ChartDirector Mark Up Language.

3.18.114 `setSize(width as Integer, height as Integer, bgColor as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setSize` method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.115 `setSize(width as Integer, height as Integer, bgColor as Integer = &hFFFFFF)` 388

3.18.115 `setSize(width as Integer, height as Integer, bgColor as Integer = &hFFFFFF)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size and background color of the drawing surface.

Notes:

See also:

- 3.18.114 `setSize(width as Integer, height as Integer, bgColor as color)` 388

3.18.116 `setTransparentColor(value as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setTransparentColor` method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.117 `setTransparentColor(value as Integer)` 389

Argument	Default	Description
id	(Mandatory)	The resource ID to be used to reference the DrawArea object.
drawArea	(Mandatory)	The DrawArea object to be used as a resource.

Argument	Default	Description
path	(Mandatory)	A list of directories, separated with the path separator of your operating system (";" for Windows, ":" for Linux/UNIX).

3.18.117 setTransparentColor(value as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Specifies a certain color to mean transparent when creating the image output, or to include alpha transparency channel in the output.

Notes: There are two types of transparency:

Alpha transparency: In addition to red, green and blue levels, there is a transparency level associated with each pixel, which can range from completely transparent to completely opaque. The data associated with the transparency information is called the alpha channel.

Single color transparency: The image itself has no alpha channel, but a certain color is used to mean completely transparent. For internal drawing, ChartDirector always use alpha transparency. However, when outputting the image as an image file, ChartDirector by default will remove the alpha channel to reduce image size. It is because many image displaying software do not support alpha transparency. For example, the IE browser only supports single color transparency but not alpha transparency.

If you want to use single color transparency in the output, you may specify the transparent color as the argument to the setTransparentColor method. Note that only GIF and PNG can support single color transparency. JPEG, BMP and WBMP cannot support transparency at all.

If you do want to keep the alpha channel in final output, you may pass -1 as the argument to setTransparentColor. Note that the only image format that can support alpha transparency is PNG.

One important thing to note is that the IE browser (and possibly many image displaying software) only supports single color transparency for palette based images with up to 256 colors, but not for true color images. For this reason, if single color transparency is used, ChartDirector will automatically reduce the image to 256 colors if it has more than 256 colors. This may result in lost of image quality, especially if the image contains gradient colors.

Therefore, due to the limitations of the current generations of image displaying software, for highest image quality, sometimes it may be beneficial to not using transparency in image output, but to set the image background color to the same color as the container background.

Argument	Default	Description
width	(Mandatory)	The width of the drawing surface in pixels.
height	(Mandatory)	The height of the drawing surface in pixels.
bgColor	FFFFFF	The background color of the drawing surface.

Argument	Default	Description
c	(Mandatory)	The color that is designated as the transparent color. If -1 is used, the full alpha transparency channel will be included in the final output.

See also:

- 3.18.116 `setTransparentColor(value as color)` 388

3.18.118 `shearTransform(xShear as Double, yShear as Double = 0, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shear the drawing surface.

Notes: Shearing can be applied along the horizontal direction and/or vertical direction. A rectangle, after shearing, will become a parallelogram.

Argument	Default	Description
xShear	(Mandatory)	The number of pixels to shear leftwards.
yShear	0	The number of pixels to shear downwards.
bgColor	FFFFFF	The background color used to fill the space left after transformation.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

See also:

- 3.18.119 `shearTransform(xShear as Double, yShear as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)` 390

3.18.119 `shearTransform(xShear as Double, yShear as Double, bgColor as color, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other shearTransform method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.118 shearTransform(xShear as Double, yShear as Double = 0, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 390

3.18.120 sphereTransform(xDiameter as Integer, yDiameter as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other sphereTransform method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.121 sphereTransform(xDiameter as Integer, yDiameter as Integer, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 391

3.18.121 sphereTransform(xDiameter as Integer, yDiameter as Integer, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Wraps the drawing surface onto a sphere or ellipsoid.

Notes:

Argument	Default	Description
xDiameter	(Mandatory)	The horizontal diameter of the sphere or ellipsoid in pixels.
yDiameter	(Mandatory)	The vertical diameter of the sphere or ellipsoid in pixels.
bgColor	FFFFFF	The background color used to fill the space left after transformation.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

See also:

- 3.18.120 sphereTransform(xDiameter as Integer, yDiameter as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0) 391

3.18.122 surface(cx1 as Double, y1 as Double, x2 as Double, y2 as Double, depthX as Integer, depthY as Integer, edgeColor as color, fillColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other surface method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.123 `surface(cx1 as Double, y1 as Double, x2 as Double, y2 as Double, depthX as Integer, depthY as Integer, edgeColor as Integer, fillColor as Integer)` 392

3.18.123 `surface(cx1 as Double, y1 as Double, x2 as Double, y2 as Double, depthX as Integer, depthY as Integer, edgeColor as Integer, fillColor as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a parallelogram.

Notes:

Argument	Default	Description
x1	(Mandatory)	The x coordinate of the first end-point of one line of the parallelogram.
y1	(Mandatory)	The y coordinate of the first end-point of one line of the parallelogram.
x2	(Mandatory)	The x coordinate of the second end-point of one line of the parallelogram.
y2	(Mandatory)	The y coordinate of the second end-point of one line of the parallelogram.
depthX	(Mandatory)	The x displacement of the line segment that is parallel to the line segment above.
depthY	(Mandatory)	The y displacement of the line segment that is parallel to the line segment above.
edgeColor	(Mandatory)	The border color. To disable border, set the edgeColor the same as the fillColor.
fillColor	(Mandatory)	The fill color. To disable filling, set the fillColor to Transparent.

See also:

- 3.18.122 `surface(cx1 as Double, y1 as Double, x2 as Double, y2 as Double, depthX as Integer, depthY as Integer, edgeColor as color, fillColor as color)` 391

3.18.124 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean) as CDTTFFTextMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a TTFText object representing a text block.

Notes: The TTFText object contains methods to measure and draw the text. This method is useful if the size of the text is needed before deciding where to draw it.

Return Value

Argument	Default	Description
text	(Mandatory)	A string representing the text to be drawn. See ChartDirector Mark Up Language on how to embed special tags in the text for sophisticated formatting.
font	(Mandatory)	The font name. See Font Specification for details on various font attributes.
fontIndex	(Mandatory)	The font index in case the font name refers to a font collection. An index of 0 means the first font.
fontHeight	(Mandatory)	The font height in points.
fontWidth	(Mandatory)	The font width in points.
angle	(Mandatory)	The rotation angle of the text.
vertical	(Mandatory)	A true value means the text is layout vertically (from top to bottom). A false value means the is layout horizontally (from left to right).

The TTFText object created.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.18.125 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7)` 393
- 3.18.126 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7)` 394
- 3.18.127 `text(str as string, font as string, fontsize as Double) as CDTTTFTextMBS` 395
- 3.18.128 `text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as color)` 396
- 3.18.129 `text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as Integer)` 396

3.18.125 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other text method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.124 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean) as CDTTTFTextMBS` 392

- 3.18.126 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7)` 394
- 3.18.127 `text(str as string, font as string, fontsize as Double)` as `CDTTFTTextMBS` 395
- 3.18.128 `text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as color)` 396
- 3.18.129 `text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as Integer)` 396

3.18.126 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws text.

Notes:

Argument	Default	Description
<code>str</code>	(Mandatory)	A string representing the text to be drawn. See ChartDirector Mark Up Language on how to embed special tags in the text for sophisticated formatting.
<code>font</code>	(Mandatory)	The font name. See Font Specification for details on various font attributes.
<code>fontIndex</code>	(Mandatory)	The font index in case the font name refers to a font collection. An index of 0 means the first font.
<code>fontHeight</code>	(Mandatory)	The font height in points.
<code>fontWidth</code>	(Mandatory)	The font width in points.
<code>angle</code>	(Mandatory)	The rotation angle of the text. The angle is measured in degrees in clockwise direction.
<code>vertical</code>	(Mandatory)	A true value means the text is layout vertically (from top to bottom). A false value means the is layout horizontally (from left to right).
<code>x</code>	(Mandatory)	The x coordinate of a reference point to align the text.
<code>y</code>	(Mandatory)	The y coordinate of a reference point to align the text.
<code>color</code>	(Mandatory)	The color of the text.
<code>alignment</code>	<code>TopLeft</code>	The position of the text relative to the reference point. See Alignment Specification for supported alignment types.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.18.124 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean)` as `CDTTFTTextMBS` 392

- 3.18. CLASS CDDRAWAREAMBS 395
- 3.18.125 text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7) 393
- 3.18.127 text(str as string, font as string, fontsize as Double) as CDTTFTTextMBS 395
- 3.18.128 text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as color) 396
- 3.18.129 text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as Integer) 396

3.18.127 text(str as string, font as string, fontsize as Double) as CDTTFTTextMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a TTFText object representing a text block.

Notes: The TTFText object contains methods to measure and draw the text. This method is useful if the size of the text is needed before deciding where the draw it.

Argument	Default	Description
str	(Mandatory)	A string representing the text to be drawn. See ChartDirector Mark Up Language on how to embed special tags in the text for sophisticated formatting.
font	(Mandatory)	The font name. See Font Specification for details on various font attributes.
fontSize	(Mandatory)	The font size in points.

Return Value

The TTFText object created.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.18.124 text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean) as CDTTFTTextMBS 392
- 3.18.125 text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7) 393
- 3.18.126 text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7) 394
- 3.18.128 text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as color) 396

- 3.18.129 `text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as Integer)` 396

3.18.128 `text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other text method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.124 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean) as CDTTFTTextMBS` 392
- 3.18.125 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7)` 393
- 3.18.126 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7)` 394
- 3.18.127 `text(str as string, font as string, fontsize as Double) as CDTTFTTextMBS` 395
- 3.18.129 `text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as Integer)` 396

3.18.129 `text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws text.

Notes:

Argument	Default	Description
<code>str</code>	(Mandatory)	A string representing the text to be drawn. See ChartDirector Mark Up Language on how to embed special tags in the text for sophisticated formatting.
<code>font</code>	(Mandatory)	The font name. See Font Specification for details on various font attributes.
<code>fontSize</code>	(Mandatory)	The font size in points.
<code>x</code>	(Mandatory)	The x coordinate of the top-left corner of the text.
<code>y</code>	(Mandatory)	The y coordinate of the top-left corner of the text.
<code>color</code>	(Mandatory)	The color of the text.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.18. *CLASS CDDRAWAREAMBS* 397
- 3.18.124 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean) as CDTTFTTextMBS` 392
 - 3.18.125 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7)` 393
 - 3.18.126 `text(str as string, font as string, fontIndex as Integer, fontHeight as Double, fontWidth as Double, angle as Double, vertical as boolean, x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7)` 394
 - 3.18.127 `text(str as string, font as string, fontsize as Double) as CDTTFTTextMBS` 395
 - 3.18.128 `text(str as string, font as string, fontsize as Double, x as Integer, y as Integer, colorvalue as color)` 396

3.18.130 `tile(d as CDDrawAreaMBS, transparency as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copy another DrawArea to the current DrawArea as a wallpaper.

Notes: The wallpaper image will be tiled repeatedly on the current DrawArea until the entire DrawArea is covered.

Argument	Default	Description
<code>d</code>	(Mandatory)	A DrawArea object representing the source.
<code>transparency</code>	(Mandatory)	Specify the transparency level when copying the other DrawArea to the current DrawArea. A value of 0 means non-transparent. A value of 255 means totally transparent.

3.18.131 `vCylinderTransform(xDiameter as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `vCylinderTransform` method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.132 `vCylinderTransform(xDiameter as Integer, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)` 397

3.18.132 `vCylinderTransform(xDiameter as Integer, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Wraps the drawing surface onto a vertical cylinder.

Notes:

Argument	Default	Description
xDiameter	(Mandatory)	The diameter of the cylinder in pixels.
bgColor	FFFFFF	The background color used to fill the space left after transformation.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

See also:

- 3.18.131 vCylinderTransform(xDiameter as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0) 397

3.18.133 vFlip

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Flip the drawing surface along the central vertical line.

3.18.134 vline(y1 as Integer, y2 as Integer, x as Integer, c as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws a vertical line.

Notes:

Argument	Default	Description
y1	(Mandatory)	The y coordinate of the first end-point of the line.
y2	(Mandatory)	The y coordinate of the second end-point of the line.
x	(Mandatory)	The x coordinate of the line.
c	(Mandatory)	The color of the line.

3.18.135 vTriangleTransform(tHeight as Integer = -1, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Wraps the drawing surface onto a vertical triangle pointing upwards.

Notes:

See also:

Argument	Default	Description
tHeight	-1	The height of the triangle in pixels.
bgColor	FFFFFF	The background color used to fill the space left after transformation.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

- 3.18.136 `vTriangleTransform(tHeight as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0)` 399

3.18.136 `vTriangleTransform(tHeight as Integer, bgColor as color, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `vTriangleTransform` method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.135 `vTriangleTransform(tHeight as Integer = -1, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)` 398

3.18.137 `waveTransform(period as Integer, amplitude as Double, direction as Double = 0, startAngle as Double = 0, longitudinal as boolean=false, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Moves the pixels on the drawing surface according to a sinusoidal function to achieve a wave effect.

Notes:

Argument	Default	Description
period	(Mandatory)	The period of the wave in pixels.
amplitude	(Mandatory)	The amplitude of the wave in pixels.
direction	0	The propagation direction of the wave. The upward pointing direction is 0 degree, and the angle is measured clockwise.
startAngle	0	The initial phase angle of the wave in degrees.
longitudinal	false	Determine if the wave is transversal or longitudinal. true means transversal. false means longitudinal.
bgColor	FFFFFF	The background color used to fill the space left after transformation.
filter	LinearFilter	The filter to use for re-sampling.
blur	1	The blur factor to use for re-sampling.

See also:

- 3.18.138 `waveTransform`(period as Integer, amplitude as Double, direction as Double, startAngle as Double, longitudinal as boolean, bgColor as color, filter as Integer = 2, blur as Double = 1.0) 400

3.18.138 `waveTransform`(period as Integer, amplitude as Double, direction as Double, startAngle as Double, longitudinal as boolean, bgColor as color, filter as Integer = 2, blur as Double = 1.0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `waveTransform` method, but uses color instead of integer data type for passing color values.

See also:

- 3.18.137 `waveTransform`(period as Integer, amplitude as Double, direction as Double = 0, startAngle as Double = 0, longitudinal as boolean=false, bgColor as Integer = &hFFFFFF, filter as Integer = 2, blur as Double = 1.0) 399

3.18.139 Properties

3.18.140 Handle as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal object reference.

Notes: (Read only property)

3.18.141 Resources as Dictionary

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal resource dictionary.

Notes: Stores references for registered resources.

Just for debugging.

(Read only property)

3.19 class CDDrawObjMBS

3.19.1 class CDDrawObjMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: This is the base class for several drawing related subclasses.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.19.2 Methods

3.19.3 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.19.4 paint(d as CDDrawAreaMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws the content to this object into a drawarea.

3.19.5 setZOrder(z as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the z-order.

3.20 class CDFinanceChartMBS

3.20.1 class CDFinanceChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: FinanceChart is a financial charting library built on top of the main ChartDirector library.

Notes: It extends CDMultiChartMBS by implementing many financial indicators, and allowing complex financial charts to be composed easily.

Key Features:

- **Arbitrary Financial Chart Composition:** Allows a complex financial chart to be constructed by stacking a main financial chart with arbitrary number of indicator charts in any order.
- **Main Chart Layers:** The main chart can show information in Candlestick or OHLC, and/or various price lines, such as closing price line, weighted close line, typical price line or median price line.
- **Moving Averages:** Chart can include arbitrary number of moving average lines of configurable types and periods. Supports simple, exponential, triangular and weighted moving averages.
- **Parabolic SAR:** A parabolic SAR indicator can be added to the main price chart.
- **Bands and Envelops:** Built-in Bollinger Band layer, Donchian Channel layer, and moving average envelops. Also supports custom-defined bands.
- **Volume Bars:** Volume Bars can be displayed as part of the main chart, or as a separate indicator chart.
- **Axis Styles:** Axis can be log or linear scale, and can display on left or right side of the chart.
- **Built-In Indicators:** Accumulation/Distribution, Aroon Up/Down, Aroon Oscillator, Average Directional Index, Average True Range (including raw True Range), Bollinger Band Width, Commodity Channel Index, Chaikin Money Flow, Chaikin Oscillator, Chaikin Volatility, Close Location Value, Detrended Price Oscillator, Donchian Channel Width, Ease of Movement, Fast Stochastic, MACD, Mass Index, Money Flow Index, Momentum, Negative Volume Index, On Balance Volume, Performance, Percentage Price Oscillator, Positive Volume Index, Percentage Volume Oscillator, Price Volume Trend, Rate of Change, Relative Strength Index, Slow Stochastic, Standard Deviation, Stochastic RSI, TRIX, Ultimate Oscillator, Volume, William %R.
- **Custom Lines and Indicators:** Extensible design allows developers to implement additional lines and indicator types.
- **Customizable Look and Feel:** Preserves the flexibility and power of the ChartDirector API. Background images, custom text boxes, logos, fonts and colors, etc., are all customizable.

Using FinanceChart:

You create an instance of FinanceChart. Next, you may add data to it using setData. After that, you may add the main chart (addMainChart) and other indicator charts to display the data graphically. These charts

may be added in any order.

For the main chart, you may:

- Display price information using Candlesticks (`addCandleStick`) or HLOC (`addHLOC`) symbols
- Add various price lines (`addCloseLine`, `addWeightedClose`, `addTypicalPrice`, `addMedianPrice`)
- Add moving average lines of different types and periods (`addSimpleMovingAvg`, `addExpMovingAvg`, `addTriMovingAvg`, `addWeightedMovingAvg`)
- Add Parabolic SAR (`addParabolicSAR`)
- Add price bands or envelopes: (`addBollingerBand`, `addDonchianChannel`, `addEnvelop`)
- Add custom lines or bands (`addLineIndicator2`, `addBand`)
- Add volume bars (`addVolBars`)

For indicator charts, `FinanceChart` includes a wide variety of indicators. Please refer to the table below for details. You may also add your own custom indicator chart using `addLineIndicator` and `addBarIndicator`.

In addition to chart building methods, `FinanceChart` also has a number of chart formatting methods, such as `addPlotAreaTitle` to add chart title, `setPlotAreaStyle` to control background color and grid lines, `setDateLabelFormat` to control date/time formatting, etc.

Furthermore, as `FinanceChart` is a subclass of `CDMultiChartMBS`, and the main and indicator charts are implemented as `XYChart` objects, they can use the powerful `ChartDirector` API for formatting (eg. background images, custom logos, custom text boxes, metallic colors, etc).
Subclass of the `CDMultiChartMBS` class.

Blog Entries

- [Chart Diagrams with Xojo](#)
- [MBS Real Studio Plugins, version 12.1pr1](#)
- [MBS Real Studio Plugins, version 12.0pr7](#)

Xojo Developer Magazine

- [7.5, pages 34 to 35: Easy Charts and Graphs Part 2, Using the ChartDirector Plugin](#)
- [18.5, page 60: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.20.2 Methods

3.20.3 addAccDist(height as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addAccDist method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.4 addAccDist(height as Integer, ColorValue as Integer) as CDXYChartMBS 404

3.20.4 addAccDist(height as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an Accumulation/Distribution indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.3 addAccDist(height as Integer, ColorValue as color) as CDXYChartMBS 404

3.20.5 addADX(height as Integer, period as Integer, posColor as color, negColor as color, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addADX method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.6 addADX(height as Integer, period as Integer, posColor as Integer, negColor as Integer, ColorValue as Integer) as CDXYChartMBS 405

3.20.6 addADX(height as Integer, period as Integer, posColor as Integer, negColor as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an Average Directional Index indicators chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
posColor	(Mandatory)	The color of the Positive Directional Index line.
negColor	(Mandatory)	The color of the Negative Directional Index line.
color	(Mandatory)	The color of the Average Directional Index line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.5 addADX(height as Integer, period as Integer, posColor as color, negColor as color, ColorValue as color) as CDXYChartMBS 404

3.20.7 addAroon(height as Integer, period as Integer, upColor as color, downColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addAroon method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.8 addAroon(height as Integer, period as Integer, upColor as Integer, downColor as Integer) as CDXYChartMBS 405

3.20.8 addAroon(height as Integer, period as Integer, upColor as Integer, downColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an Aroon Up/Down indicators chart.

Notes: Arguments:

Return Value

A CDXYChartMBS object representing the chart created.

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicators.
upColor	(Mandatory)	The color of the Aroon Up indicator line.
downColor	(Mandatory)	The color of the Aroon Down indicator line.

See also:

- 3.20.7 addAroon(height as Integer, period as Integer, upColor as color, downColor as color) as CDXYChartMBS 405

3.20.9 addAroonOsc(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addAroonOsc method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.10 addAroonOsc(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 406

3.20.10 addAroonOsc(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an Aroon Oscillator indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.9 addAroonOsc(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS 406

3.20.11 addATR(height as Integer, period as Integer, color1 as color, color2 as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addATR method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.12 addATR(height as Integer, period as Integer, color1 as Integer, color2 as Integer) as CDXY-ChartMBS 407

3.20.12 addATR(height as Integer, period as Integer, color1 as Integer, color2 as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an Average True Range indicators chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color1	(Mandatory)	The color of the True Range line.
color2	(Mandatory)	The color of the Average True Range line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.11 addATR(height as Integer, period as Integer, color1 as color, color2 as color) as CDXY-ChartMBS 407

3.20.13 addBand(upperLine() as Double, lowerLine() as Double, LineColor as color, FillColor as color, Name as string) as CDInterLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBand method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.14 addBand(upperLine() as Double, lowerLine() as Double, LineColor as Integer, FillColor as Integer, Name as string) as CDInterLineLayerMBS 408

3.20.14 `addBand(upperLine() as Double, lowerLine() as Double, LineColor as Integer, FillColor as Integer, Name as string) as CDInterLineLayerMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a generic band to the main finance chart.

Notes: This method is used internally by other methods to add various bands (eg. Bollinger band, Donchian channels, etc).

Arguments:

Argument	Default	Description
<code>upperLine</code>	(Mandatory)	The data series for the upper band line.
<code>lowerLine</code>	(Mandatory)	The data series for the lower band line.
<code>lineColor</code>	(Mandatory)	The color of the upper and lower band line.
<code>fillColor</code>	(Mandatory)	The color to fill the region between the upper and lower band lines.
<code>name</code>	(Mandatory)	The name of the band.

Return Value

An `CDInterLineLayerMBS` object representing the filled region.

See also:

- 3.20.13 `addBand(upperLine() as Double, lowerLine() as Double, LineColor as color, FillColor as color, Name as string) as CDInterLineLayerMBS` 407

3.20.15 `addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as color, name as string) as CDBarLayerMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addBarIndicator` method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.16 `addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as Integer, name as string) as CDBarLayerMBS` 409
- 3.20.17 `addBarIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS` 409
- 3.20.18 `addBarIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS` 410

3.20.16 addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as Integer, name as string) as CDBarLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a bar layer to an existing indicator chart.

Notes: Arguments:

Argument	Default	Description
c	(Mandatory)	The indicator chart to add the bar layer to.
data	(Mandatory)	The data series of the indicator bars.
color	(Mandatory)	The color of the indicator bars.
name	(Mandatory)	The name of the indicator.

Return Value

A CDBarLayerMBS object representing the bar layer created.

See also:

- 3.20.15 addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as color, name as string) as CDBarLayerMBS 408
- 3.20.17 addBarIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS 409
- 3.20.18 addBarIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS 410

3.20.17 addBarIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBarIndicator method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.15 addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as color, name as string) as CDBarLayerMBS 408
- 3.20.16 addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as Integer, name as string) as CDBarLayerMBS 409
- 3.20.18 addBarIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS 410

3.20.18 addBarIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a generic bar indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
data	(Mandatory)	The data series of the indicator bars.
color	(Mandatory)	The color of the indicator bars.
name	(Mandatory)	The name of the indicator.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.15 addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as color, name as string) as CDBarLayerMBS 408
- 3.20.16 addBarIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as Integer, name as string) as CDBarLayerMBS 409
- 3.20.17 addBarIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS 409

3.20.19 addBollingerBand(period as Integer, bandWidth as Double, lineColor as color, FillColor as color) as CDInterLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBollingerBand method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.20 addBollingerBand(period as Integer, bandWidth as Double, lineColor as Integer, FillColor as Integer) as CDInterLineLayerMBS 410

3.20.20 addBollingerBand(period as Integer, bandWidth as Double, lineColor as Integer, FillColor as Integer) as CDInterLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Bollinger band to the main chart.

Notes: Arguments:

Argument	Default	Description
period	(Mandatory)	The period to compute the band.
bandWidth	(Mandatory)	The half-width of the band in terms multiples of standard deviation. Typically 2 is used.
lineColor	(Mandatory)	The color of the lines defining the upper and lower limits.
fillColor	(Mandatory)	The color to fill the regional within the band.

Return Value

An CDInterLineLayerMBS object representing the band created.

See also:

- 3.20.19 addBollingerBand(period as Integer, bandWidth as Double, lineColor as color, FillColor as color) as CDInterLineLayerMBS 410

3.20.21 addBollingerWidth(height as Integer, period as Integer, width as Double, colorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBollingerWidth method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.22 addBollingerWidth(height as Integer, period as Integer, width as Double, colorValue as Integer) as CDXYChartMBS 411

3.20.22 addBollingerWidth(height as Integer, period as Integer, width as Double, colorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Bollinger Band Width indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
width	(Mandatory)	The band width to compute the indicator.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.21 addBollingerWidth(height as Integer, period as Integer, width as Double, colorValue as color) as CDXYChartMBS 411

3.20.23 addCandleStick(upColor as color, downColor as color) as CDCandleStickLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addCandleStick method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.24 addCandleStick(upColor as Integer, downColor as Integer) as CDCandleStickLayerMBS 412

3.20.24 addCandleStick(upColor as Integer, downColor as Integer) as CDCandleStickLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a candlestick layer to the main chart.

Notes: Arguments:

Argument	Default	Description
upColor	(Mandatory)	The candle color for a up day.
downColor	(Mandatory)	The candle color for a down day.

Return Value

A CDCandleStickLayerMBS object representing the candlestick layer created.

See also:

- 3.20.23 addCandleStick(upColor as color, downColor as color) as CDCandleStickLayerMBS 412

3.20.25 addCCI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addCCI method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.26 addCCI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 413

3.20.26 addCCI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Community Channel Index indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.
range	(Mandatory)	The distance between the middle line and the upper and lower threshold lines.
upColor	(Mandatory)	The fill color when the indicator exceeds the upper threshold line.
downColor	(Mandatory)	The fill color when the indicator falls below the lower threshold line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.25 addCCI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 412

3.20.27 addChaikinMoneyFlow(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addChaikinMoneyFlow method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.28 addChaikinMoneyFlow(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 413

3.20.28 addChaikinMoneyFlow(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Chaikin Money Flow indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.27 addChaikinMoneyFlow(height as Integer, period as Integer, ColorValue as color) as CDXY-ChartMBS 413

3.20.29 addChaikinOscillator(height as Integer, ColorValue as color) as CDXY-ChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addChaikinOscillator method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.30 addChaikinOscillator(height as Integer, ColorValue as Integer) as CDXYChartMBS 414

3.20.30 addChaikinOscillator(height as Integer, ColorValue as Integer) as CDXY-ChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Chaikin Oscillator indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.29 addChaikinOscillator(height as Integer, ColorValue as color) as CDXYChartMBS 414

3.20.31 addChaikinVolatility(height as Integer, period1 as Integer, period2 as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addChaikinVolatility method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.32 addChaikinVolatility(height as Integer, period1 as Integer, period2 as Integer, ColorValue as Integer) as CDXYChartMBS 415

3.20.32 addChaikinVolatility(height as Integer, period1 as Integer, period2 as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Chaikin Volatility indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period1	(Mandatory)	The period to smooth the range.
period2	(Mandatory)	The period to compute the rate of change of the smoothed range.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.31 addChaikinVolatility(height as Integer, period1 as Integer, period2 as Integer, ColorValue as color) as CDXYChartMBS 415

3.20.33 addCloseLine(ColorValue as color) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addCloseLine method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.34 addCloseLine(ColorValue as Integer) as CDLineLayerMBS 416

3.20.34 addCloseLine(ColorValue as Integer) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a closing price line on the main chart.

Notes: Arguments:

Argument	Default	Description
color	(Mandatory)	The color of the line.

Return Value

A CDLineLayerMBS object representing the line created.

See also:

- 3.20.33 addCloseLine(ColorValue as color) as CDLineLayerMBS 415

3.20.35 addCLV(height as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addCLV method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.36 addCLV(height as Integer, ColorValue as Integer) as CDXYChartMBS 416

3.20.36 addCLV(height as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Close Location Value indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.35 addCLV(height as Integer, ColorValue as color) as CDXYChartMBS 416

3.20.37 addComparison(data() as Double, ColorValue as color, Name as string) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addComparison method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.38 addComparison(data() as Double, ColorValue as Integer, Name as string) as CDLineLayerMBS
417

3.20.38 addComparison(data() as Double, ColorValue as Integer, Name as string) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a data series to the main chart for comparison with the closing price.

Notes: The data series added will be scaled so that the first visible point in the data series will coincide the first visible point of the main closing price. This facilitate comparing the performance of the closing price with the data series. The data series typically is the closing price of another stock, or the value of a market index.

Arguments:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data series.
color	(Mandatory)	The color of the line.
name	(Mandatory)	The name of the data series.

Return Value:

A LineLayer object representing the comparison line created.

See also:

- 3.20.37 addComparison(data() as Double, ColorValue as color, Name as string) as CDLineLayerMBS
417

3.20.39 addDonchianChannel(period as Integer, lineColor as color, FillColor as color) as CDInterLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addDonchianChannel method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.40 addDonchianChannel(period as Integer, lineColor as Integer, FillColor as Integer) as CDInterLineLayerMBS 418

3.20.40 addDonchianChannel(period as Integer, lineColor as Integer, FillColor as Integer) as CDInterLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Donchian channel to the main chart.

Notes: Arguments:

Argument	Default	Description
period	(Mandatory)	The period to compute the band.
lineColor	(Mandatory)	The color of the lines defining the upper and lower limits.
fillColor	(Mandatory)	The color to fill the regional within the band.

Return Value

An CDInterLineLayerMBS object representing the band created.

See also:

- 3.20.39 addDonchianChannel(period as Integer, lineColor as color, FillColor as color) as CDInterLineLayerMBS 417

3.20.41 addDonchianWidth(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addDonchianWidth method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.42 addDonchianWidth(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 418

3.20.42 addDonchianWidth(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Donchian Channel Width indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.41 addDonchianWidth(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS 418

3.20.43 addDPO(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addDPO method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.44 addDPO(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 419

3.20.44 addDPO(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Detrended Price Oscillator indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.43 addDPO(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS 419

3.20.45 addEaseOfMovement(height as Integer, period as Integer, ColorValue1 as color, ColorValue2 as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addEaseOfMovement method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.46 addEaseOfMovement(height as Integer, period as Integer, ColorValue1 as Integer, ColorValue2 as Integer) as CDXYChartMBS 420

3.20.46 addEaseOfMovement(height as Integer, period as Integer, ColorValue1 as Integer, ColorValue2 as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Ease of Movement indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to smooth the indicator.
color1	(Mandatory)	The color of the indicator line.
color2	(Mandatory)	The color of the smoothed indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.45 addEaseOfMovement(height as Integer, period as Integer, ColorValue1 as color, ColorValue2 as color) as CDXYChartMBS 420

3.20.47 addEnvelop(period as Integer, range as Double, lineColor as color, FillColor as color) as CDInterLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addEnvelop method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.48 addEnvelop(period as Integer, range as Double, lineColor as Integer, FillColor as Integer) as CDInterLineLayerMBS 421

3.20.48 addEnvelop(period as Integer, range as Double, lineColor as Integer, FillColor as Integer) as CDInterLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a price envelop to the main chart.

Notes: The price envelop is a defined as a ratio around a moving average. For example, a ratio of 0.2 means 20% above and below the moving average.

Arguments:

Argument	Default	Description
period	(Mandatory)	The period for the moving average.
range	(Mandatory)	The ratio above and below the moving average.
lineColor	(Mandatory)	The color of the lines defining the upper and lower limits.
fillColor	(Mandatory)	The color to fill the regional within the band.

Return Value

An CDInterLineLayerMBS object representing the band created.

See also:

- 3.20.47 addEnvelop(period as Integer, range as Double, lineColor as color, FillColor as color) as CDInterLineLayerMBS 420

3.20.49 addExpMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addExpMovingAvg method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.50 addExpMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS 421

3.20.50 addExpMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an exponential moving average line on the main chart.

Notes: Arguments:

Argument	Default	Description
period	(Mandatory)	The moving average period
color	(Mandatory)	The color of the line.

Return Value

A CDLineLayerMBS object representing the line created.

See also:

- 3.20.49 addExpMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS 421

3.20.51 addFastStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as color, ColorValue2 as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addFastStochastic method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.52 addFastStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as Integer, ColorValue2 as Integer) as CDXYChartMBS 422

3.20.52 addFastStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as Integer, ColorValue2 as Integer) as CDXY-ChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Fast Stochastic indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period1	(Mandatory)	The period to compute the %K line.
period2	(Mandatory)	The period to compute the %D line.
color1	(Mandatory)	The color of the %K line.
color2	(Mandatory)	The color of the %D line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.51 addFastStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as color, ColorValue2 as color) as CDXYChartMBS 422

3.20.53 addHLOC(upColor as color, downColor as color) as CDHLOCLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addHLOC method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.54 addHLOC(upColor as Integer, downColor as Integer) as CDHLOCLayerMBS 423

3.20.54 addHLOC(upColor as Integer, downColor as Integer) as CDHLOCLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a HLOC layer to the main chart.

Notes: Arguments:

Argument	Default	Description
upColor	(Mandatory)	The candle color for a up day.
downColor	(Mandatory)	The candle color for a down day.

Return Value

A CDHLOCLayerMBS object representing the HLOC layer created.

See also:

- 3.20.53 addHLOC(upColor as color, downColor as color) as CDHLOCLayerMBS 423

3.20.55 addIndicator(height as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a blank indicator chart to the finance chart.

Notes: Used internally to add other indicators. Override to change the default formatting (eg. axis fonts, etc.) of the various indicators.

Argument	Default	Description
height	(Mandatory)	The height of the chart in pixels.

Arguments:

Return Value

A CDXYChartMBS object representing the chart created.

3.20.56 addLineIndicator(chart as CDXYChartMBS, data() as Double, Color-Value as color, name as string) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addLineIndicator method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.57 addLineIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as Integer, name as string) as CDLineLayerMBS 424
- 3.20.58 addLineIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS 425
- 3.20.59 addLineIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS 425

3.20.57 addLineIndicator(chart as CDXYChartMBS, data() as Double, Color-Value as Integer, name as string) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a line to an existing indicator chart.

Notes: Arguments:

Argument	Default	Description
c	(Mandatory)	The indicator chart to add the line to.
data	(Mandatory)	The data series of the indicator line.
color	(Mandatory)	The color of the indicator line.
name	(Mandatory)	The name of the indicator.

Return Value

A CDLineLayerMBS object representing the line created.

See also:

3.20. CLASS CDFINANCECHARTMBS 425

- 3.20.56 addLineIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as color, name as string) as CDLineLayerMBS 424
- 3.20.58 addLineIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS 425
- 3.20.59 addLineIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS 425

3.20.58 addLineIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addLineIndicator method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.56 addLineIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as color, name as string) as CDLineLayerMBS 424
- 3.20.57 addLineIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as Integer, name as string) as CDLineLayerMBS 424
- 3.20.59 addLineIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS 425

3.20.59 addLineIndicator(height as Integer, data() as Double, ColorValue as Integer, name as string) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a generic line indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
data	(Mandatory)	The data series of the indicator line.
color	(Mandatory)	The color of the indicator line.
name	(Mandatory)	The name of the indicator.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.56 `addLineIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as color, name as string) as CDLineLayerMBS` 424
- 3.20.57 `addLineIndicator(chart as CDXYChartMBS, data() as Double, ColorValue as Integer, name as string) as CDLineLayerMBS` 424
- 3.20.58 `addLineIndicator(height as Integer, data() as Double, ColorValue as color, name as string) as CDXYChartMBS` 425

3.20.60 `addMACD(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, signalColor as color, divColor as color) as CDXYChartMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addMACD` method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.61 `addMACD(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, signalColor as Integer, divColor as Integer) as CDXYChartMBS` 426

3.20.61 `addMACD(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, signalColor as Integer, divColor as Integer) as CDXYChartMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a MACD indicator chart.

Notes: Arguments:

Argument	Default	Description
<code>height</code>	(Mandatory)	The height of the indicator chart in pixels.
<code>period1</code>	(Mandatory)	The first moving average period to compute the indicator.
<code>period2</code>	(Mandatory)	The second moving average period to compute the indicator.
<code>period3</code>	(Mandatory)	The moving average period of the signal line.
<code>color</code>	(Mandatory)	The color of the indicator line.
<code>signalColor</code>	(Mandatory)	The color of the signal line.
<code>divColor</code>	(Mandatory)	The color of the divergent bars.

Return Value

A `CDXYChartMBS` object representing the chart created.

See also:

- 3.20.60 `addMACD(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, signalColor as color, divColor as color) as CDXYChartMBS` 426

3.20.62 addMainChart(height as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds the main chart - the chart that shows the data for the prices.

Notes: The chart added is initially empty. Other methods, such as `CDFinanceChartMBS.addCandleStick` and `CDFinanceChartMBS.addSimpleMovingAvg`, may be used to add various layers to the chart.

Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the main chart in pixels.

Return Value

A `CDXYChartMBS` object representing the main chart created.

3.20.63 addMassIndex(height as Integer, ColorValue as color, upColor as color, downColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addMassIndex` method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.64 `addMassIndex(height as Integer, ColorValue as Integer, upColor as Integer, downColor as Integer) as CDXYChartMBS` 427

3.20.64 addMassIndex(height as Integer, ColorValue as Integer, upColor as Integer, downColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Mass Index indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
color	(Mandatory)	The color of the indicator line.
upColor	(Mandatory)	The fill color when the indicator exceeds the upper threshold line.
downColor	(Mandatory)	The fill color when the indicator falls below the lower threshold line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.63 addMassIndex(height as Integer, ColorValue as color, upColor as color, downColor as color) as CDXYChartMBS 427

3.20.65 addMedianPrice(ColorValue as color) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addMedianPrice method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.66 addMedianPrice(ColorValue as Integer) as CDLineLayerMBS 428

3.20.66 addMedianPrice(ColorValue as Integer) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a median price line on the main chart.

Notes: Arguments:

Argument	Default	Description
color	(Mandatory)	The color of the line.

Return Value

A CDLineLayerMBS object representing the line created.

See also:

- 3.20.65 addMedianPrice(ColorValue as color) as CDLineLayerMBS 428

3.20.67 addMFI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addMFI method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.68 addMFI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 429

3.20.68 addMFI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Money Flow Index indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.
range	(Mandatory)	The distance between the middle line and the upper and lower threshold lines.
upColor	(Mandatory)	The fill color when the indicator exceeds the upper threshold line.
downColor	(Mandatory)	The fill color when the indicator falls below the lower threshold line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.67 addMFI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 428

3.20.69 addMomentum(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addMomentum method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.70 addMomentum(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 429

3.20.70 addMomentum(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Momentum indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.69 addMomentum(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS
429

3.20.71 addNVI(height as Integer, period as Integer, ColorValue as color, signalColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addNVI method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.72 addNVI(height as Integer, period as Integer, ColorValue as Integer, signalColor as Integer) as CDXYChartMBS
430

3.20.72 addNVI(height as Integer, period as Integer, ColorValue as Integer, signalColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Negative Volume Index indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the signal line.
color	(Mandatory)	The color of the indicator line.
signalColor	(Mandatory)	The color of the signal line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

3.20. CLASS CDFINANCECHARTMBS 431

- 3.20.71 addNVI(height as Integer, period as Integer, ColorValue as color, signalColor as color) as CDXYChartMBS 430

3.20.73 addOBV(height as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addOBV method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.74 addOBV(height as Integer, ColorValue as Integer) as CDXYChartMBS 431

3.20.74 addOBV(height as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an On Balance Volume indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.73 addOBV(height as Integer, ColorValue as color) as CDXYChartMBS 431

3.20.75 addParabolicSAR(accInitial as Double, accIncrement as Double, accMaximum as Double, symbolType as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addParabolicSAR method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.76 addParabolicSAR(accInitial as Double, accIncrement as Double, accMaximum as Double, symbolType as Integer, symbolSize as Integer, fillColor as Integer, edgeColor as Integer) as CDLineLayerMBS 432

3.20.76 addParabolicSAR(accInitial as Double, accIncrement as Double, accMaximum as Double, symbolType as Integer, symbolSize as Integer, fillColor as Integer, edgeColor as Integer) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a parabolic SAR indicator to the main chart.

Notes: Arguments

Argument	Default	Description
accInitial	(Mandatory)	The initial acceleration. A common value to use is 0.02.
accIncrement	(Mandatory)	The incremental acceleration. A common value to use is 0.02.
accMaximum	(Mandatory)	The maximum acceleration. A common value to use is 0.2.
symbolType	(Mandatory)	The symbol used to plot the parabolic SAR. One of the predefined shape constants representing the symbol shape. See Shape Specification for the available built-in shapes.
symbolSize	(Mandatory)	The width and height of the symbol in pixels
fillColor	(Mandatory)	The color used to fill the symbol.
edgeColor	(Mandatory)	The edge color used to draw the edge of the symbol.

Return Value

A CDLineLayerMBS object representing the parabolic SAR created.

See also:

- 3.20.75 addParabolicSAR(accInitial as Double, accIncrement as Double, accMaximum as Double, symbolType as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDLineLayerMBS
431

3.20.77 addPerformance(height as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addPerformance method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.78 addPerformance(height as Integer, ColorValue as Integer) as CDXYChartMBS 432

3.20.78 addPerformance(height as Integer, ColorValue as Integer) as CDXY-ChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Performance indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.77 addPerformance(height as Integer, ColorValue as color) as CDXYChartMBS 432

3.20.79 addPlotAreaTitle(alignment as Integer, text as string) as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a text title above the plot area.

Notes: You may add multiple title above the plot area by calling this method multiple times.

Arguments:

Argument	Default	Description
alignment	(Mandatory)	The alignment with respect to the region that is on top of the plot area. See Alignment Specification for supported alignment types.
text	(Mandatory)	The text to add.

Return Value

The CDTextBoxMBS object representing the text box above the plot area, which may be used to fine-tune the appearance of the text.

3.20.80 addPPO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, signalColor as color, divColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addPPO method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.81 addPPO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, signalColor as Integer, divColor as Integer) as CDXYChartMBS 434

3.20.81 addPPO(*height* as Integer, *period1* as Integer, *period2* as Integer, *period3* as Integer, *ColorValue* as Integer, *signalColor* as Integer, *divColor* as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Percentage Price Oscillator indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period1	(Mandatory)	The first moving average period to compute the indicator.
period2	(Mandatory)	The second moving average period to compute the indicator.
period3	(Mandatory)	The moving average period of the signal line.
color	(Mandatory)	The color of the indicator line.
signalColor	(Mandatory)	The color of the signal line.
divColor	(Mandatory)	The color of the divergent bars.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.80 addPPO(*height* as Integer, *period1* as Integer, *period2* as Integer, *period3* as Integer, *ColorValue* as color, *signalColor* as color, *divColor* as color) as CDXYChartMBS 433

3.20.82 addPVI(*height* as Integer, *period* as Integer, *ColorValue* as color, *signalColor* as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addPVI method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.83 addPVI(*height* as Integer, *period* as Integer, *ColorValue* as Integer, *signalColor* as Integer) as CDXYChartMBS 434

3.20.83 addPVI(*height* as Integer, *period* as Integer, *ColorValue* as Integer, *signalColor* as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Positive Volume Index indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the signal line.
color	(Mandatory)	The color of the indicator line.
signalColor	(Mandatory)	The color of the signal line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.82 addPVI(height as Integer, period as Integer, ColorValue as color, signalColor as color) as CDXYChartMBS 434

3.20.84 addPVO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, signalColor as color, divColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addPVO method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.85 addPVO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, signalColor as Integer, divColor as Integer) as CDXYChartMBS 435

3.20.85 addPVO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, signalColor as Integer, divColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Percentage Volume Oscillator indicator chart.

Notes: Arguments:

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.84 addPVO(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, signalColor as color, divColor as color) as CDXYChartMBS 435

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period1	(Mandatory)	The first moving average period to compute the indicator.
period2	(Mandatory)	The second moving average period to compute the indicator.
period3	(Mandatory)	The moving average period of the signal line.
color	(Mandatory)	The color of the indicator line.
signalColor	(Mandatory)	The color of the signal line.
divColor	(Mandatory)	The color of the divergent bars.

3.20.86 addPVT(height as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addPVT method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.87 addPVT(height as Integer, ColorValue as Integer) as CDXYChartMBS 436

3.20.87 addPVT(height as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Price Volume Trend indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.86 addPVT(height as Integer, ColorValue as color) as CDXYChartMBS 436

3.20.88 addROC(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addROC method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.89 addROC(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 437

3.20.89 addROC(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Rate of Change indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.88 addROC(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS 436

3.20.90 addRSI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addRSI method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.91 addRSI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 437

3.20.91 addRSI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXY-ChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Relative Strength Index indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.
range	(Mandatory)	The distance between the middle line and the upper and lower threshold lines.
upColor	(Mandatory)	The fill color when the indicator exceeds the upper threshold line.
downColor	(Mandatory)	The fill color when the indicator falls below the lower threshold line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.90 addRSI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 437

3.20.92 addSimpleMovingAvg(period as Integer, ColorValue as color) as CD-LineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addSimpleMovingAvg method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.93 addSimpleMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS 438

3.20.93 addSimpleMovingAvg(period as Integer, ColorValue as Integer) as CD-LineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a simple moving average line on the main chart.

Notes: Arguments:

Argument	Default	Description
period	(Mandatory)	The moving average period
color	(Mandatory)	The color of the line.

Return Value

A CDLineLayerMBS object representing the line created.

See also:

- 3.20.92 addSimpleMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS 438

3.20.94 addSlowStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as color, ColorValue2 as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addSlowStochastic method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.95 addSlowStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as Integer, ColorValue2 as Integer) as CDXYChartMBS 439

3.20.95 addSlowStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as Integer, ColorValue2 as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Slow Stochastic indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period1	(Mandatory)	The period to compute the %K line.
period2	(Mandatory)	The period to compute the %D line.
color1	(Mandatory)	The color of the %K line.
color2	(Mandatory)	The color of the %D line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.94 addSlowStochastic(height as Integer, period1 as Integer, period2 as Integer, ColorValue1 as color, ColorValue2 as color) as CDXYChartMBS 439

3.20.96 addStdDev(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addStdDev method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.97 addStdDev(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS 440

3.20.97 addStdDev(height as Integer, period as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Moving Standard Deviation indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.96 addStdDev(height as Integer, period as Integer, ColorValue as color) as CDXYChartMBS 439

3.20.98 addStochRSI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addStochRSI method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.99 addStochRSI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 440

3.20.99 addStochRSI(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Stochastic RSI indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.
range	(Mandatory)	The distance between the middle line and the upper and lower threshold lines.
upColor	(Mandatory)	The fill color when the indicator exceeds the upper threshold line.
downColor	(Mandatory)	The fill color when the indicator falls below the lower threshold line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.98 addStochRSI(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 440

3.20.100 addThreshold(chart as CDXYChartMBS, layer as CDLineLayerMBS, topRange as Double, topColor as color, bottomRange as Double, bottomColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addThreshold method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.101 addThreshold(chart as CDXYChartMBS, layer as CDLineLayerMBS, topRange as Double, topColor as Integer, bottomRange as Double, bottomColor as Integer) 441

3.20.101 addThreshold(chart as CDXYChartMBS, layer as CDLineLayerMBS, topRange as Double, topColor as Integer, bottomRange as Double, bottomColor as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an upper/lower threshold range to an existing indicator chart.

Notes: Arguments:

See also:

- 3.20.100 addThreshold(chart as CDXYChartMBS, layer as CDLineLayerMBS, topRange as Double, topColor as color, bottomRange as Double, bottomColor as color) 441

Argument	Default	Description
c	(Mandatory)	The indicator chart to add the threshold range to.
layer	(Mandatory)	The line layer that the threshold range applies to.
topRange	(Mandatory)	The upper threshold.
topColor	(Mandatory)	The color to fill the region of the line that is above the upper threshold.
bottomRange	(Mandatory)	The lower threshold.
bottomColor	(Mandatory)	The color to fill the region of the line that is below the lower threshold.

3.20.102 addTriMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addTriMovingAvg method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.103 addTriMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS 443

3.20.103 addTriMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a triangular moving average line on the main chart.

Notes: Arguments:

Argument	Default	Description
period	(Mandatory)	The moving average period
color	(Mandatory)	The color of the line.

Return Value

A CDLineLayerMBS object representing the line created.

See also:

- 3.20.102 addTriMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS 442

3.20.104 addTRIX(height as Integer, Period as Integer, ColorValue as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addTRIX method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.105 addTRIX(height as Integer, Period as Integer, ColorValue as Integer) as CDXYChartMBS 443

3.20.105 addTRIX(height as Integer, Period as Integer, ColorValue as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a TRIX indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.104 addTRIX(height as Integer, Period as Integer, ColorValue as color) as CDXYChartMBS 443

3.20.106 addTypicalPrice(ColorValue as color) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addTypicalPrice method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.107 addTypicalPrice(ColorValue as Integer) as CDLineLayerMBS 444

3.20.107 addTypicalPrice(ColorValue as Integer) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a typical price line on the main chart.

Notes: Arguments:

Argument	Default	Description
color	(Mandatory)	The color of the line.

Return Value

A CDLineLayerMBS object representing the line created.

See also:

- 3.20.106 addTypicalPrice(ColorValue as color) as CDLineLayerMBS 444

3.20.108 addUltimateOscillator(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addUltimateOscillator method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.109 addUltimateOscillator(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 445

3.20.109 addUltimateOscillator(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an Ultimate Oscillator indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period1	(Mandatory)	The first moving average period to compute the indicator.
period2	(Mandatory)	The second moving average period to compute the indicator.
period3	(Mandatory)	The third moving average period to compute the indicator.
color	(Mandatory)	The color of the indicator line.
range	(Mandatory)	The distance between the middle line and the upper and lower threshold lines.
upColor	(Mandatory)	The fill color when the indicator exceeds the upper threshold line.
downColor	(Mandatory)	The fill color when the indicator falls below the lower threshold line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.108 addUltimateOscillator(height as Integer, period1 as Integer, period2 as Integer, period3 as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 444

3.20.110 addVolBars(height as Integer, upColor as color, downColor as color, flatColor as color) as CDBarLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addVolBars method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.111 addVolBars(height as Integer, upColor as Integer, downColor as Integer, flatColor as Integer) as CDBarLayerMBS 446

3.20.111 addVolBars(height as Integer, upColor as Integer, downColor as Integer, flatColor as Integer) as CDBarLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a volume bar chart layer on the main chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the bar chart layer in pixels.
upColor	(Mandatory)	The color to be used on an 'up' day. An 'up' day is a day where the closing price is higher than that of the previous day.
downColor	(Mandatory)	The color to be used on a 'down' day. A 'down' day is a day where the closing price is lower than that of the previous day.
flatColor	(Mandatory)	The color to be used on a 'flat' day. A 'flat' day is a day where the closing price is the same as that of the previous day.

Return Value

A CDBarLayerMBS object representing the bar layer created.

See also:

- 3.20.110 addVolBars(height as Integer, upColor as color, downColor as color, flatColor as color) as CDBarLayerMBS 445

3.20.112 addVolIndicator(height as Integer, upColor as color, downColor as color, flatColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addVolIndicator method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.113 addVolIndicator(height as Integer, upColor as Integer, downColor as Integer, flatColor as Integer) as CDXYChartMBS 446

3.20.113 addVolIndicator(height as Integer, upColor as Integer, downColor as Integer, flatColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a Volume indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
upColor	(Mandatory)	The color to used on an 'up' day. An 'up' day is a day where the closing price is higher than that of the previous day.
downColor	(Mandatory)	The color to used on a 'down' day. A 'down' day is a day where the closing price is lower than that of the previous day.
flatColor	(Mandatory)	The color to used on a 'flat' day. A 'flat' day is a day where the closing price is the same as that of the previous day.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.112 addVolIndicator(height as Integer, upColor as color, downColor as color, flatColor as color) as CDXYChartMBS 446

3.20.114 addWeightedClose(ColorValue as color) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addWeightedClose method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.115 addWeightedClose(ColorValue as Integer) as CDLineLayerMBS 447

3.20.115 addWeightedClose(ColorValue as Integer) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a weighted close line on the main chart.

Notes: Arguments:

Argument	Default	Description
color	(Mandatory)	The color of the line.

Return Value

A CDLineLayerMBS object representing the line created.

See also:

- 3.20.114 addWeightedClose(ColorValue as color) as CDLineLayerMBS 447

3.20.116 addWeightedMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addWeightedMovingAvg method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.117 addWeightedMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS 448

3.20.117 addWeightedMovingAvg(period as Integer, ColorValue as Integer) as CDLineLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a weighted moving average line on the main chart.

Notes: Arguments:

Argument	Default	Description
period	(Mandatory)	The moving average period
color	(Mandatory)	The color of the line.

Return Value

A CDLineLayerMBS object representing the line created.

See also:

- 3.20.116 addWeightedMovingAvg(period as Integer, ColorValue as color) as CDLineLayerMBS 448

3.20.118 addWilliamR(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addWilliamR method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.119 addWilliamR(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS 449

3.20.119 addWilliamR(height as Integer, period as Integer, ColorValue as Integer, range as Double, upColor as Integer, downColor as Integer) as CDXYChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a William %R indicator chart.

Notes: Arguments:

Argument	Default	Description
height	(Mandatory)	The height of the indicator chart in pixels.
period	(Mandatory)	The period to compute the indicator.
color	(Mandatory)	The color of the indicator line.
range	(Mandatory)	The distance between the middle line and the upper and lower threshold lines.
upColor	(Mandatory)	The fill color when the indicator exceeds the upper threshold line.
downColor	(Mandatory)	The fill color when the indicator falls below the lower threshold line.

Return Value

A CDXYChartMBS object representing the chart created.

See also:

- 3.20.118 addWilliamR(height as Integer, period as Integer, ColorValue as color, range as Double, upColor as color, downColor as color) as CDXYChartMBS 448

3.20.120 Constructor(width as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new FinanceChart object.

Notes: Arguments:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels. The height will be automatically determined as the chart is built.

3.20.121 currentChart as CDXYChartMBS

Plugin Version: 12.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the current chart of this finance chart.

Notes: While you add parts to the finance chart, you can access the last one here.

3.20.122 enableAntiAlias(antiAlias as boolean)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables/Disables anti-alias.

Notes: Enabling anti-alias makes the line smoother. Disabling anti-alias make the chart file size smaller, and so can be downloaded faster through the Internet. The default is to enable anti-alias.

Arguments:

Argument	Default	Description
antiAlias	(Mandatory)	True to enable anti-alias. False to disable anti-alias.

3.20.123 getToolTipDateFormat as string

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the date/time format string to use for tool tips.

Notes: The format string returned will be one of the format strings specified when calling `CDFinanceChartMBS.set-ToolTipDateFormat` based on data point spacing, or the equivalent default value. The format string may then be used in other `ChartDirector` functions such as `CDBaseChartMBS.getHTMLImageMap` for producing image maps.

Returns the date/time format string to use for tool tips.

3.20.124 mainChart as CDXYChartMBS

Plugin Version: 12.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the main chart of this finance chart.

3.20.125 setAxisOnRight(b as Boolean)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets whether the main y-axis is on right of left side of the plot area.

Notes: The default is on right.

Arguments:

Argument	Default	Description
b	(Mandatory)	A true value means the y-axis is on right. A false value means the y-axis is on left.

3.20.126 setData(timeStamps() as Double, highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, volData() as Double, extraPoints as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data to be used in the chart.

Notes: If some of the data are not available, some artificial values should be used. For example, if the high and low values are not available, you may use closeData as highData and lowData.

Arguments:

Argument	Default	Description
timeStamps	(Mandatory)	An array of dates/times for the time intervals.
highData	(Mandatory)	The high values in the time intervals.
lowData	(Mandatory)	The low values in the time intervals.
openData	(Mandatory)	The open values in the time intervals.
closeData	(Mandatory)	The close values in the time intervals.
volData	(Mandatory)	The volume values in the time intervals.
extraPoints	(Mandatory)	The number of leading time intervals that are not displayed in the chart. These intervals are typically used for computing indicators that require extra leading data, such as moving averages.

3.20.127 setDateLabelFormat(yearFormat as string, firstMonthFormat as string, otherMonthFormat as string, firstDayFormat as string, otherDayFormat as string, firstHourFormat as string, otherHourFormat as string)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the date/time formats to use for the x-axis labels under various cases.

Notes: See Parameter Substitution and Formatting on the syntax of ChartDirector format strings.

Arguments:

Argument	Default	Description
yearFormat	(Mandatory)	The format for displaying labels on an axis with yearly ticks. The default is " { value yyyy } ".
firstMonthFormat	(Mandatory)	The format for displaying labels on an axis with monthly ticks. This parameter applies to the first available month of a year (usually January) only, so it can be formatted differently from the other labels. The default is "{ value mmm yy } ".
otherMonthFormat	(Mandatory)	The format for displaying labels on an axis with monthly ticks. This parameter applies to months other than the first available month of a year. The default is " { value mmm } ".
firstDayFormat	(Mandatory)	The format for displaying labels on an axis with daily ticks. This parameter applies to the first available day of a month only, so it can be formatted differently from the other labels. The default is "{ value d mmm } ".
otherDayFormat	(Mandatory)	The format for displaying labels on an axis with daily ticks. This parameter applies to days other than the first available day of a month. The default is " { value d } ".
firstHourFormat	(Mandatory)	The format for displaying labels on an axis with hourly resolution. This parameter applies to the first tick of a day only, so it can be formatted differently from the other labels. The default is "{ value d mmm h:nna } ".
otherHourFormat	(Mandatory)	The format for displaying labels on an axis with hourly resolution. This parameter applies to ticks at hourly boundaries, except the first tick of a day. The default is " { value h:nna } ".

3.20.128 setDateLabelSpacing(labelSpacing as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the minimum label spacing between two labels on the time axis.

Notes: Arguments:

Argument	Default	Description
labelSpacing	(Mandatory)	The label spacing in pixels. The default is 50 pixels.

3.20.129 setLegendStyle(font as string, fontSize as Double, fontColor as color, bgColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setLegendStyle method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.130 setLegendStyle(font as string, fontSize as Double, fontColor as Integer, bgColor as Integer)

3.20.130 setLegendStyle(font as string, fontSize as Double, fontColor as Integer, bgColor as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets legend font style and background color.

Notes: The default is Arial 8 pt black font on a semi-transparent light grey (80CCCCCC) background.

Arguments:

Argument	Default	Description
font	(Mandatory)	The font of the legend text.
fontSize	(Mandatory)	The font size of the legend text in points.
fontColor	(Mandatory)	The color of the legend text.
bgColor	(Mandatory)	The background color of the legend box.

See also:

- 3.20.129 setLegendStyle(font as string, fontSize as Double, fontColor as color, bgColor as color) 452

3.20.131 setLogScale(b as Boolean)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Determines if log scale should be used for the main chart.

Notes: The default is to use linear scale.

Arguments:

Argument	Default	Description
b	(Mandatory)	A true value means using log scale. A false value means using linear scale.

3.20.132 setMargins(leftMargin as Integer, topMargin as Integer, rightMargin as Integer, bottomMargin as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the margins around the plot area.

Notes: The default is 40 pixels for the left and right margins, and 30 pixels for the top margin and 35 pixels for the bottom margin.

Arguments:

Argument	Default	Description
leftMargin	(Mandatory)	The distance from the left side of the plot area to the left side of the chart.
topMargin	(Mandatory)	The distance from the top of the plot area to the top of the chart.
rightMargin	(Mandatory)	The distance from the right side of the plot area to the right side of the chart.
bottomMargin	(Mandatory)	The distance from the bottom of the plot area to the bottom of the chart.

3.20.133 setNumberLabelFormat(formatString as string)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the number format to use.

Notes: The default number format in FinanceChart is "P3". See Parameter Substitution and Formatting on the syntax of ChartDirector format strings.

Arguments:

Argument	Default	Description
formatString	(Mandatory)	The format for displaying numbers.

3.20.134 setPercentageAxis as CDAxisMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Label the axis of the main chart in percentage unit.

Notes: By default, the axis of the main chart will be labelled in price unit. If a percentage axis is used, the axis will be labelled in percentage unit, with the first visible point of the closing price being 100%.

3.20.135 setPlotAreaBorder(borderColor as color, borderGap as Integer)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setPlotAreaBorder method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.136 setPlotAreaBorder(borderColor as Integer, borderGap as Integer)

3.20.136 setPlotAreaBorder(borderColor as Integer, borderGap as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the plot area border color and the gap distance between charts.

Notes: The default is a grey (888888) border with two 2 pixels gap between charts.

Arguments:

Argument	Default	Description
borderColor	(Mandatory)	The plot area border color.
borderGap	(Mandatory)	The distance between charts.

See also:

- 3.20.135 setPlotAreaBorder(borderColor as color, borderGap as Integer) 454

3.20.137 setPlotAreaStyle(bgColor as color, majorHGridColor as color, majorVGridColor as color, minorHGridColor as color, minorVGridColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setPlotAreaStyle method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.138 setPlotAreaStyle(bgColor as Integer, majorHGridColor as Integer, majorVGridColor as Integer, minorHGridColor as Integer, minorVGridColor as Integer) 455

3.20.138 setPlotAreaStyle(bgColor as Integer, majorHGridColor as Integer, majorVGridColor as Integer, minorHGridColor as Integer, minorVGridColor as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the plot area background and grid colors.

Notes: The default is a white background with light grey (DDDDDD) grid lines.

Arguments:

See also:

Argument	Default	Description
bgColor	(Mandatory)	The plot area background color.
majorHGridColor	(Mandatory)	Major horizontal grid color.
majorVGridColor	(Mandatory)	Major vertical grid color.
minorHGridColor	(Mandatory)	Minor horizontal grid color. In current version, minor horizontal grid is not used.
minorVGridColor	(Mandatory)	Minor vertical grid color.

- 3.20.137 `setPlotAreaStyle(bgColor as color, majorHGridColor as color, majorVGridColor as color, minorHGridColor as color, minorVGridColor as color)` 455

3.20.139 `setToolTipDateFormat(monthFormat as string, dayFormat as string, hourFormat as string)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the date/time formats to use for the tool tips under various cases.

Notes: See Parameter Substitution and Formatting on the syntax of ChartDirector format strings.

Arguments:

Argument	Default	Description
monthFormat	(Mandatory)	The tool tip format to use if the data point spacing is one or more months (more than 30 days). The default is " [{ xLabel mmm yyyy }] "
dayFormat	(Mandatory)	The tool tip format to use if the data point spacing is 1 day to less than 30 days. The default is " [{ xLabel mmm d, yyyy }] "
hourFormat	(Mandatory)	The tool tip format to use if the data point spacing is less than 1 day. The default is " [{ xLabel mmm d, yyyy hh:nn:ss }] "

3.20.140 `setXAxisStyle(font as string, fontSize as Double, fontColor as color, fontAngle as Double)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setXAxisStyle` method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.141 `setXAxisStyle(font as string, fontSize as Double, fontColor as Integer, fontAngle as Double)` 457

3.20.141 setXAxisStyle(font as string, fontSize as Double, fontColor as Integer, fontAngle as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets x-axis label style.

Notes: The default is Arial 8 pt black color as font with no text rotation.

Arguments:

Argument	Default	Description
font	(Mandatory)	The font of the axis labels.
fontSize	(Mandatory)	The font size of the axis labels in points.
fontColor	(Mandatory)	The color of the axis labels.
fontAngle	(Mandatory)	The rotation of the axis labels.

See also:

- 3.20.140 setXAxisStyle(font as string, fontSize as Double, fontColor as color, fontAngle as Double) 456

3.20.142 setYAxisStyle(font as string, fontSize as Double, fontColor as color, bgColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setYAxisStyle method, but uses color instead of integer data type for passing color values.

See also:

- 3.20.143 setYAxisStyle(font as string, fontSize as Double, fontColor as Integer, bgColor as Integer) 457

3.20.143 setYAxisStyle(font as string, fontSize as Double, fontColor as Integer, bgColor as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets y-axis label style.

Notes: The default is Arial 8 pt black color, with 14 pixels margin.

Arguments:

See also:

Argument	Default	Description
font	(Mandatory)	The font of the axis labels.
fontSize	(Mandatory)	The font size of the axis labels in points.
fontColor	(Mandatory)	The color of the axis labels.
axisMargin	(Mandatory)	The margin at the top of the y-axis in pixels (to leave space for the legend box).

- 3.20.142 `setYAxisStyle(font as string, fontSize as Double, fontColor as color, bgColor as color)` 457

3.21 class CDFinanceSimulatorMBS

3.21.1 class CDFinanceSimulatorMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: FinanceSimulator is a utility class to produce tables with random numbers that look like valid financial data series.

Notes: It facilitates testing and demonstrating ChartDirector financial charts without needing a real database table.

3.21.2 Methods

3.21.3 Constructor(seed as Integer, startTime as Double, endTime as Double, resolution as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a FinanceSimulator object.

Notes:

Argument	Default	Description
seed	(Mandatory)	The seed to be used in the random number generator.
startTime	(Mandatory)	The start date/time of the financial date series.
endTime	(Mandatory)	The end date/time of the financial date series.
resolution	(Mandatory)	The resolution of the financial data series in seconds. For example, a value of 86400 means daily data (1 day = 86400 seconds). A value of 7 * 86400 means weekly data. A value of 30 * 86400 means monthly data.

See also:

- 3.21.4 Constructor(seed as string, startTime as Double, endTime as Double, resolution as Integer) 459

3.21.4 Constructor(seed as string, startTime as Double, endTime as Double, resolution as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a FinanceSimulator object.

Notes:

See also:

- 3.21.3 Constructor(seed as Integer, startTime as Double, endTime as Double, resolution as Integer) 459

Argument	Default	Description
seed	(Mandatory)	The seed to be used in the random number generator.
startTime	(Mandatory)	The start date/time of the financial date series.
endTime	(Mandatory)	The end date/time of the financial date series.
resolution	(Mandatory)	The resolution of the financial data series in seconds. For example, a value of 86400 means daily data (1 day = 86400 seconds). A value of 7 * 86400 means weekly data. A value of 30 * 86400 means monthly data.

3.21.5 getCloseData as CDArrayMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the close data series.

3.21.6 getHighData as CDArrayMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the high data series.

3.21.7 getLowData as CDArrayMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the low data series.

3.21.8 getOpenData as CDArrayMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the open data series.

3.21.9 getTimeStamps as CDArrayMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the timestamps of the data series.

3.21.10 `getVolData` as `CArrayMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the volume data series.

3.22 class CDHLOCLayerMBS

3.22.1 class CDHLOCLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The HLOCLayer class represents high-low-open-close layers.

Notes: Subclass of the CDBaseBoxLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.22.2 Methods

3.22.3 setColorMethod(colorMethod as Integer, riseColor as color, fallColor as color, leadValue as Double = -1.7E308)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setColorMethod method, but uses color instead of integer data type for passing color values.

See also:

- 3.22.4 setColorMethod(colorMethod as Integer, riseColor as Integer, fallColor as Integer = -1, leadValue as Double = -1.7E308) 462

3.22.4 setColorMethod(colorMethod as Integer, riseColor as Integer, fallColor as Integer = -1, leadValue as Double = -1.7E308)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the coloring method for the HLOC symbols.

Notes: See XYChart.addHLOCLayer for the supported coloring methods for HLOC symbols.

Argument	Default	Description
colorMethod	(Mandatory)	The method to color the HLOC layer. Please refer to XYChart.addHLOCLayer3 for the supported coloring methods.
riseColor	(Mandatory)	The color to be used on an "up" day.
fallColor	-1	The color to be used on a "down" day. The default value of -1 means it is the same as the riseColor.
leadValue	[-Infinity]	The lead value to act as the closing pricing before the first day, so as to determine if the first day is an "up" or "down" day.

See also:

- 3.22.3 setColorMethod(colorMethod as Integer, riseColor as color, fallColor as color, leadValue as Double = -1.7E308) 462

3.23 class CDImageMapHandlerMBS

3.23.1 class CDImageMapHandlerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: ImageMapHandler is a utility class to handle image maps in HTML format (that is, as <AREA>tags).

Notes: It determines if a given point is on a hot spot as defined by the image map, and retrieves the hot spot parameters.

The BaseChart.getHTMLImageMap method can be used to automatically generate image maps for charts automatically.

The advantages of using HTML image map format is that it is easy for developers to customize the hot spots. For example, one can create custom buttons in the chart image by drawing custom text boxes, and then create custom image maps to define the text boxes as hot spots. These custom image maps can be appended to the image maps generated by BaseChart.getHTMLImageMap using simple string concatenation.

ImageMapHandler will process the various attributes of the HTML <AREA> tags as follows:

AttributeDescription

coords	This attribute defines the position and shape of the hot spot.
href	The URL specified in this attribute will become the attributes of the hot spot. The path portion of the URL will become the path attribute, while the query parameters will become the attributes of the hot spot as is.
title	This attribute defines the tool tip text to display when the mouse moves over and stops on the hot spot.

3.23.2 Methods

3.23.3 Constructor(ImageMap as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Constructs an image map handler object for the given image map.

Notes:

Argument	Default	Description
imageMap	(Mandatory)	A text string representing the image map in HTML format.

3.23.4 getHotSpot(xCoordinate as double, yCoordinate as double) as integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the hot spot under the given point as the current hot spot.

Notes: This method will retrieve the hot spot under the given point as the "current hot spot". Its attributes can then be retrieved using ImageMapHandler.getKey, ImageMapHandler.getValue and ImageMapHandler.getValue2.

This method returns an integer representing the hot spot, or -1 if the point is not over any hot spot. Each hot spot region is represented by a unique integer, which can be any arbitrary number. The number is mainly used to determine if the cursor has changed from one hot spot to another hot spot.

Argument	Default	Description
x	(Mandatory)	The x coordinate of a point of which the hot spot is to be retrieved.
y	(Mandatory)	The y coordinate of a point of which the hot spot is to be retrieved.

Return Value

An integer representing the hot spot, or -1 if the given point is not over any hot spot.

3.23.5 getKey(i as Integer) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the key of an attribute of the current hot spot.

Notes: Each hot spot is associated with an arbitrary number of attributes, in which each attribute consists of a key and a value. This method returns the key of the (i + 1)th attribute. (The first attribute is 0, while the nth attribute is n - 1.)

If the argument i is equal or larger than the number of attributes, a null string will be returned. This behaviour can be used to enumerate the attributes and to determine how many attributes are associated for the hot spot.

Argument	Default	Description
i	(Mandatory)	The index of the attribute to retrieve (index of first attribute is 0).

Return Value

A string representing the key of the attribute, or null if there is no such attribute.

3.23.6 `getValue(i as Integer) as string`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the value of an attribute of the current hot spot by using its numeric index.

Notes: Each hot spot is associated with an arbitrary number of attributes, in which each attribute consists of a key and a value. This method returns the value of the $(i + 1)$ th attribute. (The first attribute is 0, while the n th attribute is $n - 1$.)

Argument	Default	Description
<code>i</code>	(Mandatory)	The index of the attribute to retrieve (index of first attribute is 0).

Return Value

A string representing the value of the attribute, or null if there is no such attribute.

See also:

- 3.23.7 `getValue(key as string) as string` 466

3.23.7 `getValue(key as string) as string`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the value of an attribute of the current hot spot by using its key.

Notes: Each hot spot is associated with an arbitrary number of attributes, in which each attribute consists of a key and a value. This method returns the value of the attribute given the key.

Argument	Default	Description
<code>key</code>	(Mandatory)	A text string representing the key of the attribute.

Return Value

A string representing the value of the attribute, or null if there is no such attribute.

See also:

- 3.23.6 `getValue(i as Integer) as string` 466

3.24 class CDInterLineLayerMBS

3.24.1 class CDInterLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The InterLineLayer class represents interline layers.

Notes: Interline layers are used to color the region between two lines. The lines can come from a variety of layer types, such as line layers, spline layers, step line layers, trend layers or axis mark lines.

Subclass of the CDLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.24.2 Methods

3.24.3 setGapColor(gapColor12 as Color, gapColor21 as Color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setGapColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.24.4 setGapColor(gapColor12 as Integer, gapColor21 as Integer = -1) 467

3.24.4 setGapColor(gapColor12 as Integer, gapColor21 as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color used to fill the area under NoValue data points.

Notes: By default, if there are NoValue data points on the lines, ChartDirector will interpolate across the points. The region will remain continuous.

This method can be used to set up an alternative colors to represent the regions at NoValue data point positions. In particular, if the colors are set to Transparent, the NoValue data points will result in gaps in the fill region.

See also:

- 3.24.3 setGapColor(gapColor12 as Color, gapColor21 as Color) 467

Argument	Default	Description
gapColor12	(Mandatory)	The color used to fill the region under NoValue positions when value of the first line is greater of the value of the second line.
gapColor21	-1	The color used to fill the region under NoValue positions when value of the second line is greater of the value of the first line. The default value of -1 means it is the same as gapColor12.

3.25 class CDLayerMBS

3.25.1 class CDLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Layer class is the base class for all XYChart layer classes.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [Release notes for SQL or ChartDirector?](#)
- [MBS Real Studio Plugins, version 11.3pr5](#)

Xojo Developer Magazine

- [18.5, page 76: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.25.2 Methods

3.25.3 addCustomAggregateLabel(dataItem as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a custom aggregate label to an aggregated object.

Notes: Aggregate data labels applies to layer types that contains "aggregated data", such as stacked bar layer and stacked area layer. In these layer types, data labels (see `Layer.setDataLabelStyle`) represents a single data item, while aggregate labels represents the stacked object.

See [Font Specification](#) for details on various font attributes.

Return Value

A `TextBox` object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- [3.25.4 addCustomAggregateLabel\(dataItem as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0\) as CDTextBoxMBS](#) 470

Argument	Default	Description
dataItem	(Mandatory)	The index of the aggregated object. The first aggregated object is 0, while the nth aggregated object is (n - 1).
label	(Mandatory)	A text string representing the custom aggregate label. Parameter Substitution and Formatting is supported.
font	""	The font used to draw the labels.
fontSize	8	The font size used to draw the labels.
fontColor	TextColor	The color used to draw the labels.
fontAngle	0	The rotation angle of the labels.

3.25.4 addCustomAggregateLabel(dataItem as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addCustomAggregateLabel method, but uses color instead of integer data type for passing color values.

See also:

- 3.25.3 addCustomAggregateLabel(dataItem as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS 469

3.25.5 addCustomDataLabel(dataSet as Integer, dataItem as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a custom data label to a data point.

Notes: Data labels are text used label data points in the layer. Please refer to Layer.setDataLabelStyle for a more detail description.

See Font Specification for details on various font attributes.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.25.6 addCustomDataLabel(dataSet as Integer, dataItem as Integer, label as string, font as string,

Argument	Default	Description
dataSet	(Mandatory)	The data set number for the data point. The first data set is 0, while the nth data set is (n - 1).
dataItem	(Mandatory)	The data point number for the data point within the data set. The first data point is 0, while the nth data point is (n - 1).
label	(Mandatory)	A text string representing the data label. Parameter Substitution and Formatting is supported.
font	""	The font used to draw the label.
fontSize	8	The font size used to draw the label.
fontColor	TextColor	The color used to draw the label.
fontAngle	0	The rotation angle of the label.

fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS

471

3.25.6 addCustomDataLabel(dataSet as Integer, dataItem as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addCustomDataLabel method, but uses color instead of integer data type for passing color values.

See also:

- 3.25.5 addCustomDataLabel(dataSet as Integer, dataItem as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS
470

3.25.7 addCustomGroupLabel(dataGroup as Integer, dataItem as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a custom data group label to object representing the group.

Notes: Data group label applies to layer types that may represent data groups. See Layer.addDataGroup on how data groups are created and the layer types that support data groups.

See Font Specification for details on various font attributes.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

Argument	Default	Description
dataGroup	(Mandatory)	The data group number for the data point. The first data group is 0, while the nth data group is (n - 1).
dataItem	(Mandatory)	The data point number for the data point within the data group. The first data point is 0, while the nth data point is (n - 1).
label	(Mandatory)	A text string representing the data label. Parameter Substitution and Formatting is supported.
font	""	The font used to draw the label.
fontSize	8	The font size used to draw the label.
fontColor	TextColor	The color used to draw the label.
fontAngle	0	The rotation angle of the label.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.25.8 addCustomGroupLabel(dataGroup as Integer, dataItem as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS 472

3.25.8 addCustomGroupLabel(dataGroup as Integer, dataItem as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addCustomGroupLabel method, but uses color instead of integer data type for passing color values.

See also:

- 3.25.7 addCustomGroupLabel(dataGroup as Integer, dataItem as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS 471

3.25.9 addDataGroup(name as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Open a new data group.

Notes: Currently, only stacked bar layers support data groups for creating "multi-stacked" bars.

In a normal stacked bar layer, all data sets are stacked on top of one another, creating one stacked bar per x-axis position. If data grouping is used, data sets within the same data group will be stacked up. So there

may be multiple stacked bars in each x-axis position. These stacked bars are drawn side by side.

When you add a data set using `Layer.addDataSet`, the data set will belong to the current data group. The `addDataGroup` method can be used to open a new data group, so that subsequent data sets will belong to that new group.

You may associate a name with a data group. The name can then be used in data labels or image maps to identify the data group.

Argument	Default	Description
<code>name</code>	<code>""</code>	Name of the data group.

3.25.10 `addDataSet(data as CDArrayMBS, colorvalue as color, name as string = "") as CDDataSetMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addDataSet` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.25.11 `addDataSet(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "") as CDDataSetMBS` 473
- 3.25.12 `addDataSet(data() as Double, colorvalue as color, name as string = "") as CDDataSetMBS` 474
- 3.25.13 `addDataSet(data() as Double, colorvalue as Integer = -1, name as string = "") as CDDataSetMBS` 474

3.25.11 `addDataSet(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "") as CDDataSetMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a data set to the chart layer.

Notes:

Return Value

A `DataSet` object representing the data set added. You may use the methods of this object to fine-tune how the data set is drawn on the chart.

See also:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data set.
color	-1	The color to draw the data item. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.

- 3.25.10 `addDataSet(data as CDArrayMBS, colorvalue as color, name as string = "")` as `CDDataSetMBS` 473
- 3.25.12 `addDataSet(data() as Double, colorvalue as color, name as string = "")` as `CDDataSetMBS` 474
- 3.25.13 `addDataSet(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDDataSetMBS` 474

3.25.12 `addDataSet(data() as Double, colorvalue as color, name as string = "")` as `CDDataSetMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addDataSet` method, but uses `color` instead of `integer` data type for passing color values.

See also:

- 3.25.10 `addDataSet(data as CDArrayMBS, colorvalue as color, name as string = "")` as `CDDataSetMBS` 473
- 3.25.11 `addDataSet(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "")` as `CDDataSetMBS` 473
- 3.25.13 `addDataSet(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDDataSetMBS` 474

3.25.13 `addDataSet(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDDataSetMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a data set to the chart layer.

Notes:

Return Value

A `DataSet` object representing the data set added. You may use the methods of this object to fine-tune how the data set is drawn on the chart.

See also:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data set.
color	-1	The color to draw the data item. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.

- 3.25.10 addDataSet(data as CDArrayMBS, colorvalue as color, name as string = "") as CDDatasetMBS
473
- 3.25.11 addDataSet(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "") as CD-
DatasetMBS 473
- 3.25.12 addDataSet(data() as Double, colorvalue as color, name as string = "") as CDDatasetMBS
474

3.25.14 addExtraField(numbers() as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an array of numbers/dates to be used as an extra field in various places.

Notes: This method merely stores the data inside the layer object. The Parameter Substitution and Formatting mechanism will determine how the data are to be used.

A common use for extra fields is to specify extra information (such as a custom serial number for the data points) to be displayed on data labels or on tool tips, or to supply extra query parameters in clickable charts. All these are achieved by specifying the extra field on the data label template or image map templates during parameter substitution.

Argument	Default	Description
numbers	(Mandatory)	An array of numbers/dates to be stored inside the layer object.

See also:

- 3.25.15 addExtraField(texts() as string) 475

3.25.15 addExtraField(texts() as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an array of text to be used as an extra field in various places.

Notes: This method merely stores the data inside the layer object. The Parameter Substitution and Formatting mechanism will determine how the data are to be used.

A common use for extra fields is to specify extra information (such as a custom serial number for the data points) to be displayed on data labels or on tool tips, or to supply extra query parameters in clickable charts. All these are achieved by specifying the extra field on the data label template or image map templates during parameter substitution.

Argument	Default	Description
texts	(Mandatory)	An array of text to be stored inside the layer object.

See also:

- 3.25.14 `addExtraField(numbers() as Double)` 475

3.25.16 `alignLayer(layer as CDLayerMBS, dataSet as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Aligns the layer with a data set of another bar chart layer in Side layout.

Notes: The Side layout is a layout method specific to the bar chart layer. In this layout method, multiple bars belonging to the same x-position are laid out side by side. Thus the bars are not centered exactly on the x-position, but are shifted. The shift amount depends on the data set numbers of the bars.

In some cases, it may be necessary to overlay another layer (eg. a line chart layer, or a box-whisker chart layer) on top of the bars. The `alignLayer` method ensures the data points are shifted by the same amount as a given data set on a given `BarLayer`.

Argument	Default	Description
layer	(Mandatory)	The <code>BarLayer</code> which contains the bars to synchronize with.
dataSet	(Mandatory)	The data set of the bars to synchronize with.

3.25.17 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.25.18 `getDataSet(dataSet as Integer) as CDDatasetMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the requested `DataSet` object.

Notes:

Argument	Default	Description
dataSet	(Mandatory)	A data set number, starting from 0. The first data set object is 0, and the nth data set is (n-1).

Return Value

The requested DataSet object.

3.25.19 getDataSetByZ(z as Integer) as CDDatasetMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets a DataSet object based on the order in which it is being drawn.

Notes: In certain layer types, the data sets are drawn in a certain order. For example, in a line layer, the data sets represent lines and they are drawn one by one. In contrast, for a candlestick layer, the high, low, open and close data sets are combined into candlestick symbols, and there is no specific order on which data sets are drawn.

This method can be used to get the data set based on the order it is being drawn. If the data sets are not drawn in any specific order, this method will return the data sets in the reverse order to which it is added to the layer.

Argument	Default	Description
zIndex	(Mandatory)	The z-index of the required data set. The z-index of the first data set drawn is 0. The z-index for the Nth data set drawn is N - 1.

Returns the DataSet object at the specified z-order.

3.25.20 getDataSetCount as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the number of data sets in the layer.

Notes: Returns the number of data sets in the layer.

3.25.21 `getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates an HTML image map for all data points as represented on the layer.

Notes: Please refer to `BaseChart.getHTMLImageMap` for the detail description of this method.

This method should be called only after creating the chart image (eg. using `BaseChart.makeChart`, `BaseChart.makeChart2` or `BaseChart.makeChart3`). The image map cannot be determined without creating the chart image first.

Argument	Default	Description
<code>url</code>	(Mandatory)	The URL to be used in the "href" attribute of the image map. Parameter Substitution and Formatting is supported. Use an empty string if no href attribute is needed.
<code>queryFormat</code>	""	A text string representing the template of the query parameters to be appended to the URL. Parameter Substitution and Formatting is supported.

The special keyword " { default } " represents the default query parameters. This is useful for specifying appending to the default.

Note that an empty string means to use the default query query parameters. To specify no query parameter, use a space character.

<code>extraAttr</code>	""	A text string to specify additional attributes to add to the <code><area></code> tag. Parameter Substitution and Formatting is supported.
<code>offsetX</code>	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
<code>offsetY</code>	0	An offset to be added to all y coordinates in the image map. See <code>offsetX</code> above for description.

Return Value

A text string containing the image map generated.

3.25.22 `getImageCoor(dataSet as Integer, dataItem as Integer = &h80000001, offsetX as Integer = 0, offsetY as Integer = 0) as string`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the image map coordinates of a data point as represented in the layer as HTML image map attributes.

Notes: The image map coordinates will be in the following format:

```
shape=" [ shape ] " cords=" [ x1 ] , [ y1 ] , [ x2 ] , [ y2 ] ..."
```

This format is specially designed so that it can easily be incorporated into HTML image maps.

This method should be called only after creating the chart image (eg. using `BaseChart.makeChart`, `BaseChart.makeChart2` or `BaseChart.makeChart3`). The image map cannot be determined without creating the chart image first.

Argument	Default	Description
<code>dataSet</code>	(Mandatory)	The data set number for the data point. The first data set is 0, while the nth data set is (n-1).
<code>dataItem</code>	(Mandatory)	The index of the data point within the data set. The first data point is 0, while the nth data point is (n-1).
<code>offsetX</code>	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
<code>offsetY</code>	0	An offset to be added to all y coordinates in the image map. See <code>offsetX</code> above for description.

Return Value

A text string representing the coordinates of the data point as represented in the layer in HTML image map attribute format.

3.25.23 `getImageCoor2(dataItem as Integer, offsetX as Integer = 0, offsetY as Integer = 0) as string`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the image map coordinates of a region containing all data points at the given x-position as HTML image map attributes.

Notes: For example, in a stacked bar chart, this method will obtain the image map coordinates of the whole stacked bar, which contains multiple data points at the same x-position.

The image map coordinates will be in the following format:

```
shape="" cords=" [ x1 ] , [ y1 ] , [ x2 ] , [ y2 ] ..."
```

This format is specially designed so that it can easily be included into HTML image maps.

This method should be called only after creating the chart image (eg. using `BaseChart.makeChart`, `BaseChart.makeChart2` or `BaseChart.makeChart3`). The image map cannot be determined without creating the chart image first.

Argument	Default	Description
dataItem	(Mandatory)	The x-position of the data points.
offsetX	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
offsetY	0	An offset to be added to all y coordinates in the image map. See offsetX above for description.

Return Value

A text string representing the image map coordinates of a region containing all data points at the given x-position as HTML image map attributes.

3.25.24 getLegendIcon(dataSetNo as Integer) as string

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the textual representation of the legend icon for a data set.

Notes: The textual representation can be used in any ChartDirector API that supports the ChartDirector Mark Up Language. A common usage is to insert the icon to a cell in a CDMLTable to label the data series in the table, or to create a custom legend table.

The legend icon is the same legend icon that would be used in the CDLegendBoxMBS. If you modify the icon appearance using the methods of the CDLegendBoxMBS object (such as using CDLegendBoxMBS.setKeyBorder to configure the legend key border), the modification will also apply to the icon returned this method.

Arguments:

Argument	Default	Description
dataSetNo	(Mandatory)	The data set for which the legend icon represents.

Return Value

The textual representation of the legend icon for the data set.

3.25.25 getNearestXValue(target as Double) as Double

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x data value that is nearest to the specified x pixel coordinate.

Notes: This method will search all x data values in the Layer to look for the x data value that is nearest to

the given x coordinate. If there are two x data values equally near to the specified x pixel coordinate, this method will arbitrarily return one of the values.

Argument	Default	Description
xCoord	(Mandatory)	The x pixel coordinate to search for.

Returns the x data value that is nearest to the specified x coordinate.

3.25.26 getXCoord(value as Double) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x pixel coordinate of a point given the x data value.

Notes: Note: You must call BaseChart.layout first before calling this method. It is because ChartDirector needs to perform auto-scaling and determine the axis scale first before it can compute the coordinates.

Argument	Default	Description
v	(Mandatory)	The x data value.

Return Value

The x coordinate of the x data value.

3.25.27 getXIndexOf(xValue as Double, tolerance as Double = 0) as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the data point index at the specified x data value.

Notes: The index can be used in DataSet.getValue and DataSet.getPosition to obtain the y values and positions of the data points for all data sets in the layer.

The tolerance argument is to allow this method to consider data points that are not exactly at xValue, but are within tolerance from it.

If multiple data points are within tolerance, this method will return the index of the data point nearest to xValue. If multiple data points are equally near to xValue, this method will choose one arbitrarily and return its data point index. If no data points are found, this method will return -1.

Returns the data point index of the data point nearest to the specified x data value up to the specified tolerance, or -1 if no data points are found.

Argument	Default	Description
xValue	(Mandatory)	The x data value used to look for the data points in order to get the data point index of the nearest data point.
tolerance	0	The tolerance allowed for for data points that are not exactly at the specified xValue.

3.25.28 getXPosition(i as Integer) as Double

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x positional value of a data point, as measured on the primary x-axis.

Notes:

Argument	Default	Description
i	(Mandatory)	The data point index of the data point. The first data point is 0; the nth data point is (n - 1).

3.25.29 getYCoor(value as Double, axis as boolean=true) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y pixel coordinate of a point given the y data value.

Notes: Note: You must call BaseChart.layout first before calling this method. It is because ChartDirector needs to perform auto-scaling and determine the axis scale first before it can compute the coordinates.

Argument	Default	Description
v	(Mandatory)	The y data value.
yAxis	nil	The y-axis to use to determine the pixel coordinates of data values. The y-axis may be obtained using XYChart.yAxis, XYChart.yAxis2 or XYChart.addAxis. The default is to use the primary y-axis.

For backward compatibility, the axis argument can also be a boolean value. A true value means the primary y-axis. A false value means the secondary y-axis.

Return Value

The y coordinate of the y data value.

See also:

- 3.25.30 getYCoor(value as Double, axis as CDAxisMBS) as Integer

482

3.25.30 getYCoor(value as Double, axis as CDAxisMBS) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y pixel coordinate of a point given the y data value.

Notes: Note: You must call BaseChart.layout first before calling this method. It is because ChartDirector needs to perform auto-scaling and determine the axis scale first before it can compute the coordinates.

Argument	Default	Description
v	(Mandatory)	The y data value.
yAxis	nil	The y-axis to use to determine the pixel coordinates of data values. The y-axis may be obtained using XYChart.yAxis, XYChart.yAxis2 or XYChart.addAxis. The default is to use the primary y-axis.

For backward compatibility, the axis argument can also be a boolean value. A true value means the primary y-axis. A false value means the secondary y-axis.

Return Value

The y coordinate of the y data value.

See also:

- 3.25.29 getYCoord(value as Double, axis as boolean=true) as Integer

482

3.25.31 moveBack(layer as CDLayerMBS=nil)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Moves the layer in front of another layer.

Notes: By default, the front to back ordering of the Layer objects are the same as the order in which they are added to the chart. For example, a layer added first will be in front of a layer added last.

This method can be used to move the layer behind another layer.

Arguments:

Argument	Default	Description
layer	nil	The Layer for this layer to move behind. A nil object means the layer will be moved behind all current layers.

3.25.32 moveFront(layer as CDLayerMBS=nil)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Moves the layer in front of another layer.

Notes: By default, the front to back ordering of the CDLayerMBS objects are the same as the order in

which they are added to the chart. For example, a layer added first will be in front of a layer added last.

This method can be used to move the layer in front of another layer.

Arguments:

Argument	Default	Description
layer	nil	The Layer for this layer to move in front of. The default value of null means the layer will be moved in front of all current layers.

3.25.33 set3D(d as Integer = -1, zGap as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the 3D depth of the layer, and the 3D gap between the current layer and the next layer.

Notes:

Argument	Default	Description
d	-1	The 3D depth of the layer in pixels. -1 means the 3D depth is automatically calculated.
zGap	0	The 3D gap between the current layer and the next layer in pixels.

3.25.34 setAggregateLabelFormat(formatString as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the aggregate label format.

Notes: By default, the aggregate label format is " { value } ". Please refer to Parameter Substitution and Formatting on available parameters and how to format them.

Argument	Default	Description
formatString	(Mandatory)	The format string.

3.25.35 setAggregateLabelStyle(font as string = "", fontSize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0) as CD-TextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables aggregate labels and sets their styles.

Notes: Aggregate data labels applies to layer types that contains "aggregated data", such as stacked bar layer and stacked area layer. In these layer types, data labels (see `Layer.setDataLabelStyle`) represents a single data item, while aggregate labels represents the stacked object.

See Font Specification for details on various font attributes.

Argument	Default	Description
font	""	The font used to draw the labels.
fontSize	8	The font size used to draw the labels.
fontColor	TextColor	The color used to draw the labels.
fontAngle	0	The rotation angle of the labels.

Return Value

A `TextBox` object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.25.36 `setAggregateLabelStyle(font as string, fontSize as Double, fontcolor as color, fontAngle as Double = 0)` as `CDTextBoxMBS` 485

3.25.36 `setAggregateLabelStyle(font as string, fontSize as Double, fontcolor as color, fontAngle as Double = 0)` as `CDTextBoxMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setAggregateLabelStyle` method, but uses color instead of integer data type for passing color values.

See also:

- 3.25.35 `setAggregateLabelStyle(font as string = "", fontSize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0)` as `CDTextBoxMBS` 484

3.25.37 `setBaseLine(BaseLine as Double)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the baseline for the data representation.

Notes: Certain data representation, such as bars in bar layers and area in area layers, are drawn from a

base line. The base line by default is $y = 0$.

For example, for a vertical bar layer, the bars start from $y = 0$ and grow upwards for positive data (assuming the y-axis is not reversed), and downwards for negative data.

This method can be used to modify the base line to other values.

Arguments:

Argument	Default	Description
baseLine	(Mandatory)	The value of the base line.

3.25.38 setBorderColor(colorvalue as color, lightingEffect as Integer = 0)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the default border color and 3D raised effect when drawing data sets on the layer.

Notes: This method only applies to layers that represents data with elements that have borders (e.g. bar layer).

This method affect all data sets. To set the color of one particular data set, use `DataSet.setDataColor`.

Argument	Default	Description
color	(Mandatory)	The border color.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument is also used to support <code>glassEffect</code> and <code>softLighting</code> effects.

See also:

- 3.25.39 `setBorderColor(colorvalue as Integer, lightingEffect as Integer = 0)` 486

3.25.39 setBorderColor(colorvalue as Integer, lightingEffect as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the default border color and 3D raised effect when drawing data sets on the layer.

Notes: This method only applies to layers that represents data with elements that have borders (e.g. bar layer).

This method affect all data sets. To set the color of one particular data set, use `DataSet.setDataColor`.

Argument	Default	Description
color	(Mandatory)	The border color.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument is also used to support <code>glassEffect</code> and <code>softLighting</code> effects.

See also:

- 3.25.38 `setBorderColor(colorvalue as color, lightingEffect as Integer = 0)` 486

3.25.40 `setDataCombineMethod(m as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the method used to combine multiple data sets in a layer.

Notes: This method is used only for layers that can combine multiple data sets in different ways. Some layers can only support a subset of the data combine methods defined below.

ConstantValueDescription

Side	3	The data sets are combined by plotting the data representation side by side.
Stack	1	The data sets are combined by stacking up their data representations.
Overlay	0	The data sets are combined by drawing them independently, overlapping each others.
Percentage	4	The data sets are combined similar to the Stack method, except that the data is scaled so that each stack always summed up to 100. In other words, the region that a data item occupies in a stack represents the percentage of the data item relative to sum of all the data items in the stack.

Argument	Default	Description
m	(Mandatory)	one of the predefined constants representing the data combine method to use.

3.25.41 `setDataLabelFormat(formatString as string)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data label format.

Notes: This method affects all data labels in the layer. To set the label format for one particular data set only, use `DataSet.setDataLabelFormat`.

Data labels are text used label data points in the layer. Please refer to `Layer.setDataLabelStyle` for a more detail description.

By default, the data label format is " { value } ". Please refer to Parameter Substitution and Formatting on available parameters and how to format them.

Argument	Default	Description
<code>formatString</code>	(Mandatory)	The format string.

3.25.42 `setDataLabelStyle(font as string = "", fontSize as Double = 8, fontcolor as color, fontAngle as Double = 0)` as `CDTextBoxMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setDataLabelStyle` method, but uses color instead of integer data type for passing color values.

See also:

- 3.25.43 `setDataLabelStyle(font as string = "", fontSize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0)` as `CDTextBoxMBS` 488

3.25.43 `setDataLabelStyle(font as string = "", fontSize as Double = 8, fontcolor as Integer = &hfff0002, fontAngle as Double = 0)` as `CDTextBoxMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables data labels and sets their styles.

Notes: This method affects all data labels in the layer. To set the style for one particular data set only, use `DataSet.setDataLabelStyle`.

Data labels are text used label data points in the layer. Different layer types put data labels in different positions. For example, in a bar chart, data labels are put at the internal end of the bar. For a line chart, data labels are put above the data points.

For some chart types, the position of the data labels can be manipulated by calling the `TextBox.setAlignment` method of the `TextBox` object returned by the `setDataLabelStyle` method.

See Font Specification for details on various font attributes.

Argument	Default	Description
font	""	The font used to draw the labels.
fontSize	8	The font size used to draw the labels.
fontColor	TextColor	The color used to draw the labels.
fontAngle	0	The rotation angle of the labels.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.25.42 setDataLabelStyle(font as string = "", fontSize as Double = 8, fontcolor as color, fontAngle as Double = 0) as CDTextBoxMBS 488

3.25.44 setHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "")

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Override the default arguments used when generating HTML image map for the layer.

Notes: BaseChart.getHTMLImageMap can be used to generate HTML image map for the whole chart. When BaseChart.getHTMLImageMap is used, the image map for all layers will be generated based on the arguments supplied to BaseChart.getHTMLImageMap.

The setHTMLImageMap method can be used to override those arguments for a chart layer, so the image map for that layer can be different.

For a detail description of image maps, please refer to BaseChart.getHTMLImageMap.

Argument	Default	Description
url	(Mandatory)	The URL to be used in the "href" attribute of the image map. Parameter Substitution and Formatting is supported.

The special keyword " { default } " represents the global URL as specified in BaseChart.getHTMLImageMap. This field is useful for specifying appending to the global URL.

Note that an empty string also means to use the global URL. To specify no URL, use the special keyword " { none } ".

To disable the entire image map, use the special keyword " { disable } ".

`queryFormat` "" A text string representing the template of the query parameters to be appended to the URL. Parameter Substitution and Formatting is supported.

The special keyword " { default } " represents the global query parameters as specified in `BaseChart.getHTMLImageMap`. This field is useful for specifying appending to the global query parameters.

Note that an empty string also means to use the global query parameters. To specify no query parameters, use the special keyword " { none } ".

`extraAttr` "" A text string to specify additional attributes to add to the `<area>` tag. Parameter Substitution and Formatting is supported.

The special keyword " { default } " represents the global additional attributes as specified in `BaseChart.getHTMLImageMap`. This field is useful for specifying appending to the global additional attributes.

Note that an empty string also means to use the global additional attributes. To specify no additional attributes, use the special keyword " { none } ".

3.25.45 `setLegend(m as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the order of the data set names as appeared in the legend box.

Notes: This method is for backward compatibility. It is equivalent to `Layer.setLegendOrder(m)`.

Argument	Default	Description
<code>m</code>	(Mandatory)	One of the predefined constants representing the legend entry ordering method for data sets within the layer.

3.25.46 `setLegendOrder(dataSetOrder as Integer, layerOrder as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the order of the data set names as appeared in the legend box.

Notes: By default, ChartDirector will add named data sets to the the legend box. The ordering of the entries follows the order in which the layers are created. For data sets within the same layer, the ordering follows the order in which the data sets are added.

To support flexible ordering of the legend entries, ChartDirector employs a legend entry priority system. Please refer to LegendBox for details of the legend entry priority system.

The base priority of the layer can be specified using the layerOrder argument. The priority of the data sets within a layer can be modified by using the dataSetOrder argument, which must be one of the following predefined constant.

ConstantValueDescription

NormalLegend	0	The data set priority is $10 \times (\text{data_set_index} + 1)$. Legend entries are ordered in the order with which the data sets are created.
ReverseLegend	1	The data sets priority is the reverse of the normal data set priority. Legend entries are ordered in the reverse order with which the data sets are created.
NoLegend	2	The data set names are not added to the legend box at all.

Argument	Default	Description
dataSetOrder	(Mandatory)	One of the predefined constants representing the legend entry ordering method for data sets within the layer.
layerOrder	-1	The base legend entry priority of the layer. The default value of -1 means the priority is $10000 \times (\text{layer_index} + 1)$.

3.25.47 setLineWidth(w as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the default line width of lines when drawing data sets on the layer.

Notes: This method only applies to layers that employ lines to represent data (e.g. line layer).

This method affect all data sets. To set the line width of one particular data set, use DataSet.setLineWidth.

Argument	Default	Description
w	(Mandatory)	The width of the line in pixels.

3.25.48 setUseYAxis(axis as CDAxisMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Determine the y-axis to use when drawing the data sets.

Notes: This method affects all data sets in the layer. To set the y-axis to use for a particular data set, use `DataSet.setUseYAxis`.

Argument	Default	Description
a	(Mandatory)	The y-axis to use when drawing the data sets.

3.25.49 setUseYAxis2(b as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Determine if the primary or secondary y-axis should be used when drawing the data sets.

Notes: This method affects all data sets in the layer. To set the y-axis to use for a particular data set, use `DataSet.setUseYAxis2`.

Note: `Layer.setUseYAxis` is a more general method that can support more than 2 y-axes.

Argument	Default	Description
b	true	A true value means the secondary y-axis will be used. A false value means the primary y-axis will be used.

3.25.50 setXData(data as CDArrayMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the x values of the data points in the data sets.

Notes: In most `ChartDirector` XY chart layers, when a data set is added, only the y Values for the data points are needed. For example, in a bar layer, you just need to specify the values of the bars (y values). `ChartDirector` will automatically layout the bars evenly on the x-axis. You can then specify the labels on the x-axis using `Axis.setLabels`.

In `ChartDirector`, this type of x-axis scaling is called "enumerated" scale. Please refer to `Axis.setLabels` for a more detail explanation of "enumerated" scale.

However, in some cases, it may be necessary to specify the x values explicitly. For example, if a chart contains data points that are not evenly distributed on the x-axis, it is necessary to specify the x values explicitly.

It is because enumerated scale always assume the data points are distributed evenly on the x- axis. In this case, the `setXData` method can be used to specify the x values for the data points.

In general, if the data points are evenly distributed on the x-axis, it is recommended enumerated x-scale be used and no x values are necessary. Even the data points are evenly distributed, except that some data points are missing, it is still possible to use enumerated x-scale by using the `NoValue` constant to represent missing data points.

On the other hand, if the data points are by its nature not evenly distributed, the x values should be provided explicitly using the `setXData` method.

Each layer only supports one x values series. All data sets in the layer will use the same x value series. If two data sets have different x values, they should be put in two separate layers.

Argument	Default	Description
<code>xData</code>	(Mandatory)	An array of numbers representing the x value series.

See also:

- 3.25.51 `setXData(data() as Double)` 493
- 3.25.52 `setXData(dates() as date)` 494
- 3.25.53 `setXData(dates() as dateTime)` 495
- 3.25.54 `setXData(minValue as Double, maxValue as Double)` 495

3.25.51 `setXData(data() as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the x values of the data points in the data sets.

Notes: In most `ChartDirector` XY chart layers, when a data set is added, only the y Values for the data points are needed. For example, in a bar layer, you just need to specify the values of the bars (y values). `ChartDirector` will automatically layout the bars evenly on the x-axis. You can then specify the labels on the x-axis using `Axis.setLabels`.

In `ChartDirector`, this type of x-axis scaling is called "enumerated" scale. Please refer to `Axis.setLabels` for a more detail explanation of "enumerated" scale.

However, in some cases, it may be necessary to specify the x values explicitly. For example, if a chart contains data points that are not evenly distributed on the x-axis, it is necessary to specify the x values explicitly. It is because enumerated scale always assume the data points are distributed evenly on the x- axis. In this

case, the `setXData` method can be used to specify the x values for the data points.

In general, if the data points are evenly distributed on the x-axis, it is recommended enumerated x-scale be used and no x values are necessary. Even the data points are evenly distributed, except that some data points are missing, it is still possible to use enumerated x-scale by using the `NoValue` constant to represent missing data points.

On the other hand, if the data points are by its nature not evenly distributed, the x values should be provided explicitly using the `setXData` method.

Each layer only supports one x values series. All data sets in the layer will use the same x value series. If two data sets have different x values, they should be put in two separate layers.

Argument	Default	Description
<code>xData</code>	(Mandatory)	An array of numbers representing the x value series.

See also:

- 3.25.50 `setXData(data as CArrayMBS)` 492
- 3.25.52 `setXData(dates() as date)` 494
- 3.25.53 `setXData(dates() as dateTime)` 495
- 3.25.54 `setXData(minValue as Double, maxValue as Double)` 495

3.25.52 `setXData(dates() as date)`

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Sets the x values of the data points in the data sets.

Notes: In most ChartDirector XY chart layers, when a data set is added, only the y Values for the data points are needed. For example, in a bar layer, you just need to specify the values of the bars (y values). ChartDirector will automatically layout the bars evenly on the x-axis. You can then specify the labels on the x-axis using `Axis.setLabels`.

In ChartDirector, this type of x-axis scaling is called "enumerated" scale. Please refer to `Axis.setLabels` for a more detail explanation of "enumerated" scale.

However, in some cases, it may be necessary to specify the x values explicitly. For example, if a chart contains data points that are not evenly distributed on the x-axis, it is necessary to specify the x values explicitly. It is because enumerated scale always assume the data points are distributed evenly on the x- axis. In this case, the `setXData` method can be used to specify the x values for the data points.

In general, if the data points are evenly distributed on the x-axis, it is recommended enumerated x-scale be used and no x values are necessary. Even the data points are evenly distributed, except that some data points are missing, it is still possible to use enumerated x-scale by using the NoValue constant to represent missing data points.

On the other hand, if the data points are by its nature not evenly distributed, the x values should be provided explicitly using the setXData method.

Each layer only supports one x values series. All data sets in the layer will use the same x value series. If two data sets have different x values, they should be put in two separate layers.

Argument	Default	Description
xData	(Mandatory)	An array of numbers representing the x value series.

See also:

- 3.25.50 setXData(data as CDArrayMBS) 492
- 3.25.51 setXData(data() as Double) 493
- 3.25.53 setXData(dates() as dateTime) 495
- 3.25.54 setXData(minValue as Double, maxValue as Double) 495

3.25.53 setXData(dates() as dateTime)

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the x values of the data points in the data sets.

See also:

- 3.25.50 setXData(data as CDArrayMBS) 492
- 3.25.51 setXData(data() as Double) 493
- 3.25.52 setXData(dates() as date) 494
- 3.25.54 setXData(minValue as Double, maxValue as Double) 495

3.25.54 setXData(minValue as Double, maxValue as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the x values of the data points in the data sets as evenly distributed in a range.

Notes: This method is most useful when two layers contain data at different x axis scale. An example is a

line layer with one data point per minute, and another line layer with one data point per 5 minutes.

In the above example, in one hour, the first layer will have 60 data points, while the second layer will have 12 data points. If enumerated x-scale is used, the x-axis will contain 60 positions evenly distributed. The 60 data points in the first layer will corresponds to the 60 positions in the x-axis, which is correct. However, the 12 data points in the second layer will corresponds to the first 12 positions on the x-axis, which is not the desired result.

Instead, the 12 data points in the second layer should corresponds to positions 0, 5, 10, 15, A little thought will review that the 12 data points in the second layer should be distributed evenly among positions 0 - 55. The `setXData2` method can be used to inform `ChartDirector` about the scaling used in the second layer.

Argument	Default	Description
<code>minValue</code>	(Mandatory)	The x value of the first point in a data set.
<code>maxValue</code>	(Mandatory)	The x value of the last point in the data set.

See also:

- 3.25.50 `setXData(data as CArrayMBS)` 492
- 3.25.51 `setXData(data() as Double)` 493
- 3.25.52 `setXData(dates() as date)` 494
- 3.25.53 `setXData(dates() as dateTime)` 495

3.25.55 `xZoneColor(threshold as Double, belowColor as color, aboveColor as color) as Integer`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `xZoneColor` method, but uses color instead of integer data type for passing color values.

See also:

- 3.25.56 `xZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer) as Integer` 496

3.25.56 `xZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a x-zone color. A x-zone will change from one color to another depending on a threshold value on the x-axis.

Notes: For example, if a x-zone color is used as the line color in a line layer, the line will switch from one color to another when it passes through a certain value on the x-axis. Similarly, if a x-zone color is used as the fill color in an area layer, the area will switch from one color to another when it passes through a certain value on the x-axis.

The two colors used in a x-zone color can be other dynamic colors. For example, one color could be a solid color, while the other color could be a dash line color (see !BaseChart.dashLineColor). When this x-zone color is as the line color, the line will change from a solid style to a dash line style when the line passes through a certain value on the x-axis

You may create x-zone colors with more than 2 zones by cascading multiple x-zone colors.

Argument	Default	Description
threshold	(Mandatory)	The x value serving as the threshold for switching between two colors.
belowColor	(Mandatory)	The color to use when the x-axis value of the pixel is smaller than the threshold.
aboveColor	(Mandatory)	The color to use when the x-axis value of the pixel is greater than the threshold.

Return Value

A 32-bit integer representing the x-zone color.

See also:

- 3.25.55 xZoneColor(threshold as Double, belowColor as color, aboveColor as color) as Integer 496

3.25.57 yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as boolean=true) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other yZoneColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.25.58 yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as CDAX-isMBS) as Integer 498
- 3.25.59 yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as boolean=true) as Integer 498
- 3.25.60 yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as CDAX-isMBS) as Integer 499

3.25.58 `yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as CDAxisMBS)` as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `yZoneColor` method, but uses color instead of integer data type for passing color values.

See also:

- 3.25.57 `yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as boolean=true) as Integer` 497
- 3.25.59 `yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as boolean=true) as Integer` 498
- 3.25.60 `yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as CDAxisMBS) as Integer` 499

3.25.59 `yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as boolean=true)` as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a y-zone color. A y-zone will change from one color to another depending on a threshold value on the y-axis.

Notes: For example, if a y-zone color is used as the fill color in an area layer, the area will switch from one color to another when its value is higher than a certain value on the y-axis.

The two colors used in a y-zone color can be other dynamic colors. You may create y-zone colors with more than 2 zones by cascading multiple y-zone colors.

Argument	Default	Description
<code>threshold</code>	(Mandatory)	The y value serving as the threshold for switching between two colors.
<code>belowColor</code>	(Mandatory)	The color to use when the y-axis value of the pixel is smaller than the threshold.
<code>aboveColor</code>	(Mandatory)	The color to use when the y-axis value of the pixel is greater than the threshold.
<code>yAxis</code>	nil	The y-axis to use to determine the pixel coordinates of data values. The y-axis may be obtained using <code>XYChart.yAxis</code> , <code>XYChart.yAxis2</code> or <code>XYChart.addAxis</code> . The default is to use the primary y-axis.

For backward compatibility, the axis argument can also be a boolean value. A true value means the primary y-axis. A false value means the secondary y-axis.

Return Value

A 32-bit integer representing the y-zone color.

See also:

3.25. CLASS CDLAYERMBS

499

- 3.25.57 `yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as boolean=true) as Integer` 497
- 3.25.58 `yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as CDAxisMBS) as Integer` 498
- 3.25.60 `yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as CDAxisMBS) as Integer` 499

3.25.60 `yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as CDAxisMBS) as Integer`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a y-zone color. A y-zone will change from one color to another depending on a threshold value on the y-axis.

Notes: For example, if a y-zone color is used as the fill color in an area layer, the area will switch from one color to another when its value is higher than a certain value on the y-axis.

The two colors used in a y-zone color can be other dynamic colors. You may create y-zone colors with more than 2 zones by cascading multiple y-zone colors.

Argument	Default	Description
<code>threshold</code>	(Mandatory)	The y value serving as the threshold for switching between two colors.
<code>belowColor</code>	(Mandatory)	The color to use when the y-axis value of the pixel is smaller than the threshold.
<code>aboveColor</code>	(Mandatory)	The color to use when the y-axis value of the pixel is greater than the threshold.
<code>yAxis</code>	nil	The y-axis to use to determine the pixel coordinates of data values. The y-axis may be obtained using <code>XYChart.yAxis</code> , <code>XYChart.yAxis2</code> or <code>XYChart.addAxis</code> . The default is to use the primary y-axis.

For backward compatibility, the axis argument can also be a boolean value. A true value means the primary y-axis. A false value means the secondary y-axis.

Return Value

A 32-bit integer representing the y-zone color.

See also:

- 3.25.57 `yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as boolean=true) as Integer` 497
- 3.25.58 `yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as CDAxisMBS) as Integer` 498
- 3.25.59 `yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as boolean=true) as Integer` 498

3.26 class CDLegendBoxMBS

3.26.1 class CDLegendBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class LegendBox represents legend boxes.

Notes: Subclass of the CDTextBoxMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [Chart Diagrams with Xojo](#)

Xojo Developer Magazine

- [7.5, page 32: Easy Charts and Graphs Part 2, Using the ChartDirector Plugin](#)
- [7.4, page 34: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [20.4, page 40: PDF Pie Charts, Adding Xojo Charts to Your PDFs by Stefanie Juchmes](#)
- [20.1, page 46: Cool Charts and Heatmaps, Using Monkeybread Software's ChartDirector Plugin by Stefanie Juchmes](#)
- [18.5, page 71: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.26.2 Methods

3.26.3 addKey(pos as Integer, text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addKey method, but uses color instead of integer data type for passing color values.

See also:

- [3.26.4 addKey\(pos as Integer, text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil\)](#) 501
- [3.26.5 addKey\(text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil\)](#) 501
- [3.26.6 addKey\(text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CD-DrawAreaMBS=nil\)](#) 502

3.26.4 addKey(pos as Integer, text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a custom entry to the legend box with a given legend entry priority.

Notes: See LegendBox for more information on legend entry priority system.

Argument	Default	Description
pos	(Mandatory)	The legend entry priority
text	(Mandatory)	The text of the legend entry.
color	(Mandatory)	The icon color of the legend entry.
lineWidth	0	The line width for legend entry that represents lines in line charts.
drawarea	nil	A DrawArea containing the data symbol represents the legend entry. This is primarily used for legend entries in line charts with data symbols, or in scatter charts.

See also:

- 3.26.3 addKey(pos as Integer, text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil) 500
- 3.26.5 addKey(text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil) 501
- 3.26.6 addKey(text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CD-DrawAreaMBS=nil) 502

3.26.5 addKey(text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addKey method, but uses color instead of integer data type for passing color values.

See also:

- 3.26.3 addKey(pos as Integer, text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil) 500
- 3.26.4 addKey(pos as Integer, text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil) 501
- 3.26.6 addKey(text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CD-DrawAreaMBS=nil) 502

3.26.6 addKey(text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a custom entry to the legend box.

Notes: This method adds an entry within a legend entry priority of 1 (followed by 2, 3, 4, ... for repeated calls). This has higher priority than entries added automatically by ChartDirector for representing data sets. As a result, by default, the custom entry will appear before the automatic entries.

To control the order of the custom entry relative to the automatic entries, use `LegendBox.addKey2`.

See `LegendBox` for more information on legend entry priority system.

Argument	Default	Description
text	(Mandatory)	The text of the legend entry.
color	(Mandatory)	The icon color of the legend entry.
lineWidth	0	The line width for legend entry that represents lines in line charts.
drawarea	nil	A <code>DrawArea</code> containing the data symbol represents the legend entry. This is primarily used for legend entries in line charts with data symbols, or in scatter charts.

See also:

- 3.26.3 `addKey(pos as Integer, text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)` 500
- 3.26.4 `addKey(pos as Integer, text as string, colorvalue as Integer, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)` 501
- 3.26.5 `addKey(text as string, colorvalue as color, lineWidth as Integer = -1, drawarea as CDDrawAreaMBS=nil)` 501

3.26.7 addText(pos as Integer, text as string)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a custom text to the legend box.

See also:

- 3.26.8 `addText(text as string)` 502

3.26.8 addText(text as string)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a custom text to the legend box.

See also:

- 3.26.7 addText(pos as Integer, text as string)

502

3.26.9 getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates an HTML image map for the legend.

Notes: By default, ChartDirector will add named data representation to the legend box. For pie charts, named sectors (sectors that have text labels) will be added. For xy charts, named data sets will be added. For polar charts, named layers will be added.

You may add custom entries to the legend box by using LegendBox.addKey.

To control the ordering of the legend entries, ChartDirector employs a legend entry priority system. Each entry is given a priority number, and the entries are ordered according to ascending priority (unless reversed by using LegendBox.setReverse).

For a PieChart, the legend entry priority of a sector is $10000 \times (\text{sector_index} + 1)$. The Nth sector has sector_index (N - 1). That means the ordering of the sectors in the legend box is the same as the data array.

For an XYChart, the base legend entry priority for a layer is $10000 \times (\text{layer_index} + 1)$. Within a layer, the data set priority is $10 \times (\text{data_set_index} + 1)$. For example, the legend entry priority for the 5th dataset in the 3rd layer will be 30050. That means the ordering of the data sets in the legend box follows the order in which the layers are created. For data sets within the same layer, the ordering follows the order in which the data sets are added. The ordering can be modified by using Layer.setLegendOrder.

For an PolarChart, the legend entry priority of a layer is $10000 \times (\text{layer_index} + 1)$. That means the ordering of the layers in the legend box follows the order in which the layers are created.

For custom entries added using LegendBox.addKey2, you may control the legend entry priority. That means you may insert custom entries anywhere relative to the automatic entries added by ChartDirector.

3.26.10 getImageCoor(dataItem as Integer, offsetX as Integer = 0, offsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the image map coordinates of an legend entry as HTML image map attributes.

Notes: The image map coordinates will be of the following format:

```
shape="rect" cords=" [ x1 ] , [ y1 ] , [ x2 ] , [ y2 ] "
```

where (x1, y1) and (x2, y2) are opposite corners of the box that enclosed the legend entry. The format is specially designed so that it can easily be included into HTML image maps.

This method should be called only after creating the chart image (eg. using `BaseChart.makeChart`, `BaseChart.makeChart2` or `BaseChart.makeChart3`). The image map cannot be determined without creating the chart image first.

Argument	Default	Description
<code>dataItem</code>	(Mandatory)	The legend entry number, starting from 0. The first legend entry is 0. The nth legend entry is (n-1).
<code>offsetX</code>	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
<code>offsetY</code>	0	An offset to be added to all y coordinates in the image map. See <code>offsetX</code> above for description.

Return Value

A text string representing the coordinates of the legend entry in HTML image map attribute format.

3.26.11 `setCols(noOfCols as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the number of columns in the legend box.

Notes:

Argument	Default	Description
<code>noOfCols</code>	(Mandatory)	The number of columns in the legend box. The special value <code>AutoGrid</code> (= -2) means the number of columns is automatically determined. If this argument is 0, the legend box will use a flow layout (from left to right and then top to bottom, in which the entries may not be vertically aligned).

3.26.12 `setKeyBorder(edgeColor as color, raisedEffect as Integer = 0)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setKeyBorder` method, but uses color instead of integer data type for passing color values.

See also:

- 3.26.13 setKeyBorder(edgeColor as Integer, raisedEffect as Integer = 0) 505

3.26.13 setKeyBorder(edgeColor as Integer, raisedEffect as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the default border color of the legend icon.

Notes: This method applies only to data sets of which the icons are rectangles showing the colors of the data sets. If the icon is a shape or symbol (such as for a line with data point symbols), the border color of the original shape or symbol is always used.

Argument	Default	Description
edgeColor	(Mandatory)	The border color.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument is also used to support Chart::glassEffect and Chart::softLighting effects.

See also:

- 3.26.12 setKeyBorder(edgeColor as color, raisedEffect as Integer = 0) 504

3.26.14 setKeySize(width as Integer = -1, height as Integer = -1, gap as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the legend icon and its distance from the legend text.

Notes: Each legend entry consists of an icon and a text description. By default, the size of the icon and its distance from the text is determined automatically based on font size. This method can be used to override the default.

Argument	Default	Description
width	(Mandatory)	The width of the legend icon in pixels.
height	-1	The height of the legend icon in pixels. -1 means the height is automatically determined.
gap	-1	The distance between the legend icon and the legend text in pixels. -1 means the distance is automatically determined.

3.26.15 setKeySpacing(keySpacing as Integer, lineSpacing as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the distance between two legend entries.

Notes: In horizontal legend layout, the legend entries will flow from left to right, top to bottom, with a horizontal gap between two legend entries. In vertical legend layout, the legend entries will flow from top to bottom, with one entry per line.

By default, the horizontal gap and the line spacing is automatically determined based on font size. This method can be used to set the horizontal gap and line spacing in pixels.

Argument	Default	Description
keySpacing	(Mandatory)	The horizontal gap between two legend entries for horizontal legend layout, expressed in pixels.
lineSpacing	-1	The line spacing between two lines in the legend box, expressed in pixels. -1 means the line spacing is automatically determined.

3.26.16 setLineStyleKey(b as boolean=true)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Determines whether to always use "line style legend keys" or not.

Notes: By default, ChartDirector will use a small colored square box as the legend key icon. This allows people to determine the colors for the data representations.

In some cases, colors alone cannot distinguish the data representations. For example, in a line chart with multiple lines, it is possible all lines are of the same color, and are distinguished by line styles (solid, dash, dotted, etc), or the lines may be distinguished by data symbols (small circles, squares, etc) on the lines.

For charts with non-solid lines (eg. dash, dotted, etc) or lines with data symbols, ChartDirector will use "line style legend keys" for the lines. A line style legend key consist a of line with an optional data symbol, which reflects the line style and data symbol representing the data series.

This method can be used to force ChartDirector to always use the "line style legend keys".

Arguments:

Argument	Default	Description
b	true	A true value means to always use "line style legend keys". A false value means to automatically determined if "line style legend keys" should be used.

3.26.17 setReverse(b as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Reverses the ordering of the legend entries.

Notes: By default, the legend entries are ordered based on ascending legend entry priority number. This method can be used to change the ordering to following descending legend entry priority number instead.

See LegendBox for more information on legend entry priority system.

Argument	Default	Description
b	true	A true value means the legend entries are ordered based on descending legend entry priority number. false value means the legend entries are ordered based on ascending legend entry priority number.

3.27 class CDLinearMeterMBS

3.27.1 class CDLinearMeterMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The LinearMeter class represents linear meters.

Notes: Subclass of the CDBaseMeterMBS class.

Blog Entries

- [ChartDirector Meters and Gauges](#)

Xojo Developer Magazine

- [21.6, page 92: From 0 to 100, Creating Gauges with MBS and ChartDirector by Stefanie Juchmes](#)
- [21.6, page 89: From 0 to 100, Creating Gauges with MBS and ChartDirector by Stefanie Juchmes](#)

3.27.2 Methods

3.27.3 addBar(startValue as Double, endValue as Double, colorvalue as color, effect as Integer = 0, roundedCorner as Integer = 0) as CDTextBoxMBS

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a bar to the meter.

Notes: This method adds an empty TextBox to the meter. It configures the textbox position, size and color so that it looks like a bar on the meter scale.

Argument	Default	Description
startValue	(Mandatory)	The start value of the bar.
endValue	(Mandatory)	The end value of the bar.
color	(Mandatory)	The color of the bar.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument can also be used to specify Chart::glassEffect, Chart::softLighting, Chart::cylinderEffect or Chart::flatBorder effects.
roundedCorners	0	The radius for the two corners at the end position of the bar.

Returns a CDTextBoxMBS object representing the bar. This may be used to fine-tune the appearance of the bar.

See also:

- [3.27.4 addBar\(startValue as Double, endValue as Double, colorvalue as Integer, effect as Integer = 0, roundedCorner as Integer = 0\) as CDTextBoxMBS](#) 509

3.27.4 addBar(startValue as Double, endValue as Double, colorvalue as Integer, effect as Integer = 0, roundedCorner as Integer = 0) as CDTextBoxMBS

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a bar to the meter.

Notes: This method adds an empty TextBox to the meter. It configures the textbox position, size and color so that it looks like a bar on the meter scale.

Argument	Default	Description
startValue	(Mandatory)	The start value of the bar.
endValue	(Mandatory)	The end value of the bar.
color	(Mandatory)	The color of the bar.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument can also be used to specify Chart::glassEffect, Chart::softLighting, Chart::cylinderEffect or Chart::flatBorder effects.
roundedCorners	0	The radius for the two corners at the end position of the bar.

Returns a CDTextBoxMBS object representing the bar. This may be used to fine-tune the appearance of the bar.

See also:

- 3.27.3 addBar(startValue as Double, endValue as Double, colorvalue as color, effect as Integer = 0, roundedCorner as Integer = 0) as CDTextBoxMBS 508

3.27.5 addZone(startValue as Double, endValue as Double, colorvalue as color, label as string = "") as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addZone method, but uses color instead of integer data type for passing color values.

See also:

- 3.27.6 addZone(startValue as Double, endValue as Double, colorvalue as Integer, label as string = "") as CDTextBoxMBS 509

3.27.6 addZone(startValue as Double, endValue as Double, colorvalue as Integer, label as string = "") as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a zone to the meter.

Notes:

Argument	Default	Description
startValue	(Mandatory)	The start value (the lower bound) for the zone.
endValue	(Mandatory)	The end value (the upper bound) for the zone.
color	(Mandatory)	The color of the zone.
label	""	The text to be put at the center of the zone.

Return Value

A TextBox object representing the label. This may be used to fine-tune the appearance of the label.

See also:

- 3.27.5 addZone(startValue as Double, endValue as Double, colorvalue as color, label as string = "") as CDTextBoxMBS 509

3.27.7 Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other Constructor method, but uses color instead of integer data type for passing color values.

See also:

- 3.27.8 Constructor(width as Integer, height as Integer, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0) 510

3.27.8 Constructor(width as Integer, height as Integer, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new LinearMeter object.

Notes:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	BackgroundColor	The background color of the chart.
edgeColor	Transparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

See also:

- 3.27.7 Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0) 510

3.27.9 setMeter(leftX as Integer, topY as Integer, width as Integer, height as Integer, axisPos as Integer = 4, isReversed as boolean=false)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the orientation and position of the meter plot area.

Notes: The plot area is a rectangular region of the linear meter. The meter scale (the labels showing the values) will be on one external side of the plot area (say on the left side). The pointer start on the opposite side and points towards the meter scale.

This method defines the position and size of the rectangular region, as well as which side the meter scale is on. If the scale is on the left or right side, the meter is assumed to be a horizontal linear meter. If the scale is on the top or bottom side, the meter is assumed to be a vertical linear meter.

Argument	Default	Description
leftX	(Mandatory)	The x-coordinate of the top-left corner of the meter plot area.
topY	(Mandatory)	The y-coordinate of the top-left corner of the meter plot area.
width	(Mandatory)	The horizontal width of the meter plot area.
height	(Mandatory)	The vertical height of the meter plot area.
axisPos	Left	The position of the meter scale. The scale should be on one of the 4 sides of the plot area border, specified by Left, Top, Right and Bottom.
isReversed	false	By default, if the meter is horizontal, the meter scale will run from left to right. If the meter is vertical, the meter scale will run from bottom to top. If this argument is set to true, the meter scale will be reversed.

3.27.10 setRail(railColor as color, railWidth as Integer = 2, railOffset as Integer = 6)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setRail method, but uses color instead of integer data type for passing color values.

See also:

- 3.27.11 setRail(railColor as Integer, railWidth as Integer = 2, railOffset as Integer = 6) 511

3.27.11 setRail(railColor as Integer, railWidth as Integer = 2, railOffset as Integer = 6)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position, width and color of the pointer rail.

Notes: The pointer rail is the locus traced out by the starting point of the meter pointer as the pointer slides on the meter. The pointer rail is on the opposite side of the meter scale.

Argument	Default	Description
railColor	(Mandatory)	The color of the pointer rail.
railWidth	2	The line width of the the pointer rail in pixels.
railOffset	6	The gap between the pointer rail and the plot area in pixels. A negative value will mean the pointer rail may move within the plot area.

See also:

- 3.27.10 setRail(railColor as color, railWidth as Integer = 2, railOffset as Integer = 6) 511

3.28 class CDLineLayerMBS

3.28.1 class CDLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The LineLayer class represents line layers.

Notes: Subclass of the CDLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [Charts with more than one x or y axis](#)
- [A special chart with our MBS REALbasic ChartDirector Plugin](#)

Xojo Developer Magazine

- [7.4, page 34: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [18.5, page 72: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.28.2 Methods

3.28.3 `getLine(dataSet as Integer = 0)` as CDLineObjMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves an opaque LineObj representing a line in the line layer. The opaque LineObj is to be used in XYChart.addInterLineLayer for adding coloring between lines.

Notes:

Argument	Default	Description
dataSet	0	The data set number for the line. The first data set is 0. The nth data set is (n - 1).

Return Value

An opaque LineObj representing the requested line.

3.28.4 `setFastLineMode(b as boolean = true)`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables or disables fast line mode.

Notes: In a chart, the plot area width is usually a few hundred to around 2000 pixels. For a line chart that "flows" horizontally (eg. from left to right), if there are a lot more data points (eg. 100000 data points than the plot area pixel width, many of the points would be at the same x-pixel coordinates. The line segments joining the data points at the same x-pixel coordinates would just be oscillating up and down, overlapping themselves. What is visible at each x-pixel position is essentially a line joining the minimum point and the maximum point.

In fast line mode, ChartDirector will automatically detect that there are too many points in the same x-pixel coordinate, and draws only a line segment joining the extreme points at that coordinate. The resulting line is visually indistinguishable from a line in which the line segments join all data points. The fast line mode can significantly reduce the number of points drawn without affecting the chart appearance.

The fast line mode has no effect if there are not significantly more data points than the pixel width of the plot area. For charts with evenly spaced data points, the fast line mode would only act on x-pixel positions with too many data points. Thus the fast line mode can be enabled on line charts regardless of the number of data points.

The fast line mode should not be used if the data points are not flowing horizontally (or vertically if XY-Chart.swapXY is in effect), but are flowing in random directions.

Argument	Default	Description
b	true	A true value enables fast line mode. A false value disables fast line mode.

3.28.5 setGapColor(lineColor as color, lineWidth as Integer = -1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setGapColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.28.6 setGapColor(lineColor as Integer, lineWidth as Integer = -1) 514

3.28.6 setGapColor(lineColor as Integer, lineWidth as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color and style of the line used for jumping across NoValue data points.

Notes: By default, the color of the line for jumping across NoValue data points is Transparent, which means the line will become discontinuous.

Argument	Default	Description
lineColor	(Mandatory)	The line color of the line used for jumping across NoValue data points
lineWidth	-1	The line width of the line used for jumping across NoValue data points. -1 means the width will be the same as the line width of the line for drawing normal data points.

See also:

- 3.28.5 setGapColor(lineColor as color, lineWidth as Integer = -1) 514

3.28.7 setImageMapWidth(Width as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the effective width of the line used for producing image maps.

Notes: For thin lines, it is hard to click on the lines. So for the purpose of producing image maps for a line chart, ChartDirector can assume the line is very thick. The default is 10 pixels.

Argument	Default	Description
width	(Mandatory)	The effective width of the line used for producing image maps.

3.28.8 setSymbolScale(zDataX() as Double, scaleTypeX as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the symbol for each data point (for creating bubble charts).

Notes: One common usage for this method is to draw circle symbols of different sizes at each data points, creating a bubble chart.

This method supports any valid data symbols. You can create bubble charts with square bubbles, or even custom data symbols. Also, this method supports independent x and y direction sizing, so you can create bubbles elliptical in shape, and use the horizontal and vertical radius to represent different data.

ChartDirector supports specifying sizes as pixels or in axis scale. The unit is specified by using the following predefined constants.

ConstantValueDescription

See also:

PixelScale	0	The unit is measured in pixels.
XAxisScale	1	The unit is measured in x-axis scale.
YAxisScale	2	The unit is measured in y-axis scale.

Argument	Default	Description
zDataX	(Mandatory)	The sizes of the symbols at the x-axis direction, expressed using the unit defined by the scaleTypeX argument.
scaleTypeX	PixelScale	The unit for zDataX, which must be one of the predefined constants in the table above.
zDataY	[Empty_Array]	The sizes of the symbols at the y-axis direction, expressed using the unit defined by the scaleTypeY argument. An empty array means the sizes at the y-axis direction are the means as the sizes at the x-axis direction.
scaleTypeY	PixelScale	The unit for zDataY, which must be one of the predefined constants in the table above.

- 3.28.9 setSymbolScale(zDataX() as Double, scaleTypeX as Integer, zDataY() as Double, scaleTypeY as Integer = 0) 516

3.28.9 setSymbolScale(zDataX() as Double, scaleTypeX as Integer, zDataY() as Double, scaleTypeY as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the symbol for each data point (for creating bubble charts).

Notes: One common usage for this method is to draw circle symbols of different sizes at each data points, creating a bubble chart.

This method supports any valid data symbols. You can create bubble charts with square bubbles, or even custom data symbols. Also, this method supports independent x and y direction sizing, so you can create bubbles elliptical in shape, and use the horizontal and vertical radius to represent different data.

ChartDirector supports specifying sizes as pixels or in axis scale. The unit is specified by using the following predefined constants.

ConstantValueDescription

PixelScale	0	The unit is measured in pixels.
XAxisScale	1	The unit is measured in x-axis scale.
YAxisScale	2	The unit is measured in y-axis scale.

See also:

Argument	Default	Description
zDataX	(Mandatory)	The sizes of the symbols at the x-axis direction, expressed using the unit defined by the scaleTypeX argument.
scaleTypeX	PixelScale	The unit for zDataX, which must be one of the predefined constants in the table above.
zDataY	[Empty_Array]	The sizes of the symbols at the y-axis direction, expressed using the unit defined by the scaleTypeY argument. An empty array means the sizes at the y-axis direction are the means as the sizes at the x-axis direction.
scaleTypeY	PixelScale	The unit for zDataY, which must be one of the predefined constants in the table above.

- 3.28.8 setSymbolScale(zDataX() as Double, scaleTypeX as Integer = 0)

3.29 class CDLineMBS

3.29.1 class CDLineMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Line class represents straight lines.

Notes: Subclass of the CDDrawObjMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.29.2 Methods

3.29.3 setColor(c as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.29.4 setColor(c as Integer) 518

3.29.4 setColor(c as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color of the line.

Notes: By default, the color of the line is LineColor. To draw a dash line, you can use a dash line color (created using BaseChart.dashLineColor or DrawArea.dashLineColor).

Argument	Default	Description
c	(Mandatory)	The color of the line.

See also:

- 3.29.3 setColor(c as color) 518

3.29.5 setPos(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the end-points of the line..

Notes:

Argument	Default	Description
x1	(Mandatory)	The x coordinate of the first end-point of the line.
y1	(Mandatory)	The y coordinate of the first end-point of the line.
x2	(Mandatory)	The x coordinate of the second end-point of the line.
y2	(Mandatory)	The y coordinate of the second end-point of the line.

3.29.6 setWidth(w as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the width of the line in pixels.

Notes:

Argument	Default	Description
w	(Mandatory)	The width (thickness) of the line in pixels.

3.30 class CDLineObjMBS

3.30.1 class CDLineObjMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The line object class.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.30.2 Methods

3.30.3 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.31 class CDMarkMBS

3.31.1 class CDMarkMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Mark class represents mark lines.

Notes: Subclass of the CDTextBoxMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Xojo Developer Magazine

- [7.4, page 34: Easy Charts and Graphs, Using the ChartDirector Plugin](#)

3.31.2 Methods

3.31.3 getLine as CDLineObjMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves an opaque LineObj representing the mark line. The opaque LineObj is to be used in XYChart.addInterLineLayer for adding coloring between lines.

Notes: Return Value

An opaque LineObj representing the mark line.

3.31.4 setDrawOnTop(b as boolean)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Determine whether the mark line is drawn at the front of the chart layers, or at the back of the plot area (that is, like grid lines).

Notes:

Argument	Default	Description
b	(Mandatory)	A true value means drawing the mark line at the front of the chart layers. A false value means drawing the mark line at the back of the plot area.

3.31.5 setLineWidth(width as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the line width of the mark line.

Notes:

Argument	Default	Description
w	(Mandatory)	The mark line width in pixels.

3.31.6 setMarkColor(lineColor as color, textColor as color, tickColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setMarkColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.31.7 setMarkColor(lineColor as Integer, textColor as Integer = -1, tickColor as Integer = -1) 522

3.31.7 setMarkColor(lineColor as Integer, textColor as Integer = -1, tickColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the line, text and tick colors of the mark line.

Notes:

Argument	Default	Description
lineColor	(Mandatory)	The color of the mark line.
textColor	-1	The color of the text label that will be shown on the axis. -1 means the text label color is the same as the line color.
tickColor	-1	The color of the tick that will be shown on the axis. -1 means the tick color is the same as the line color.

See also:

- 3.31.6 setMarkColor(lineColor as color, textColor as color, tickColor as color) 522

3.31.8 setValue(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the value of the mark line.

Notes:

Argument	Default	Description
value	(Mandatory)	The value of the mark.

3.32 class CDMeterPointerMBS

3.32.1 class CDMeterPointerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The MeterPointer class represents meter pointers.

Notes: Subclass of the CDDrawObjMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [ChartDirector Meters and Gauges](#)

Xojo Developer Magazine

- [21.6, pages 91 to 92: From 0 to 100, Creating Gauges with MBS and ChartDirector by Stefanie Juchmes](#)

3.32.2 Methods

3.32.3 setColor(fillColor as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setColor method, but uses color instead of integer data type for passing color values.

See also:

- [3.32.4 setColor\(fillColor as Integer, edgeColor as Integer = -1\)](#) 523

3.32.4 setColor(fillColor as Integer, edgeColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the fill and border colors of the meter pointer.

Notes:

Argument	Default	Description
fillColor	(Mandatory)	The fill color of the meter pointer.
edgeColor	-1	The border color of the meter pointer. The default value of -1 means it is the same as the fill color.

See also:

- [3.32.3 setColor\(fillColor as color, edgeColor as color\)](#) 523

3.32.5 setPos(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the value the pointer points to on the meter scale.

Notes:

Argument	Default	Description
value	(Mandatory)	The value the pointer points to.

3.32.6 setShape(pointerCoor() as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of the meter pointer to a custom polygon.

Notes: The custom polygon is specified as a list of numbers representing the (x, y) coordinates of polygon vertices.

For an angular meter, the polygon should be defined with a nominal dimension of 1000 units. ChartDirector will scale the polygon so that 1000 units will become the radius of the meter.

For a linear meter, the length of the pointer should be defined with a nominal dimension of 1000 units. ChartDirector will scale the polygon so that 1000 units will be the length required for the pointer to reach the meter scale. The width of the pointer will be in 0.1 pixel units (that is, 10 units = 1 pixel).

The coordinate system for defining the polygon is that the x-axis points from left to right, and the y-axis points from bottom to top. The polygon will be defined as a pointer with the pivot at the origin, pointing to the upward direction.

As an example, the coordinates of the standard diamond pointer for an angular meter are:

0, -100, -50, -50, 0, 1000, 50, -50

The coordinates for the standard pencil pointer for a linear meter are:

-30, 0, 30, 0, 30, 768, 0, 1000, -30, 768

After scaling the pointers based on the nominal units, ChartDirector will in addition applies the lengthRatio and widthRatio scaling factor to the pointer. These ratios allow you to change the pointer sizes without changing the polygon definition. For example, if you want the meter pointer to be only 80% of the radius in the angular meter, you can set the lengthRatio to 0.80.

See also:

Argument	Default	Description
pointerCoor	(Mandatory)	An array of numbers x0, y0, x1, y1, x2, y2, ..., representing the coordinates the polygon vertices.
lengthRatio	NoValue	The length ratio applies to the pointer. NoValue means the lengthRatio is not modified. The default is 1.0 for angular meters and 0.75 for linear meters.
widthRatio	NoValue	The width ratio applies to the pointer. NoValue means the widthRatio is not modified. The default is 1.0.

- 3.32.7 setShape(pointerCoor() as Integer, lengthRatio as Double) 525
- 3.32.8 setShape(pointerCoor() as Integer, lengthRatio as Double, widthRatio as Double) 526
- 3.32.9 setShape(pointerType as Integer) 527
- 3.32.10 setShape(pointerType as Integer, lengthRatio as Double) 528
- 3.32.11 setShape(pointerType as Integer, lengthRatio as Double, widthRatio as Double) 529

3.32.7 setShape(pointerCoor() as Integer, lengthRatio as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of the meter pointer to a custom polygon.

Notes: The custom polygon is specified as a list of numbers representing the (x, y) coordinates of polygon vertices.

For an angular meter, the polygon should be defined with a nominal dimension of 1000 units. ChartDirector will scale the polygon so that 1000 units will become the radius of the meter.

For a linear meter, the length of the pointer should be defined with a nominal dimension of 1000 units. ChartDirector will scale the polygon so that 1000 units will be the length required for the pointer to reach the meter scale. The width of the pointer will be in 0.1 pixel units (that is, 10 units = 1 pixel).

The coordinate system for defining the polygon is that the x-axis points from left to right, and the y-axis points from bottom to top. The polygon will be defined as a pointer with the pivot at the origin, pointing to the upward direction.

As an example, the coordinates of the standard diamond pointer for an angular meter are:

0, -100, -50, -50, 0, 1000, 50, -50

The coordinates for the standard pencil pointer for a linear meter are:

-30, 0, 30, 0, 30, 768, 0, 1000, -30, 768

After scaling the pointers based on the nominal units, ChartDirector will in addition applies the lengthRatio

and `widthRatio` scaling factor to the pointer. These ratios allow you to change the pointer sizes without changing the polygon definition. For example, if you want the meter pointer to be only 80% of the radius in the angular meter, you can set the `lengthRatio` to 0.80.

Argument	Default	Description
<code>pointerCoor</code>	(Mandatory)	An array of numbers <code>x0</code> , <code>y0</code> , <code>x1</code> , <code>y1</code> , <code>x2</code> , <code>y2</code> , ..., representing the coordinates the polygon vertices.
<code>lengthRatio</code>	NoValue	The length ratio applies to the pointer. NoValue means the <code>lengthRatio</code> is not modified. The default is 1.0 for angular meters and 0.75 for linear meters.
<code>widthRatio</code>	NoValue	The width ratio applies to the pointer. NoValue means the <code>widthRatio</code> is not modified. The default is 1.0.

See also:

- 3.32.6 `setShape(pointerCoor() as Integer)` 524
- 3.32.8 `setShape(pointerCoor() as Integer, lengthRatio as Double, widthRatio as Double)` 526
- 3.32.9 `setShape(pointerType as Integer)` 527
- 3.32.10 `setShape(pointerType as Integer, lengthRatio as Double)` 528
- 3.32.11 `setShape(pointerType as Integer, lengthRatio as Double, widthRatio as Double)` 529

3.32.8 `setShape(pointerCoor() as Integer, lengthRatio as Double, widthRatio as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of the meter pointer to a custom polygon.

Notes: The custom polygon is specified as a list of numbers representing the (x, y) coordinates of polygon vertices.

For an angular meter, the polygon should be defined with a nominal dimension of 1000 units. `ChartDirector` will scale the polygon so that 1000 units will become the radius of the meter.

For a linear meter, the length of the pointer should be defined with a nominal dimension of 1000 units. `ChartDirector` will scale the polygon so that 1000 units will be the length required for the pointer to reach the meter scale. The width of the pointer will be in 0.1 pixel units (that is, 10 units = 1 pixel).

The coordinate system for defining the polygon is that the x-axis points from left to right, and the y-axis points from bottom to top. The polygon will be defined as a pointer with the pivot at the origin, pointing to the upward direction.

As an example, the coordinates of the standard diamond pointer for an angular meter are:

0, -100, -50, -50, 0, 1000, 50, -50

The coordinates for the standard pencil pointer for a linear meter are:

-30, 0, 30, 0, 30, 768, 0, 1000, -30, 768

After scaling the pointers based on the nominal units, `ChartDirector` will in addition apply the `lengthRatio` and `widthRatio` scaling factor to the pointer. These ratios allow you to change the pointer sizes without changing the polygon definition. For example, if you want the meter pointer to be only 80% of the radius in the angular meter, you can set the `lengthRatio` to 0.80.

Argument	Default	Description
<code>pointerCoor</code>	(Mandatory)	An array of numbers <code>x0, y0, x1, y1, x2, y2, ...</code> , representing the coordinates the polygon vertices.
<code>lengthRatio</code>	NoValue	The length ratio applies to the pointer. NoValue means the <code>lengthRatio</code> is not modified. The default is 1.0 for angular meters and 0.75 for linear meters.
<code>widthRatio</code>	NoValue	The width ratio applies to the pointer. NoValue means the <code>widthRatio</code> is not modified. The default is 1.0.

See also:

- 3.32.6 `setShape(pointerCoor() as Integer)` 524
- 3.32.7 `setShape(pointerCoor() as Integer, lengthRatio as Double)` 525
- 3.32.9 `setShape(pointerType as Integer)` 527
- 3.32.10 `setShape(pointerType as Integer, lengthRatio as Double)` 528
- 3.32.11 `setShape(pointerType as Integer, lengthRatio as Double, widthRatio as Double)` 529

3.32.9 `setShape(pointerType as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of the meter pointer to one of the built-in shapes.

Notes: The built-in symbols are specified by using the following predefined constants as the `pointerType` argument.

ConstantValueDescription

The length and width of the pointer can be scaled by using the `lengthRatio` and `widthRatio` arguments. The meters above are using default length and width ratios, which are both 1 for angular meters, and 0.75 and 1 for linear meters.

DiamondPointer	0	The blue pointers in the meters above.
TriangularPointer	1	The purple pointers in the meters above.
ArrowPointer	2	The red pointers in the meters above.
ArrowPointer2	3	The yellow pointers in the meters above.
LinePointer	4	The green pointers in the meters above.
PencilPointer	5	The grey pointers in the meters above.

Argument	Default	Description
pointerType	(Mandatory)	One of the predefined pointer shape constants to specify the pointer shape to use.
lengthRatio	NoValue	The length ratio applies to the pointer. NoValue means the lengthRatio is not modified. The default is 1.0 for angular meters and 0.75 for linear meters.
widthRatio	NoValue	The width ratio applies to the pointer. NoValue means the widthRatio is not modified. The default is 1.0.

See also:

- 3.32.6 `setShape(pointerCoor() as Integer)` 524
- 3.32.7 `setShape(pointerCoor() as Integer, lengthRatio as Double)` 525
- 3.32.8 `setShape(pointerCoor() as Integer, lengthRatio as Double, widthRatio as Double)` 526
- 3.32.10 `setShape(pointerType as Integer, lengthRatio as Double)` 528
- 3.32.11 `setShape(pointerType as Integer, lengthRatio as Double, widthRatio as Double)` 529

3.32.10 `setShape(pointerType as Integer, lengthRatio as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of the meter pointer to one of the built-in shapes.

Notes: The built-in symbols are specified by using the following predefined constants as the `pointerType` argument.

ConstantValueDescription

DiamondPointer	0	The blue pointers in the meters above.
TriangularPointer	1	The purple pointers in the meters above.
ArrowPointer	2	The red pointers in the meters above.
ArrowPointer2	3	The yellow pointers in the meters above.
LinePointer	4	The green pointers in the meters above.
PencilPointer	5	The grey pointers in the meters above.

The length and width of the pointer can be scaled by using the `lengthRatio` and `widthRatio` arguments. The meters above are using default length and width ratios, which are both 1 for angular meters, and 0.75 and 1 for linear meters.

Argument	Default	Description
<code>pointerType</code>	(Mandatory)	One of the predefined pointer shape constants to specify the pointer shape to use.
<code>lengthRatio</code>	NoValue	The length ratio applies to the pointer. NoValue means the <code>lengthRatio</code> is not modified. The default is 1.0 for angular meters and 0.75 for linear meters.
<code>widthRatio</code>	NoValue	The width ratio applies to the pointer. NoValue means the <code>widthRatio</code> is not modified. The default is 1.0.

See also:

- 3.32.6 `setShape(pointerCoor() as Integer)` 524
- 3.32.7 `setShape(pointerCoor() as Integer, lengthRatio as Double)` 525
- 3.32.8 `setShape(pointerCoor() as Integer, lengthRatio as Double, widthRatio as Double)` 526
- 3.32.9 `setShape(pointerType as Integer)` 527
- 3.32.11 `setShape(pointerType as Integer, lengthRatio as Double, widthRatio as Double)` 529

3.32.11 `setShape(pointerType as Integer, lengthRatio as Double, widthRatio as Double)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of the meter pointer to one of the built-in shapes.

Notes: The built-in symbols are specified by using the following predefined constants as the `pointerType` argument.

ConstantValueDescription

<code>DiamondPointer</code>	0	The blue pointers in the meters above.
<code>TriangularPointer</code>	1	The purple pointers in the meters above.
<code>ArrowPointer</code>	2	The red pointers in the meters above.
<code>ArrowPointer2</code>	3	The yellow pointers in the meters above.
<code>LinePointer</code>	4	The green pointers in the meters above.
<code>PencilPointer</code>	5	The grey pointers in the meters above.

The length and width of the pointer can be scaled by using the `lengthRatio` and `widthRatio` arguments. The meters above are using default length and width ratios, which are both 1 for angular meters, and 0.75 and 1 for linear meters.

1 for linear meters.

Argument	Default	Description
pointerType	(Mandatory)	One of the predefined pointer shape constants to specify the pointer shape to use.
lengthRatio	NoValue	The length ratio applies to the pointer. NoValue means the lengthRatio is not modified. The default is 1.0 for angular meters and 0.75 for linear meters.
widthRatio	NoValue	The width ratio applies to the pointer. NoValue means the widthRatio is not modified. The default is 1.0.

See also:

- 3.32.6 `setShape(pointerCoor() as Integer)` 524
- 3.32.7 `setShape(pointerCoor() as Integer, lengthRatio as Double)` 525
- 3.32.8 `setShape(pointerCoor() as Integer, lengthRatio as Double, widthRatio as Double)` 526
- 3.32.9 `setShape(pointerType as Integer)` 527
- 3.32.10 `setShape(pointerType as Integer, lengthRatio as Double)` 528

3.32.12 `setShapeAndOffset(pointerCoor() as Integer)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of new style angular meter pointers to a custom polygon.

Notes: The custom polygon is specified as a list of numbers representing the (x, y) coordinates of the polygon vertices, with the x-axis pointing from left to right and the y-axis pointing from bottom to top. The direction of the pointer should be pointing upwards, with the starting point at $y = 0$, and the ending point at $y = 1000$.

As an example, the coordinates of the new style triangular pointer are:

```
-15, 0, 15, 0, 0, 1000, 1000
```

The coordinates for the new style line pointer are:

```
-5, 0, 5, 0, 5, 1000, -5, 1000
```

In actual usage, ChartDirector will rotate the polygon to point it to the desired value, and adjust the polygon size and position based on the `startOffset`, `endOffset` and `widthRatio` arguments. Please refer to `MeterPointer.setShapeAndOffset2` for the meaning of these arguments.

See also:

Argument	Default	Description
pointerCoord	(Mandatory)	An array of numbers x0, y0, x1, y1, x2, y2, ..., representing the coordinates the polygon vertices.
startOffset	NoValue	The position of the starting point as a ratio to the scale radius. NoValue means the position is automatically determined.
endOffset	NoValue	The position of the ending point as a ratio to the scale radius. NoValue means the position is automatically determined.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

- 3.32.13 setShapeAndOffset(pointerCoord() as Integer, startOffset as Double) 531
- 3.32.14 setShapeAndOffset(pointerCoord() as Integer, startOffset as Double, endOffset as Double) 532
- 3.32.15 setShapeAndOffset(pointerCoord() as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) 533
- 3.32.16 setShapeAndOffset(pointerType as Integer) 534
- 3.32.17 setShapeAndOffset(pointerType as Integer, startOffset as Double) 535
- 3.32.18 setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double) 536
- 3.32.19 setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) 537

3.32.13 setShapeAndOffset(pointerCoord() as Integer, startOffset as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of new style angular meter pointers to a custom polygon.

Notes: The custom polygon is specified as a list of numbers representing the (x, y) coordinates of the polygon vertices, with the x-axis pointing from left to right and the y-axis pointing from bottom to top. The direction of the pointer should be pointing upwards, with the starting point at y = 0, and the ending point at y = 1000.

As an example, the coordinates of the new style triangular pointer are:

```
-15, 0, 15, 0, 0, 1000, 1000
```

The coordinates for the new style line pointer are:

```
-5, 0, 5, 0, 5, 1000, -5, 1000
```

In actual usage, ChartDirector will rotate the polygon to point it to the desired value, and adjust the polygon size and position based on the startOffset, endOffset and widthRatio arguments. Please refer to

MeterPointer.setShapeAndOffset2 for the meaning of these arguments.

Argument	Default	Description
pointerCoor	(Mandatory)	An array of numbers x0, y0, x1, y1, x2, y2, ..., representing the coordinates the polygon vertices.
startOffset	NoValue	The position of the starting point as a ratio to the scale radius. NoValue means the position is automatically determined.
endOffset	NoValue	The position of the ending point as a ratio to the scale radius. NoValue means the position is automatically determined.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

See also:

- 3.32.12 setShapeAndOffset(pointerCoor() as Integer) 530
- 3.32.14 setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double) 532
- 3.32.15 setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) 533
- 3.32.16 setShapeAndOffset(pointerType as Integer) 534
- 3.32.17 setShapeAndOffset(pointerType as Integer, startOffset as Double) 535
- 3.32.18 setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double) 536
- 3.32.19 setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) 537

3.32.14 setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of new style angular meter pointers to a custom polygon.

Notes: The custom polygon is specified as a list of numbers representing the (x, y) coordinates of the polygon vertices, with the x-axis pointing from left to right and the y-axis pointing from bottom to top. The direction of the pointer should be pointing upwards, with the starting point at y = 0, and the ending point at y = 1000.

As an example, the coordinates of the new style triangular pointer are:

```
-15, 0, 15, 0, 0, 1000, 1000
```

The coordinates for the new style line pointer are:

-5, 0, 5, 0, 5, 1000, -5, 1000

In actual usage, ChartDirector will rotate the polygon to point it to the desired value, and adjust the polygon size and position based on the startOffset, endOffset and widthRatio arguments. Please refer to MeterPointer.setShapeAndOffset2 for the meaning of these arguments.

Argument	Default	Description
pointerCoor	(Mandatory)	An array of numbers x0, y0, x1, y1, x2, y2, ..., representing the coordinates the polygon vertices.
startOffset	NoValue	The position of the starting point as a ratio to the scale radius. NoValue means the position is automatically determined.
endOffset	NoValue	The position of the ending point as a ratio to the scale radius. NoValue means the position is automatically determined.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

See also:

- 3.32.12 setShapeAndOffset(pointerCoor() as Integer) 530
- 3.32.13 setShapeAndOffset(pointerCoor() as Integer, startOffset as Double) 531
- 3.32.15 setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) 533
- 3.32.16 setShapeAndOffset(pointerType as Integer) 534
- 3.32.17 setShapeAndOffset(pointerType as Integer, startOffset as Double) 535
- 3.32.18 setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double) 536
- 3.32.19 setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) 537

3.32.15 setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of new style angular meter pointers to a custom polygon.

Notes: The custom polygon is specified as a list of numbers representing the (x, y) coordinates of the polygon vertices, with the x-axis pointing from left to right and the y-axis pointing from bottom to top. The direction of the pointer should be pointing upwards, with the starting point at y = 0, and the ending point at y = 1000.

As an example, the coordinates of the new style triangular pointer are:

-15, 0, 15, 0, 0, 1000, 1000

The coordinates for the new style line pointer are:

-5, 0, 5, 0, 5, 1000, -5, 1000

In actual usage, `ChartDirector` will rotate the polygon to point it to the desired value, and adjust the polygon size and position based on the `startOffset`, `endOffset` and `widthRatio` arguments. Please refer to `MeterPointer.setShapeAndOffset2` for the meaning of these arguments.

Argument	Default	Description
<code>pointerCoord</code>	(Mandatory)	An array of numbers <code>x0, y0, x1, y1, x2, y2, ...</code> , representing the coordinates the polygon vertices.
<code>startOffset</code>	NoValue	The position of the starting point as a ratio to the scale radius. NoValue means the position is automatically determined.
<code>endOffset</code>	NoValue	The position of the ending point as a ratio to the scale radius. NoValue means the position is automatically determined.
<code>widthRatio</code>	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

See also:

- 3.32.12 `setShapeAndOffset(pointerCoord() as Integer)` 530
- 3.32.13 `setShapeAndOffset(pointerCoord() as Integer, startOffset as Double)` 531
- 3.32.14 `setShapeAndOffset(pointerCoord() as Integer, startOffset as Double, endOffset as Double)` 532
- 3.32.16 `setShapeAndOffset(pointerType as Integer)` 534
- 3.32.17 `setShapeAndOffset(pointerType as Integer, startOffset as Double)` 535
- 3.32.18 `setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double)` 536
- 3.32.19 `setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` 537

3.32.16 `setShapeAndOffset(pointerType as Integer)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of new style angular meter pointers.

Notes: New style pointers are designed to have configurable starting and ending points, as well as configurable width. In this documentation, the ending point refers to the "tip" of the pointer, while the starting point is the "base" of the pointer.

By default, the starting and ending points are at -0.15 and 0.95. It means the distance between the starting point and the center is 15% of the scale radius. The negative sign means that relative to the center, the starting point is at the opposite direction to the value the pointer is supposed to point to. Similarly, the distance between the ending point and the center is 95% of the scale radius, which means the ending point should be quite close to the outer rim of the meter scale.

Argument	Default	Description
pointerType	(Mandatory)	Should be TriangularPointer2 for new style triangular pointer, and LinePointer2 for new style line pointer.
startOffset	NoValue	The radius the pointer starts at, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined.
endOffset	NoValue	The radius the pointer ends at, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

See also:

- 3.32.12 `setShapeAndOffset(pointerCoor() as Integer)` 530
- 3.32.13 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double)` 531
- 3.32.14 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double)` 532
- 3.32.15 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` 533
- 3.32.17 `setShapeAndOffset(pointerType as Integer, startOffset as Double)` 535
- 3.32.18 `setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double)` 536
- 3.32.19 `setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` 537

3.32.17 `setShapeAndOffset(pointerType as Integer, startOffset as Double)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of new style angular meter pointers.

Notes: New style pointers are designed to have configurable starting and ending points, as well as configurable width. In this documentation, the ending point refers to the "tip" of the pointer, while the starting point is the "base" of the pointer.

By default, the starting and ending points are at -0.15 and 0.95. It means the distance between the starting point and the center is 15% of the scale radius. The negative sign means that relative to the center, the starting point is at the opposite direction to the value the pointer is supposed to point to. Similarly, the distance between the ending point and the center is 95% of the scale radius, which means the ending point

should be quite close to the outer rim of the meter scale.

Argument	Default	Description
pointerType	(Mandatory)	Should be TriangularPointer2 for new style triangular pointer, and LinePointer2 for new style line pointer.
startOffset	NoValue	The radius the pointer starts at, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined.
endOffset	NoValue	The radius the pointer ends at, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

See also:

- 3.32.12 `setShapeAndOffset(pointerCoor() as Integer)` 530
- 3.32.13 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double)` 531
- 3.32.14 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double)` 532
- 3.32.15 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` 533
- 3.32.16 `setShapeAndOffset(pointerType as Integer)` 534
- 3.32.18 `setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double)` 536
- 3.32.19 `setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` 537

3.32.18 `setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of new style angular meter pointers.

Notes: New style pointers are designed to have configurable starting and ending points, as well as configurable width. In this documentation, the ending point refers to the "tip" of the pointer, while the starting point is the "base" of the pointer.

By default, the starting and ending points are at -0.15 and 0.95. It means the distance between the starting point and the center is 15% of the scale radius. The negative sign means that relative to the center, the starting point is at the opposite direction to the value the pointer is supposed to point to. Similarly, the distance between the ending point and the center is 95% of the scale radius, which means the ending point should be quite close to the outer rim of the meter scale.

See also:

Argument	Default	Description
pointerType	(Mandatory)	Should be TriangularPointer2 for new style triangular pointer, and LinePointer2 for new style line pointer.
startOffset	NoValue	The radius the pointer starts at, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined.
endOffset	NoValue	he radius the pointer ends at, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

- 3.32.12 `setShapeAndOffset(pointerCoor() as Integer)` 530
- 3.32.13 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double)` 531
- 3.32.14 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double)` 532
- 3.32.15 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` 533
- 3.32.16 `setShapeAndOffset(pointerType as Integer)` 534
- 3.32.17 `setShapeAndOffset(pointerType as Integer, startOffset as Double)` 535
- 3.32.19 `setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)` 537

3.32.19 `setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double, widthRatio as Double)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the shape of new style angular meter pointers.

Notes: New style pointers are designed to have configurable starting and ending points, as well as configurable width. In this documentation, the ending point refers to the "tip" of the pointer, while the starting point is the "base" of the pointer.

By default, the starting and ending points are at -0.15 and 0.95. It means the distance between the starting point and the center is 15% of the scale radius. The negative sign means that relative to the center, the starting point is at the opposite direction to the value the pointer is supposed to point to. Similarly, the distance between the ending point and the center is 95% of the scale radius, which means the ending point should be quite close to the outer rim of the meter scale.

See also:

- 3.32.12 `setShapeAndOffset(pointerCoor() as Integer)` 530
- 3.32.13 `setShapeAndOffset(pointerCoor() as Integer, startOffset as Double)` 531

Argument	Default	Description
pointerType	(Mandatory)	Should be TriangularPointer2 for new style triangular pointer, and LinePointer2 for new style line pointer.
startOffset	NoValue	The radius the pointer starts at, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined.
endOffset	NoValue	The radius the pointer ends at, expressed as a ratio to the scale radius. NoValue means the radius is automatically determined.
widthRatio	NoValue	The width of the pointer relative to the default width. NoValue means the width is automatically determined.

- 3.32.14 setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double) 532
- 3.32.15 setShapeAndOffset(pointerCoor() as Integer, startOffset as Double, endOffset as Double, widthRatio as Double) 533
- 3.32.16 setShapeAndOffset(pointerType as Integer) 534
- 3.32.17 setShapeAndOffset(pointerType as Integer, startOffset as Double) 535
- 3.32.18 setShapeAndOffset(pointerType as Integer, startOffset as Double, endOffset as Double) 536

3.33 class CDMLTableMBS

3.33.1 class CDMLTableMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a legend table.

Notes: Subclass of the CDDrawObjMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.33.2 Methods

3.33.3 appendCol as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Appends a column.

3.33.4 appendRow as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Appends a row.

3.33.5 getCell(col as Integer, row as Integer) as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the textbox object for the cell.

Notes: The index of column and row is zero based.

3.33.6 getColCount as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of columns.

3.33.7 getColStyle(col as Integer) as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the column style.

Notes: col is zero based.

3.33.8 getColWidth(col as Integer) as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the width of a column.

Notes: col is zero based.

3.33.9 getHeight as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the height of the table.

3.33.10 getRowCount as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of row.

3.33.11 getRowHeight(row as Integer) as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the height of a row.

Notes: row is zero based.

3.33.12 getRowStyle(row as Integer) as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the row style.

Notes: row is zero based.

3.33.13 `getStyle` as `CDTextBoxMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the textbox defining the style of the table.

3.33.14 `getWidth` as `Integer`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the width of the table.

3.33.15 `insertCol(col as Integer)` as `CDTextBoxMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Inserts a column at the given column.

Notes: col is zero based.

3.33.16 `insertRow(row as Integer)` as `CDTextBoxMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Inserts a row on the given position.

Notes: row is zero based.

3.33.17 `layout`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Layouts the table, so you can get the size.

3.33.18 `setCell(col as Integer, row as Integer, width as Integer, height as Integer, text as string)` as `CDTextBoxMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the cell to the given text.

Notes: col and row are zero based.

3.33.19 `setPos(x as Integer, y as Integer, alignment as Integer = 7)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position of the table.

Notes: x and y are zero based.

3.33.20 `setText(col as Integer, row as Integer, text as string) as CDTextBoxMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the text of a cell.

Notes: col and row are zero based.

3.34 class CDMultiChartMBS

3.34.1 class CDMultiChartMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The MultiChart class represents multi-chart.

Notes: A multi-chart is a container to contain multiple charts. You can use a multi-chart to combine multiple BaseChart objects (or its subclass such as PieChart, XYChart, PolarChart, AngularMeter or LinearMeter) into one image.

Subclass of the CDBaseChartMBS class.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.2](#)
- [MBS Xojo / Real Studio plug-ins version 16.2](#)
- [MBS Xojo / Real Studio Plugins, version 16.2pr3](#)
- [MBS Real Studio Plugins, version 12.5pr8](#)

3.34.2 Methods

3.34.3 addChart(x as Integer, y as Integer, c as CDBaseChartMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a BaseChart object (or its subclass such as PieChart, XYChart, PolarChart, AngularMeter or LinearMeter) into the multi-chart.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of a point in the multi-chart that is to align with the top-left corner of the added BaseChart.
y	(Mandatory)	The y coordinate of a point in the multi-chart that is to align with the top-left corner of the added BaseChart.
c	(Mandatory)	The BaseChart object to be added to the multi-chart.

3.34.4 Constructor(width as Integer = 640, height as Integer = 480, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other Constructor method, but uses color instead of integer data type for passing color values.

See also:

- 3.34.5 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0) 544

3.34.5 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new MultiChart object.

Notes:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	BackgroundColor	The background color of the chart.
edgeColor	Transparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

See also:

- 3.34.4 Constructor(width as Integer = 640, height as Integer = 480, bgColor as color, edgeColor as color, raisedEffect as Integer = 0) 543

3.34.6 getChart(index as Integer) as CDBaseChartMBS

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the specified chart in the MultiChart.

Notes:

Argument	Default	Description
i	0	The index of the chart. The index of the first chart set added to the MultiChart is 0. The index of the Nth data set added to the MultiChart is N - 1.

3.34.7 getChartCount as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the number of charts in the MultiChart.

Notes: Returns the number of charts in the MultiChart.

3.34.8 setMainChart(c as CDBaseChartMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the chart that receives view port mouse actions (for zooming and scrolling support).

Notes:

Argument	Default	Description
c	(Mandatory)	The chart that is to receive view port mouse actions.

3.35 class CDMultiPagePDFMBS

3.35.1 class CDMultiPagePDFMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The MultiPagePDF class is a utility class used to create PDF documents with multiple pages.

Notes: MultiPagePDF works by generating a PDF page for CDBaseChartMBS or CDDrawAreaMBS added to it. The CDBaseChartMBS can be a CDMultiChartMBS, and can contain free form CDML text, tables, shape and images. In this way, CDMultiPagePDFMBS can create complete PDF reports.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.2](#)
- [MBS Xojo Plugins, version 21.2pr5](#)
- [ChartDirector 7 update](#)

3.35.2 Methods

3.35.3 addPage(chart as CDBaseChartMBS)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds a BaseChart object as a PDF page.

Notes: You can use BaseChart.setOutputOptions to configure the PDF page options, such as the page size, margins, etc.

MultiPagePDF will immediately generate the PDF page using the BaseChart object. You can immediately delete the BaseChart object after calling this method. That means you do not need to keep a lot of BaseChart objects in memory even if the PDF has a lot of pages.

Argument	Default	Description
c	(Mandatory)	The BaseChart object used to create the PDF page.

See also:

- [3.35.4 addPage\(DrawArea as CDDrawAreaMBS\)](#)

546

3.35.4 addPage(DrawArea as CDDrawAreaMBS)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds a DrawArea object as a PDF page.

Notes: You can use DrawArea.setOutputOptions to configure the PDF page options, such as the page size, margins, etc.

MultiPagePDF will immediately generate the PDF page using the DrawArea object. You can immediately delete the DrawArea object after calling this method. That means you do not need to keep a lot of DrawArea objects in memory even if the PDF has a lot of pages.

Argument	Default	Description
d	(Mandatory)	The DrawArea object used to create the PDF page.

See also:

- 3.35.3 addPage(chart as CDBaseChartMBS) 546

3.35.5 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Creates a MultiPagePDF object.

3.35.6 outPDF as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Outputs the PDF document in memory.

Notes: Returns a memory block containing the PDF document.

See also:

- 3.35.7 outPDF(file as FolderItem) as Boolean 547
- 3.35.8 outPDF(path as String) as Boolean 548

3.35.7 outPDF(file as FolderItem) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Outputs the PDF document as a file.

Notes:

A true value indicates no error. A false value indicates the operation is unsuccessful.

See also:

Argument	Default	Description
filename	(Mandatory)	The filename of the PDF file.

- 3.35.6 outPDF as String 547
- 3.35.8 outPDF(path as String) as Boolean 548

3.35.8 outPDF(path as String) as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Outputs the PDF document as a file.

Notes:

Argument	Default	Description
filename	(Mandatory)	The filename of the PDF file.

A true value indicates no error. A false value indicates the operation is unsuccessful.
See also:

- 3.35.6 outPDF as String 547
- 3.35.7 outPDF(file as FolderItem) as Boolean 547

3.36 class CDNotInitializedExceptionMBS

3.36.1 class CDNotInitializedExceptionMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Exception raised if method is called on an uninitialized object.

Notes: Please report this as it may be a bug.

Subclass of the RuntimeException class.

Blog Entries

- [MBS Real Studio Plugins, version 12.3pr17](#)

3.37 class CDPieChartMBS

3.37.1 class CDPieChartMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PieChart class represents pie charts.

Notes: Subclass of the CDBaseChartMBS class.

Blog Entries

- [Chart Diagrams with Xojo](#)
- [MBS Xojo Plugins, version 18.5pr4](#)
- [A chart similar to the one on Apple Watch](#)

Xojo Developer Magazine

- [7.4, page 29: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [20.4, pages 38 to 41: PDF Pie Charts, Adding Xojo Charts to Your PDFs by Stefanie Juchmes](#)
- [18.5, page 58: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)
- [18.5, page 52: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.37.2 Methods

3.37.3 Constructor(width as Integer = 640, height as Integer = 480, bgcolor as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other Constructor method, but uses color instead of integer data type for passing color values.

See also:

- [3.37.4 Constructor\(width as Integer = 640, height as Integer = 480, bgcolor as Integer = &hFFFF0000, edgeColor as Integer = &hFF000000, raisedEffect as Integer = 0\)](#) 550

3.37.4 Constructor(width as Integer = 640, height as Integer = 480, bgcolor as Integer = &hFFFF0000, edgeColor as Integer = &hFF000000, raisedEffect as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new PieChart object.

Notes:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	BackgroundColor	The background color of the chart.
edgeColor	Transparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

See also:

- 3.37.3 Constructor(width as Integer = 640, height as Integer = 480, bgcolor as color, edgeColor as color, raisedEffect as Integer = 0) 550

3.37.5 sector(sectorNo as Integer) as CDSectorMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the Sector object representing a single sector in the pie chart.

Notes:

Argument	Default	Description
sectorNo	(Mandatory)	The sector number of the sector to retrieve, starting from 0. The first sector is 0. The nth sector is (n-1).

Return Value

The requested Sector object.

3.37.6 set3D(depth as Integer = -1, angle as Double = -1, shadowMode as boolean=false)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds 3D effects to the pie.

Notes:

See also:

- 3.37.7 set3D(depths() as Double, angle as Double = 45, shadowMode as boolean=false) 552

Argument	Default	Description
depth	-1	The 3D depth of the pie in pixels. -1 means the depth is automatically determined.
angle	-1	The 3D view angle in degrees. Must be 0 - 90 for standard 3D mode, and 0 - 360 in shadow 3D mode. -1 means the angle is automatically determined.
shadowMode	false	Flag to indicate whether the pie is in standard 3D or shadow 3D mode. A true value means shadow 3D mode. A false value means standard 3D mode.

3.37.7 set3D(depths() as Double, angle as Double = 45, shadowMode as boolean=false)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds 3D effects to the pie, where each sector can have a different 3D depth.

Notes:

Argument	Default	Description
depths	(Mandatory)	An array of integers representing the 3D depths for the sectors.
angle	45	The 3D view angle in degrees. Must be 0 - 90 for standard 3D mode, and 0 - 360 in shadow 3D mode. -1 means the angle is automatically determined.
shadowMode	false	Flag to indicate whether the pie is in standard 3D or shadow 3D mode. A true value means shadow 3D mode. A false value means standard 3D mode.

See also:

- 3.37.6 set3D(depth as Integer = -1, angle as Double = -1, shadowMode as boolean=false) 551

3.37.8 setData(data() as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data used to draw the pie chart.

Notes:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data points.
labels	[Empty_Array]	An array of text strings representing the labels of the sectors. An empty array means no sector label.

See also:

- 3.37.9 setData(data() as Double, label() as string) 553

3.37.9 setData(data() as Double, label() as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data used to draw the pie chart.

Notes:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data points.
labels	[Empty_Array]	An array of text strings representing the labels of the sectors. An empty array means no sector label.

See also:

- 3.37.8 setData(data() as Double)

552

3.37.10 setDonutSize(x as Integer, y as Integer, r as Integer, r2 as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position and size of the donut in the donut chart.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of the donut center.
y	(Mandatory)	The y coordinate of the donut center.
r	(Mandatory)	The inner radius of the donut.
r2	(Mandatory)	The outer radius of the donut.

3.37.11 setExplode(sectorNo as Integer, distance as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Explode a sector from the pie.

Notes:

Argument	Default	Description
sectorNo	(Mandatory)	The sector number of the sector to be exploded from the pie, starting from 0. The first sector is 0. The nth sector is (n - 1).
distance	-1	The explosion distance in pixels. -1 means the distance is automatically determined.

3.37.12 `setExplodeGroup(startSector as Integer, endSector as Integer, distance as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Explode a group of sectors from the pie.

Notes:

Argument	Default	Description
<code>startSector</code>	(Mandatory)	The sector number of the first sector in the sector group to be exploded. The first sector is 0. The nth sector is (n - 1).
<code>endSector</code>	(Mandatory)	The sector number of the last sector in the sector group to be exploded. The first sector is 0. The nth sector is (n - 1).
<code>distance</code>	-1	The explosion distance in pixels. -1 means the distance is automatically determined.

3.37.13 `setJoinLine(joinLineColor as color, joinLineWidth as Integer = -1)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setJoinLine` method, but uses color instead of integer data type for passing color values.

See also:

- 3.37.14 `setJoinLine(joinLineColor as Integer, joinLineWidth as Integer = -1)` 554

3.37.14 `setJoinLine(joinLineColor as Integer, joinLineWidth as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color and width of the join lines used to connect the sector labels to the sector perimeter.

Notes: This method affects all sectors. To set the color and width of the join line for one particular sector only, use `Sector.setJoinLine`.

By default, for circular label layout, the join line color is `Transparent`. For side label layout, the join line color is `SameAsMainColor`.

Argument	Default	Description
<code>joinLineColor</code>	(Mandatory)	The color of the line that joins the sector perimeter with the sector label.
<code>joinLineWidth</code>	1	The line width of the join line.

See also:

- 3.37.13 `setJoinLine(joinLineColor as color, joinLineWidth as Integer = -1)` 554

3.37.15 setLabelFormat(formatString as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the format of the all sector labels.

Example:

```
dim c as CDPieChartMBS
```

```
// you can use label formats like this:
```

```
c.setLabelFormat("<*block,halign=left*><*font=timesbi.ttf,size=12,underline=1*>{ label } </font*><*br*>US$  
{ value } K ( { percent } %)")
```

```
// we can reduce that to this:
```

```
c.setLabelFormat(" { label } { value } { percent } %")
```

```
// and it shows 3 numbers. With | 1 after the variable name, we define the decimals after dot:
```

```
c.setLabelFormat(" { label } { value | 1 } { percent | 1 } %")
```

```
// and
```

```
c.setLabelFormat(" { label } { value | 1., } { percent | 1., } %")
```

```
// uses dot for thousands and comma for decimal separator.
```

Notes: This method affects all sectors. To set the label format for one particular sector only, use Sector.setLabelFormat.

The default sector label format depends on the label layout method used (see PieChart.setLabelLayout).

Label Layout Method Default Label Format

Circular Layout { label } <*br*> { percent } %

(The "<*br*>" above is the CDML syntax for a line break.)

Side Layout { label } ({ percent } %)

Please refer to Parameter Substitution and Formatting on all available parameters and how to format them.

Argument	Default	Description
formatString	(Mandatory)	The format string.

3.37.16 setLabelLayout(layoutMethod as Integer, pos as Integer = -1, topBound as Integer = -1, bottomBound as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the layout method and location of the sector labels.

Notes: This method affects all sectors. To set the sector label layout method and location for one particular sector only, use Sector.setLabelLayout.

ChartDirector supports two sector label layout methods - circular layout and side layout. The layout method is determined using the layoutMethod argument, which must be one of the following predefined constants.

Constant Value Description

SideLayout	0	Side Label Layout
CircleLayout	1	Circular Label Layout

In the circular layout method, the sector labels are positioned around the perimeter of the pie.

In the side layout method, there is an invisible rectangle containing the pie, where the rectangle can be wider than the diameter of the pie. The sector labels are positioned on the left and right sides outside the rectangle. Because the labels can be quite far away from the sectors, join lines are typically used to connect the labels to the sectors.

The circular layout method usually uses less space and is the default layout method. However, if the pie chart contains a lot of small sectors, the labels may overlap with each others, due to insufficient space on the pie perimeter to position the labels.

The side layout method has the advantages that it can avoid label overlapping. In the side layout method, labels will automatically shift up and down to avoid overlapping.

One common issue in pie charts is the data contain a lot of small sectors. If the data are sorted, the small sectors will be crowded together instead of distributed evenly. Although the side layout method can avoid label overlapping by shifting the labels up and down, some labels may need to be shifted great distances.

Label layout can often be improved if the small sectors are near the horizontal axis. It is because the amount of vertical label space for a sector is greatest at the horizontal axis. This can be achieved by choosing an appropriate start angle (using `PieChart.setStartAngle`).

If the data is in ascending order (small sectors crowded at the beginning), a start angle of 45 degrees with clockwise sector layout is recommended. With this setting, the first few sectors (the smallest sectors) will be at around 45 - 135 degrees, so is near the horizontal axis (90 degrees). Similarly, if the data is in descending order (small sectors crowded at the end), a start angle of 135 degrees with clockwise sector layout can be used.

Argument	Default	Description
<code>layoutMethod</code>	(Mandatory)	Specify the layout method. Must be one of the predefined constants <code>CircleLayout</code> or <code>SideLayout</code> .
<code>pos</code>	-1	For circular layout, it is the distance between the sector perimeter and the sector label. A negative value (but not -1) means the sector label will be drawn in the interior of the sector.

For side layout, it is the distance between the pie perimeter and the left or right edges of the invisible containing rectangle (equal to the width of the rectangle minus the pie diameter and then divided by 2).

In either case, -1 means the distance is automatically determined.

<code>topBound</code>	-1	This parameter applies only to side label layout. It controls the top bound of the labels (the minimum y coordinate), thereby limiting the extent that the labels can be shifted up to avoid overlapping. This parameter is typically used avoid the labels from moving up too much and overlap with other chart objects (such as the chart title).
<code>bottomBound</code>	-1	This parameter applies only to side label layout. It controls the bottom bound of the labels (the maximum y coordinate), thereby limiting the extent which the labels can be shifted down to avoid overlapping. This parameter is typically used to avoid the labels from moving down too much and overlap with other chart objects.

3.37.17 `setLabelPos(pos as Integer, joinLineColor as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setLabelPos` method, but uses color instead of integer data type for passing color values.

See also:

- 3.37.18 `setLabelPos(pos as Integer, joinLineColor as Integer = -1)`

3.37.18 setLabelPos(pos as Integer, joinLineColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the circular label layout method, and configure the join lines used to connect the sector labels to the sector perimeter.

Notes: This method affects all sectors. To set the sector label position or join line color for one particular sector only, use Sector.setLabelPos.

Argument	Default	Description
pos	(Mandatory)	The distance between the sector perimeter and the sector label. A negative value means the sector label will be drawn in the interior of the sector.
joinLineColor	-1	The color of the line that joins the sector perimeter with the sector label. The default is Transparent. The join line is ignored if the sector label is inside the sector.

See also:

- 3.37.17 setLabelPos(pos as Integer, joinLineColor as color) 557

3.37.19 setLabelStyle(font as string = "", fontsize as Double = 8, fontColor as Integer = &hfff0002) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the style used to draw all sector labels.

Notes: This method affects all sectors. To set the label style for one particular sector only, use Sector.setLabelStyle.

See Font Specification for details on various font attributes.

Argument	Default	Description
font	""	The font used to draw the sector labels.
fontSize	8	The font size in points.
fontColor	TextColor	The text color for the sector labels.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.37.20 setLabelStyle(font as string, fontsize as Double, fontColor as color) as CDTextBoxMBS 559

3.37.20 setLabelStyle(font as string, fontsize as Double, fontColor as color) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setLabelStyle method, but uses color instead of integer data type for passing color values.

See also:

- 3.37.19 setLabelStyle(font as string = "", fontsize as Double = 8, fontColor as Integer = &hfff0002) as CDTextBoxMBS 558

3.37.21 setLineColor(edgeColor as color, joinLineColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setLineColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.37.22 setLineColor(edgeColor as Integer, joinLineColor as Integer = -1) 559

3.37.22 setLineColor(edgeColor as Integer, joinLineColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the sector edge color and join line color.

Notes: This method affects all sectors. To set the sector label position and join line color for one particular sector, use Sector.setColor.

Argument	Default	Description
edgeColor	(Mandatory)	The colors for the edges of the sectors. By default, the edge color is SameAsMainColor, which means the edge color is the same as the fill color of the sector, and the sector will appear borderless.
joinLineColor	-1	The color of the line that join the sector perimeter with the sector label. By default, for circular label layout, the join line color is Transparent. For side label layout, the join line color is SameAsMainColor.

See also:

- 3.37.21 setLineColor(edgeColor as color, joinLineColor as color) 559

3.37.23 setPieSize(x as Integer, y as Integer, r as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position and size of the pie in the pie chart.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of the pie center.
y	(Mandatory)	The y coordinate of the pie center.
r	(Mandatory)	The radius of the pie.

3.37.24 setSectorStyle(shadingMethod as Integer, edgeColor as color, edgeWidth as Integer = -1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setSectorStyle method, but uses color instead of integer data type for passing color values.

See also:

- 3.37.25 setSectorStyle(shadingMethod as Integer, edgeColor as Integer = -1, edgeWidth as Integer = -1) 560

3.37.25 setSectorStyle(shadingMethod as Integer, edgeColor as Integer = -1, edgeWidth as Integer = -1)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the sector shading style, edge color and edge width.

Notes: This method affects all sectors. To set the sector shading style, edge color and edge width for one particular sector, use CDSectorMBS.setStyle.

ChartDirector supports various shading effects, which are best illustrated using examples.

The followings are textual descriptions of the shading effects.

Arguments:

See also:

- 3.37.24 setSectorStyle(shadingMethod as Integer, edgeColor as color, edgeWidth as Integer = -1) 560

Constant	Value	Description
DefaultShading	0	This is the default shading method. The top surfaces are shaded with flat colors. For 3D sectors, the cylindrical surfaces are shaded with cylindrical lighting effects.
FlatShading	1	All surfaces are shaded with flat colors.
LocalGradientShading	2	The top surfaces are shaded with linear gradient colors. For each sector, the gradient is from the bottom of the sector bounding box to the top of the sector bounding box, with the bottom side brighter and the top side darker. For 3D sectors, the cylindrical surfaces are shaded with cylindrical lighting effects.
GlobalGradientShading	3	The top surfaces are shaded with linear gradient colors. The gradient is from the bottom the pie bounding box to the top of the pie bounding box, with the bottom side brighter and the top side darker. For 3D sectors, the cylindrical surfaces are shaded with cylindrical lighting effects.
ConcaveShading	4	The top surfaces are shaded with a special effect so that they look concave, with the pie center appears to be depressed relative to the perimeter. For 3D sectors, the cylindrical surfaces are shaded with cylindrical lighting effects.
RoundedEdgeShading	6	The top surfaces are shaded with a special effect so that the pie looks raised with a rounded raised edge at the perimeter.
RadialShading	7	The top surfaces are shaded with radial gradient colors, with the pie center brighter and the perimeter darker. For 3D sectors, the cylindrical surfaces are shaded with cylindrical lighting effects.
RingShading	8	This effect is intended to be used with 2D donut charts only. The top surfaces are shaded with a special effect so that a 2D donut will look like a torus.

Argument	Default	Description
shadingMethod	(Mandatory)	The sector shading style to use, which must be one of the constants in the table above. -1 means to keep the existing value unchanged.
edgeColor	-1	The edge color of the sectors. -1 means to keep the existing value unchanged.
edgeWidth	-1	The edge width of the sectors. -1 means to keep the existing value unchanged.

3.37.26 setStartAngle(startAngle as Double, clockWise as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the angle of the first sector in the pie and the layout direction for the sectors.

Notes: By default, the start angle is 0 degree (the upward pointing direction), and subsequent sectors are drawn clockwise.

Argument	Default	Description
startAngle	(Mandatory)	The angle to start drawing the first sector in degrees. The angle is measured from the upward pointing direction in the clockwise direction.
clockWise	true	Flag to control the layout direction of the sectors. A true value means clockwise. A false value means anti-clockwise.

3.38 class CDPlotAreaMBS

3.38.1 class CDPlotAreaMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PlotArea class represents plot areas in XY charts.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [Runtime error with DLLs](#)

Xojo Developer Magazine

- [7.4, page 34: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [20.1, page 43: Cool Charts and Heatmaps, Using Monkeybread Software's ChartDirector Plugin by Stefanie Juchmes](#)
- [18.5, page 74: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.38.2 Methods

3.38.3 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.38.4 getBottomY as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the bottom y pixel coordinate of the plot area.

Notes: Returns the bottom y pixel coordinate of the plot area.

3.38.5 getHeight as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the height of the plot area.

3.38.6 getLeftX as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the left x pixel coordinate of the plot area.

3.38.7 getRightX as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the right x pixel coordinate of the plot area.

Notes: Returns the right x pixel coordinate of the plot area.

3.38.8 getTopY as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the top y pixel coordinate of the plot area.

Notes: In some cases, the top y coordinate of a box may be dynamically determined. An example is the top y coordinate of an `LegendBox` with alignment set to `Center`. To determine the top y coordinate, the size of the box must be known first. For these cases, the top y coordinate is undefined until the legend box or the entire chart has been laid out (using `CDBaseChartMBS.layout` or `CDBaseChartMBS.layoutLegend`), or the chart image has been drawn (eg. using `CDBaseChartMBS.makeChart`).

Arguments:

None

Return Value

The top y pixel coordinate of the plot area.

3.38.9 getWidth as Integer

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the width of the plot area.

3.38.10 moveGridBefore(layer as CDLayerMBS=nil)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Moves the grid lines in front of a Layer.

Notes: By default, the grid lines will be drawn at the back of the plot area, behind all the layers. This method can be used to move the grid lines in front of a given layer.

Arguments:

Argument	Default	Description
layer	nil	The Layer for the grid lines to move in front of.

3.38.11 set4QBgColor(Q1Color as color, Q2Color as color, Q3Color as color, Q4Color as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other set4QBgColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.38.12 set4QBgColor(Q1Color as Integer, Q2Color as Integer, Q3Color as Integer, Q4Color as Integer, edgeColor as Integer = -1) 564

3.38.12 set4QBgColor(Q1Color as Integer, Q2Color as Integer, Q3Color as Integer, Q4Color as Integer, edgeColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets using 4 background plot area colors for 4 quadrants.

Notes: The first quadrant is the region defined by $x \geq 0$ and $y \geq 0$. The second quadrant is the region defined by $x < 0$ and $y \geq 0$. The third quadrant is the region defined by $x < 0$ and $y < 0$. The fourth quadrant is the region defined by $x \geq 0$ and $y < 0$. These 4 regions can be colored using 4 different background colors.

Argument	Default	Description
Q1Color	(Mandatory)	The background color to be used for the first quadrant.
Q2Color	(Mandatory)	The background color to be used for the second quadrant.
Q3Color	(Mandatory)	The background color to be used for the third quadrant.
Q4Color	(Mandatory)	The background color to be used for the fourth quadrant.
edgeColor	-1	The border color of the plot area. -1 means to use the default, which is LineColor. However, if the axes are configured in 4 quadrant mode (see XYChart.setAxisAtOrigin), the default will change to Transparent.

See also:

- 3.38.11 set4QBgColor(Q1Color as color, Q2Color as color, Q3Color as color, Q4Color as color, edgeColor as color) 564

3.38.13 setAltBgColor(horizontal as Boolean, color1 as color, color2 as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setAltBgColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.38.14 setAltBgColor(horizontal as Boolean, color1 as Integer, color2 as Integer, edgeColor as Integer = -1) 565

3.38.14 setAltBgColor(horizontal as Boolean, color1 as Integer, color2 as Integer, edgeColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets alternating plot area background color.

Notes: This method can be used to specify two colors that will be used alternatively to draw horizontal or vertical bands on the plot area background, using major grid lines as boundaries for the bands.

Argument	Default	Description
horizontal	(Mandatory)	true means to use horizontal bands. false means to use vertical bands.
color1	(Mandatory)	The first color to be used as the alternating background color.
color2	(Mandatory)	The second color to be used as the alternating background color.
edgeColor	-1	The border color of the plot area. -1 means to use the default, which is LineColor. However, if the axes are configured in 4 quadrant mode (see XYChart.setAxisAtOrigin), the default will change to Transparent.

See also:

- 3.38.13 setAltBgColor(horizontal as Boolean, color1 as color, color2 as color, edgeColor as color) 565

3.38.15 setBackground(colorvalue as color, altBgColor as color, edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setBackground method, but uses color instead of integer data type for passing color values.

See also:

- 3.38.16 `setBackground(colorvalue as Integer, altBgColor as Integer = -1, edgeColor as Integer = -1)`
566
- 3.38.17 `setBackground(file as folderitem, align as Integer = 5)` 566

3.38.16 `setBackground(colorvalue as Integer, altBgColor as Integer = -1, edgeColor as Integer = -1)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the background colors and the border color of the plot area.

Notes: A plot area can have one or two background colors. If it has two background colors, they are drawn alternatively as horizontal bands on the major background grid.

ChartDirector Ver 4.0 introduces two new methods `PlotArea.setAltBgColor` and `PlotArea.set4QBgColor`. They are used for vertical alternating bands, and for supporting 4 background colors for 4 quadrants.

Argument	Default	Description
<code>color</code>	(Mandatory)	The background color.
<code>altBgColor</code>	-1	The second background color. -1 means there is no second background color.
<code>edgeColor</code>	-1	The border color of the plot area. -1 means to use the default, which is <code>LineColor</code> . However, if the axes are configured in 4 quadrant mode (see <code>XYChart.setAxisAtOrigin</code>), the default will change to <code>Transparent</code> .

See also:

- 3.38.15 `setBackground(colorvalue as color, altBgColor as color, edgeColor as color)` 565
- 3.38.17 `setBackground(file as folderitem, align as Integer = 5)` 566

3.38.17 `setBackground(file as folderitem, align as Integer = 5)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the specified image file as the background image of the plot area.

Notes: ChartDirector will automatically detect the image file format using the file extension, which must either `png`, `jpg`, `jpeg`, `gif`, `wbmp` or `wmp` (case insensitive).

Please refer to `BaseChart.setSearchPath` on the directory that ChartDirector will search for the file.

See also:

- 3.38.15 `setBackground(colorvalue as color, altBgColor as color, edgeColor as color)` 565

Argument	Default	Description
img	(Mandatory)	The image file that is used as the background image of the plot area.
align	Center	The alignment of the background image relative to the plot area. See Alignment Specification for supported alignment types.

- 3.38.16 setBackground(colorvalue as Integer, altBgColor as Integer = -1, edgeColor as Integer = -1)
566

3.38.18 setGridAxis(xGridAxis as CDAxisMBS, yGridAxis as CDAxisMBS)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the axis used for drawing grid lines.

Notes: By default, the grid lines will be based on the ticks on the primary x-axis (CDXYChartMBS.xAxis) and primary y-axis (CDXYChartMBS.yAxis). This method may be used to specify alternative x-axis and y-axis to base the grid lines on.

Arguments:

Argument	Default	Description
xGridAxis	(Mandatory)	The x-axis used for drawing grid lines.
yGridAxis	(Mandatory)	The y-axis used for drawing grid lines.

3.38.19 setGridColor(hGridColor as color, vGridColor as color, minorHGridColor as color, minorVGridColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setGridColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.38.20 setGridColor(hGridColor as Integer, vGridColor as Integer = &hff000000, minorHGridColor as Integer = -1, minorVGridColor as Integer = -1)
567

3.38.20 setGridColor(hGridColor as Integer, vGridColor as Integer = &hff000000, minorHGridColor as Integer = -1, minorVGridColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the horizontal and vertical grid colors of the plot area.

Notes:

Argument	Default	Description
hGridColor	(Mandatory)	The color for the horizontal grid lines associated with major ticks.
vGridColor	Transparent	The color for the vertical grid lines associated with major ticks.
minorHGridColor	-1	The color for the horizontal grid lines associated with minor ticks. -1 means the color is the same as hGridColor.
minorVGridColor	-1	The color for the vertical grid lines associated with minor ticks. -1 means the color is the same as vGridColor.

See also:

- 3.38.19 setGridColor(hGridColor as color, vGridColor as color, minorHGridColor as color, minorVGridColor as color) 567

3.38.21 setGridWidth(hGridWidth as Integer, vGridWidth as Integer = -1, minorHGridWidth as Integer = -1, minorVGridWidth as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the horizontal and vertical grid line width.

Notes:

Argument	Default	Description
hGridWidth	(Mandatory)	The line width for the horizontal grid lines associated with major ticks.
vGridWidth	-1	The line width for the vertical grid lines associated with major ticks. -1 means the vertical grid line width is the same as hGridWidth.
minorHGridWidth	-1	The line width for the horizontal grid lines associated with minor ticks. -1 means the color is the same as hGridWidth.
minorVGridWidth	-1	The line width for the horizontal grid lines associated with minor ticks. -1 means the color is the same as vGridWidth.

3.39 class CDPolarAreaLayerMBS

3.39.1 class CDPolarAreaLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PolarAreaLayer class represents polar area layers.

Notes: Subclass of the CDPolarLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.40 class CDPolarChartMBS

3.40.1 class CDPolarChartMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PolarChart class represents polar charts (including radar charts).

Notes: Subclass of the CDBaseChartMBS class.

Blog Entries

- [MBS Xojo Plugins, version 21.2pr5](#)
- [MBS Xojo Plugins, version 18.2pr5](#)

Xojo Developer Magazine

- [7.5, pages 31 to 32: Easy Charts and Graphs Part 2, Using the ChartDirector Plugin](#)

3.40.2 Methods

3.40.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addAreaLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS 571
- 3.40.5 addAreaLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 571
- 3.40.6 addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS 572
- 3.40.7 addAreaLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 572
- 3.40.8 addAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarAreaLayerMBS 573

3.40.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar area layer to the polar chart.

Notes:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data points in the layer.
color	-1	The color to draw the area. -1 means that the color is automatically selected from the palette.
name	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Return Value

A PolarAreaLayer object representing the chart layer created.

See also:

- 3.40.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 570
- 3.40.5 addAreaLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 571
- 3.40.6 addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS 572
- 3.40.7 addAreaLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 572
- 3.40.8 addAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarAreaLayerMBS 573

3.40.5 addAreaLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Same as the other addAreaLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 570
- 3.40.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS 571

- 3.40.6 `addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarAreaLayerMBS` 572
- 3.40.7 `addAreaLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarAreaLayerMBS` 572
- 3.40.8 `addAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarAreaLayerMBS` 573

3.40.6 `addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarAreaLayerMBS`

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a polar area layer to the polar chart.

Notes:

Argument	Default	Description
<code>data</code>	(Mandatory)	An array of numbers representing the data points in the layer.
<code>color</code>	-1	The color to draw the area. -1 means that the color is automatically selected from the palette.
<code>name</code>	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Return Value

A `PolarAreaLayer` object representing the chart layer created.

See also:

- 3.40.3 `addAreaLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarAreaLayerMBS` 570
- 3.40.4 `addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarAreaLayerMBS` 571
- 3.40.5 `addAreaLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarAreaLayerMBS` 571
- 3.40.7 `addAreaLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarAreaLayerMBS` 572
- 3.40.8 `addAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarAreaLayerMBS` 573

3.40.7 `addAreaLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarAreaLayerMBS`

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar area layer to the polar chart.

See also:

- 3.40.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 570
- 3.40.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS 571
- 3.40.5 addAreaLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 571
- 3.40.6 addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS 572
- 3.40.8 addAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarAreaLayerMBS 573

3.40.8 addAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarAreaLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar area layer to the polar chart.

See also:

- 3.40.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 570
- 3.40.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS 571
- 3.40.5 addAreaLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 571
- 3.40.6 addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarAreaLayerMBS 572
- 3.40.7 addAreaLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarAreaLayerMBS 572

3.40.9 addLineLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addLineLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.10 `addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 574
- 3.40.11 `addLineLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 575
- 3.40.12 `addLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 575
- 3.40.13 `addLineLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 576
- 3.40.14 `addLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 576

3.40.10 `addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarLineLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar line layer to the polar chart.

Notes:

Argument	Default	Description
<code>data</code>	(Mandatory)	An array of numbers representing the data points in the layer.
<code>color</code>	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
<code>name</code>	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Return Value

A `PolarLineLayer` object representing the chart layer created.

See also:

- 3.40.9 `addLineLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 573
- 3.40.11 `addLineLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 575
- 3.40.12 `addLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 575
- 3.40.13 `addLineLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 576
- 3.40.14 `addLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 576

3.40.11 addLineLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Same as the other addLineLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.9 addLineLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarLineLayerMBS 573
- 3.40.10 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarLineLayerMBS 574
- 3.40.12 addLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarLineLayerMBS 575
- 3.40.13 addLineLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarLineLayerMBS 576
- 3.40.14 addLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarLineLayerMBS 576

3.40.12 addLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarLineLayerMBS

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a polar line layer to the polar chart.

Notes:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data points in the layer.
color	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
name	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Return Value

A PolarLineLayer object representing the chart layer created.

See also:

- 3.40.9 addLineLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarLineLayerMBS 573
- 3.40.10 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarLineLayerMBS 574

- 3.40.11 `addLineLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 575
- 3.40.13 `addLineLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 576
- 3.40.14 `addLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 576

3.40.13 `addLineLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a polar line layer to the polar chart.

Notes: Variation taking `dateTime` array and `color`.

See also:

- 3.40.9 `addLineLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 573
- 3.40.10 `addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 574
- 3.40.11 `addLineLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 575
- 3.40.12 `addLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 575
- 3.40.14 `addLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 576

3.40.14 `addLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarLineLayerMBS`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a polar line layer to the polar chart.

Notes: Variation taking `dateTime` array.

See also:

- 3.40.9 `addLineLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarLineLayerMBS` 573
- 3.40.10 `addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarLineLayerMBS` 574

3.40. CLASS CDPOLARCHARTMBS 577

- 3.40.11 addLineLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarLineLayerMBS 575
- 3.40.12 addLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarLineLayerMBS 575
- 3.40.13 addLineLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarLineLayerMBS 576

3.40.15 addSplineAreaLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarSplineAreaLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addSplineAreaLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.16 addSplineAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarSplineAreaLayerMBS 577
- 3.40.17 addSplineAreaLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarSplineAreaLayerMBS 578
- 3.40.18 addSplineAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarSplineAreaLayerMBS 578
- 3.40.19 addSplineAreaLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarSplineAreaLayerMBS 579
- 3.40.20 addSplineAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarSplineAreaLayerMBS 580

3.40.16 addSplineAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarSplineAreaLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar spline area layer to the polar chart.

Notes:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data points in the layer.
color	-1	The color to draw the spline area. -1 means that the color is automatically selected from the palette.
name	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Return Value

A `PolarSplineAreaLayer` object representing the chart layer created.

See also:

- 3.40.15 `addSplineAreaLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 577
- 3.40.17 `addSplineAreaLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 578
- 3.40.18 `addSplineAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 578
- 3.40.19 `addSplineAreaLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 579
- 3.40.20 `addSplineAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 580

3.40.17 `addSplineAreaLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Same as the other `addSplineAreaLayer` method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.15 `addSplineAreaLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 577
- 3.40.16 `addSplineAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 577
- 3.40.18 `addSplineAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 578
- 3.40.19 `addSplineAreaLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 579
- 3.40.20 `addSplineAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 580

3.40.18 `addSplineAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS`

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data points in the layer.
color	-1	The color to draw the spline area. -1 means that the color is automatically selected from the palette.
name	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Function: Adds a polar spline area layer to the polar chart.

Notes:

Return Value

A `PolarSplineAreaLayer` object representing the chart layer created.

See also:

- 3.40.15 `addSplineAreaLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 577
- 3.40.16 `addSplineAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 577
- 3.40.17 `addSplineAreaLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 578
- 3.40.19 `addSplineAreaLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 579
- 3.40.20 `addSplineAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 580

3.40.19 `addSplineAreaLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS`

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar spline area layer to the polar chart.

See also:

- 3.40.15 `addSplineAreaLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 577
- 3.40.16 `addSplineAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 577
- 3.40.17 `addSplineAreaLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 578
- 3.40.18 `addSplineAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 578

- 3.40.20 `addSplineAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 580

3.40.20 `addSplineAreaLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS`

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar spline area layer to the polar chart.

See also:

- 3.40.15 `addSplineAreaLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 577
- 3.40.16 `addSplineAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 577
- 3.40.17 `addSplineAreaLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 578
- 3.40.18 `addSplineAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineAreaLayerMBS` 578
- 3.40.19 `addSplineAreaLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineAreaLayerMBS` 579

3.40.21 `addSplineLineLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addSplineLineLayer` method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.22 `addSplineLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 581
- 3.40.23 `addSplineLineLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 581
- 3.40.24 `addSplineLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 582
- 3.40.25 `addSplineLineLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 582
- 3.40.26 `addSplineLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 583

3.40.22 addSplineLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarSplineLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar spline line layer to the polar chart.

Notes:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data points in the layer.
color	-1	The color to draw the spline line. -1 means that the color is automatically selected from the palette.
name	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Return Value

A PolarSplineLineLayer object representing the chart layer created.

See also:

- 3.40.21 addSplineLineLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarSplineLineLayerMBS 580
- 3.40.23 addSplineLineLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarSplineLineLayerMBS 581
- 3.40.24 addSplineLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "") as CDPolarSplineLineLayerMBS 582
- 3.40.25 addSplineLineLayer(dates() as dateTime, colorvalue as color, name as string = "") as CDPolarSplineLineLayerMBS 582
- 3.40.26 addSplineLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "") as CDPolarSplineLineLayerMBS 583

3.40.23 addSplineLineLayer(dates() as date, colorvalue as color, name as string = "") as CDPolarSplineLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Same as the other addSplineLineLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.21 addSplineLineLayer(data() as Double, colorvalue as color, name as string = "") as CDPolarSplineLineLayerMBS 580
- 3.40.22 addSplineLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDPolarSplineLineLayerMBS 581

- 3.40.24 `addSplineLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 582
- 3.40.25 `addSplineLineLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 582
- 3.40.26 `addSplineLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 583

3.40.24 `addSplineLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS`

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a polar spline line layer to the polar chart.

Notes:

Argument	Default	Description
<code>data</code>	(Mandatory)	An array of numbers representing the data points in the layer.
<code>color</code>	-1	The color to draw the spline line. -1 means that the color is automatically selected from the palette.
<code>name</code>	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Return Value

A `PolarSplineLineLayer` object representing the chart layer created.

See also:

- 3.40.21 `addSplineLineLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 580
- 3.40.22 `addSplineLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 581
- 3.40.23 `addSplineLineLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 581
- 3.40.25 `addSplineLineLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 582
- 3.40.26 `addSplineLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 583

3.40.25 `addSplineLineLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS`

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar spline line layer to the polar chart.

See also:

- 3.40.21 `addSplineLineLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 580
- 3.40.22 `addSplineLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 581
- 3.40.23 `addSplineLineLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 581
- 3.40.24 `addSplineLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 582
- 3.40.26 `addSplineLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 583

3.40.26 `addSplineLineLayer(dates() as dateTime, colorvalue as integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS`

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a polar spline line layer to the polar chart.

See also:

- 3.40.21 `addSplineLineLayer(data() as Double, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 580
- 3.40.22 `addSplineLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 581
- 3.40.23 `addSplineLineLayer(dates() as date, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 581
- 3.40.24 `addSplineLineLayer(dates() as date, colorvalue as Integer = -1, name as string = "")` as `CDPolarSplineLineLayerMBS` 582
- 3.40.25 `addSplineLineLayer(dates() as dateTime, colorvalue as color, name as string = "")` as `CDPolarSplineLineLayerMBS` 582

3.40.27 `addVectorLayer(rdata() as Double, adata() as Double, lengths() as Double, directions() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDPolarVectorLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a vector layer to the chart.

Notes: The vectors are specified as 4 data series, representing the radial and angular coordinates of the

reference points to put the vectors, and the lengths and directions of the vectors.

By default, the vector starts from the reference point and points away from it. You may use `PolarVectorLayer.setArrowAlignment` to specify other options, such as for the vectors to point into the reference point, or to have the reference as a pivot at the mid-point of the vector.

`ChartDirector` supports specifying vectors lengths as pixels or in axis scale. The unit is specified by using the following predefined constants.

ConstantValueDescription

<code>PixelScale</code>	0	The unit is measured in pixels.
<code>RadialAxisScale</code>	2	The unit is measured in the radial axis scale.

Argument	Default	Description
<code>rData</code>	(Mandatory)	An array of numbers representing the radial coordinates for the reference points of the vectors.
<code>aData</code>	(Mandatory)	An array of numbers representing the angular coordinates for the reference points of the vectors.
<code>lengths</code>	(Mandatory)	An array of numbers representing the lengths of the vectors, in unit as specified in the <code>lengthScale</code> argument.
<code>directions</code>	(Mandatory)	An array of numbers representing the direction of the vectors as a clockwise angle in degrees, where 0 is upward pointing direction.
<code>lengthScale</code>	<code>PixelScale</code>	The unit for the lengths, which must be one of the predefined constants in the table above.
<code>color</code>	-1	The color to draw the data points. -1 means that the color is automatically selected from the palette.
<code>name</code>	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Return Value

A `PolarVectorLayer` object representing the vector layer created.

See also:

- 3.40.28 `addVectorLayer(rdata() as Double, adata() as Double, lengths() as Double, directions() as Double, lengthScale as Integer, colorvalue as color, name as string = "")` as `CDPolarVectorLayerMBS`

3.40.28 `addVectorLayer(rdata() as Double, adata() as Double, lengths() as Double, directions() as Double, lengthScale as Integer, colorvalue as color, name as string = "") as CDPolarVectorLayerMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addVectorLayer` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.40.27 `addVectorLayer(rdata() as Double, adata() as Double, lengths() as Double, directions() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "") as CDPolarVectorLayerMBS` 583

3.40.29 `Constructor(width as Integer = 640, height as Integer = 480, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `Constructor` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.40.30 `Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)` 585

3.40.30 `Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new `PolarChart` object.

Notes:

Argument	Default	Description
<code>width</code>	(Mandatory)	The width of the chart in pixels.
<code>height</code>	(Mandatory)	The height of the chart in pixels.
<code>bgColor</code>	<code>BackgroundColor</code>	The background color of the chart.
<code>edgeColor</code>	<code>Transparent</code>	The edge color of the chart.
<code>raisedEffect</code>	<code>0</code>	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

See also:

- 3.40.29 Constructor(width as Integer = 640, height as Integer = 480, bgColor as color, edgeColor as color, raisedEffect as Integer = 0) 585

3.40.31 getXCoord(r as Double, a as Double) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x pixel coordinate of a point given its radial and angular coordinates.

Notes: The radial and angular coordinates are measured using the scale of the radial and angular axes. In particular, the scale of the angular axis may not be in degrees or radians. See `AngularAxis.setLabels` and `AngularAxis.setLinearScale` on how the angular axis scale is defined.

Note: You must call `BaseChart.layout` first before calling this method. It is because `ChartDirector` needs to perform auto-scaling and determine the axis scale first before it can compute the coordinates.

Argument	Default	Description
r	(Mandatory)	The radial coordinate of the point.
a	(Mandatory)	The angular coordinate of the point.

Return Value

The x pixel coordinate of the point.

3.40.32 getYCoord(r as Double, a as Double) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y pixel coordinate of a point given its radial and angular coordinates.

Notes: The radial and angular coordinates are measured using the scale of the radial and angular axes. In particular, the scale of the angular axis may not be in degrees or radians. See `AngularAxis.setLabels` and `AngularAxis.setLinearScale` on how the angular axis scale is defined.

Note: You must call `BaseChart.layout` first before calling this method. It is because `ChartDirector` needs to perform auto-scaling and determine the axis scale first before it can compute the coordinates.

Argument	Default	Description
r	(Mandatory)	The radial coordinate of the point.
a	(Mandatory)	The angular coordinate of the point.

Return Value

The y pixel coordinate of the point.

3.40.33 setGridColor(rGridColor as color, rGridWidth as Integer, aGridColor as color, aGridWidth as Integer = 1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setGridColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.34 setGridColor(rGridColor as Integer = &h80000000, rGridWidth as Integer = 1, aGridColor as Integer = &h80000000, aGridWidth as Integer = 1) 587

3.40.34 setGridColor(rGridColor as Integer = &h80000000, rGridWidth as Integer = 1, aGridColor as Integer = &h80000000, aGridWidth as Integer = 1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the grid colors and widths of the polar plot area.

Notes:

Argument	Default	Description
rGridColor	80000000	The color of grid lines in the radial direction (from the center outwards).
rGridWidth	1	The line width for grid lines in the radial direction (from the center outwards).
aGridColor	80000000	The color of grid lines in the angular direction (concentric circles around the center).
aGridWidth	1	The line width of grid lines in the angular direction (concentric circles around the center) The line width is only used if the polar plot area is using a polygon grid (see PolarChart.setGridStyle). If circular grid is used, the line width is always 1.

See also:

- 3.40.33 setGridColor(rGridColor as color, rGridWidth as Integer, aGridColor as color, aGridWidth as Integer = 1) 587

3.40.35 setGridStyle(polygonGrid as boolean, gridOnTop as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configure whether to use circular grids or polygon grids, and whether the grid lines are on top of the polar plot area or are at the back.

Notes:

Argument	Default	Description
polygonGrid	(Mandatory)	A true value means polygonal grid will be used. A false value means circular grid will be used.
gridOnTop	true	A true value means the grid lines will be on top of the polar plot area. A false value means the grid lines will be at the bottom of the polar plot area.

3.40.36 setPlotArea(x as Integer, y as Integer, r as Integer, bgColor as color, edgeColor as color, edgeWidth as Integer = 1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setPlotArea method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.37 setPlotArea(x as Integer, y as Integer, r as Integer, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000, edgeWidth as Integer = 1) 588

3.40.37 setPlotArea(x as Integer, y as Integer, r as Integer, bgColor as Integer = &hff000000, edgeColor as Integer = &hff000000, edgeWidth as Integer = 1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position, background colors, border color and border width of the polar plot area.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of the center of the polar plot area.
y	(Mandatory)	The y coordinate of the center of the polar plot area.
r	(Mandatory)	The radius of the polar plot area in pixels.
bgColor	Transparent	The background color of the polar plot area.
edgeColor	Transparent	The border color of the polar plot area.
edgeWidth	1	The border width of the polar plot area. The border width is only used if the polar plot area is using a polygon grid (see PolarChart.setGridStyle). If circular grid is used, the border width is always 1.

See also:

- 3.40.36 setPlotArea(x as Integer, y as Integer, r as Integer, bgColor as color, edgeColor as color, edgeWidth as Integer = 1) 588

3.40.38 setPlotAreaBg(bgColor1 as color, bgColor2 as color, altRings as boolean = true)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setPlotAreaBg method, but uses color instead of integer data type for passing color values.

See also:

- 3.40.39 setPlotAreaBg(bgColor1 as Integer, bgColor2 as Integer, altRings as boolean = true) 589

3.40.39 setPlotAreaBg(bgColor1 as Integer, bgColor2 as Integer, altRings as boolean = true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets alternating background colors for the polar plot area.

Notes: ChartDirector supports using two alternating background colors for the polar plot area. The colors can change on the radial direction, resulting in concentric circles, on an the angular direction, resulting in alternating sectors.

Argument	Default	Description
bgColor1	(Mandatory)	The first background color.
bgColor2	-1	The second background color. The default value of -1 means it is the same as the first background color.
altRings	true	A true value means the background colors alternate in the radial direction, resulting in concentric circles. A false value means the background colors alternates in the angular direction, resulting in alternating sectors.

See also:

- 3.40.38 setPlotAreaBg(bgColor1 as color, bgColor2 as color, altRings as boolean = true) 589

3.40.40 setStartAngle(startAngle as Double, clockwise as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the angle of the radial axis, and the layout direction of radial grid lines.

Notes: The default radial axis angle is 0 degree, which means the radial axis is upward pointing.

Argument	Default	Description
startAngle	(Mandatory)	The angle of the radial axis in degrees. The upward pointing direction is 0 degree, with angle measured in the clockwise direction.
clockwise	true	A true value means the radial grid line will be laid out in the clockwise direction. A false value means the radial grid lines will be laid out in the counter-clockwise direction.

3.40.41 Properties

3.40.42 angularAxis as CDAngularAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the AngularAxis object representing the angular axis of the polar chart.

Notes: (Read only property)

3.40.43 radialAxis as CDRadialAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the RadialAxis object representing the radial axis of the polar chart.

Notes: (Read only property)

3.41 class CDPolarLayerMBS

3.41.1 class CDPolarLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PolarLayer class is the base class for all PolarChart layer classes.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Xojo Plugins, version 18.5pr7](#)
- [MBS Real Studio Plugins, version 12.4pr3](#)
- [MBS Real Studio Plugins, version 12.1pr1](#)

3.41.2 Methods

3.41.3 addCustomDataLabel(i as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a custom data label to a data point.

Notes: See Font Specification for details on various font attributes.

Argument	Default	Description
i	(Mandatory)	The data point number for the data point. The first data point is 0, while the nth data point is (n - 1).
label	(Mandatory)	A text string representing the data label. Parameter Substitution and Formatting is supported.
font	""	The font used to draw the label.
fontSize	8	The font size used to draw the label.
fontColor	TextColor	The color used to draw the label.
fontAngle	0	The rotation angle of the label.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.41.4 addCustomDataLabel(i as Integer, label as string, font as string, fontSize as Double, fontColor as color, fontAngle as Double = 0) as CDTextBoxMBS 592

3.41.4 addCustomDataLabel(i as Integer, label as string, font as string, font-Size as Double, fontColor as color, fontAngle as Double = 0) as CD-TextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addCustomDataLabel method, but uses color instead of integer data type for passing color values.

See also:

- 3.41.3 addCustomDataLabel(i as Integer, label as string, font as string = "", fontSize as Double = 8, fontColor as Integer = &hfff0002, fontAngle as Double = 0) as CDTextBoxMBS 591

3.41.5 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.41.6 getHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "", offsetX as Integer = 0, offsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Generates an HTML image map for all data points on the layer.

Notes: This method should be called only after creating the chart image (eg. using BaseChart.makeChart, BaseChart.makeChart2 or BaseChart.makeChart3). The image map cannot be determined without creating the chart image first.

Argument	Default	Description
url	(Mandatory)	The URL to be used in the "href" attribute of the image map. Parameter Substitution and Formatting is supported. Use an empty string if no href attribute is needed.
queryFormat	""	A text string representing the template of the query parameters to be appended to the URL. Parameter Substitution and Formatting is supported.

The special keyword " { default } " represents the default query parameters. This is useful for specifying appending to the default.

Note that an empty string means to use the default query query parameters. To specify no query parameter, use a space character.

extraAttr	""	A text string to specify additional attributes to add to the <area> tag. Parameter Substitution and Formatting is supported.
offsetX	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
offsetY	0	An offset to be added to all y coordinates in the image map. See offsetX above for description.

Return Value

A text string containing the image map generated.

3.41.7 getImageCoor(dataItem as Integer, offsetX as Integer = 0, offsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the image map coordinates of a data point.

Notes: The image map coordinates will be in the following format:

```
shape="rect" cords=" [ x1 ] , [ y1 ] , [ x2 ] , [ y2 ] "
```

This format is specially designed so that it can easily be included into HTML image maps.

This method should be called only after creating the chart image (eg. using BaseChart.makeChart, BaseChart.makeChart2 or BaseChart.makeChart3). The image map cannot be determined without creating the chart image first.

Argument	Default	Description
dataItem	(Mandatory)	The data point number for the data point. The first data point is 0, while the nth data point is (n - 1).
offsetX	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
offsetY	0	An offset to be added to all y coordinates in the image map. See offsetX above for description.

Return Value

A text string representing the image map coordinates of the data points as HTML image map attributes.

3.41.8 setAngles(data() as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the angular coordinates of the data points.

Notes: Note that the angular coordinates are measured using the scale of the angular axis, which may not be in degrees or radians. See `AngularAxis.setLabels` and `AngularAxis.setLinearScale` on how the angular axis scale is defined.

If this method is not called, the first data point is assumed to have an angular coordinate of 0, and the n th data point is assumed to have an angular coordinate of $(n - 1)$. This is common for radar charts, in which enumerated scale is used for the angular axis (see `AngularAxis.setLabels`).

Argument	Default	Description
angles	(Mandatory)	An array of numbers representing the angular coordinates of the data points.

3.41.9 setBorderColor(edgeColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setBorderColor` method, but uses color instead of integer data type for passing color values.

See also:

- 3.41.10 `setBorderColor(edgeColor as Integer)` 594

3.41.10 setBorderColor(edgeColor as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the border color for drawing the data on the layer.

Notes: This method only applies to layers that represents data with elements that have borders (e.g. polar area layer and polar spline area layer).

Argument	Default	Description
edgeColor	(Mandatory)	The border color.

See also:

- 3.41.9 `setBorderColor(edgeColor as color)` 594

3.41.11 setData(data() as Double, colorvalue as color, name as string = "")

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setData method, but uses color instead of integer data type for passing color values.

See also:

- 3.41.12 setData(data() as Double, colorvalue as Integer = -1, name as string = "") 595

3.41.12 setData(data() as Double, colorvalue as Integer = -1, name as string = "")

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the values of the data points.

See also:

- 3.41.11 setData(data() as Double, colorvalue as color, name as string = "") 595

3.41.13 setDataLabelFormat(formatString as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data label format.

Notes: By default, the data label format is " { value } ". Please refer to Parameter Substitution and Formatting on available parameters and how to format them.

Argument	Default	Description
formatString	(Mandatory)	The format string.

3.41.14 setDataLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = 0, fontangle as Double = 0) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables data labels and sets their styles.

Notes:

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See also:

Argument	Default	Description
font	""	The font used to draw the labels.
fontSize	8	The font size used to draw the labels.
fontColor	TextColor	The color used to draw the labels.
fontAngle	0	The rotation angle of the labels.

- 3.41.15 setDataLabelStyle(font as string, fontsize as Double, fontcolor as color, fontangle as Double = 0) as CDTextBoxMBS 596

3.41.15 setDataLabelStyle(font as string, fontsize as Double, fontcolor as color, fontangle as Double = 0) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setDataLabelStyle method, but uses color instead of integer data type for passing color values.

See also:

- 3.41.14 setDataLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = 0, fontangle as Double = 0) as CDTextBoxMBS 595

3.41.16 setDataSymbol(area as CDDrawAreaMBS)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a DrawArea object as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
obj	(Mandatory)	A DrawArea object to be used as the symbol.

See also:

- 3.41.17 setDataSymbol(image as folderitem) 597
- 3.41.18 setDataSymbol(pic as Picture) 597
- 3.41.19 setDataSymbol(polygon() as Integer, size as Integer = 11, fillcolor as Integer = -1, edgecolor as Integer = -1) 598
- 3.41.20 setDataSymbol(polygon() as Integer, size as Integer, fillcolor as color, edgecolor as Integer = -1) 599
- 3.41.21 setDataSymbol(symbol as Integer, size as Integer = 7, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 599

- 3.41.22 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 600

3.41.17 setDataSymbol(image as folderitem)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load an image from a file and use it as the graphics symbol to plot the data points.

Notes: ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to BaseChart.setSearchPath on the directory that ChartDirector will search for the file.

Argument	Default	Description
image	(Mandatory)	The filename of the image file. The image type is determined based on file extension, which must be png, jpg/jpeg, gif or wbmp/wmp.

See also:

- 3.41.16 setDataSymbol(area as CDDrawAreaMBS) 596
- 3.41.18 setDataSymbol(pic as Picture) 597
- 3.41.19 setDataSymbol(polygon() as Integer, size as Integer = 11, fillcolor as Integer = -1, edgcolor as Integer = -1) 598
- 3.41.20 setDataSymbol(polygon() as Integer, size as Integer, fillcolor as color, edgcolor as Integer = -1) 599
- 3.41.21 setDataSymbol(symbol as Integer, size as Integer = 7, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 599
- 3.41.22 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 600

3.41.18 setDataSymbol(pic as Picture)

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a picture object as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
obj	(Mandatory)	A picture object to be used as the symbol.

See also:

- 3.41.16 setDataSymbol(area as CDDrawAreaMBS) 596
- 3.41.17 setDataSymbol(image as folderitem) 597
- 3.41.19 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgecolor as Integer = -1) 598
- 3.41.20 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgecolor as Integer = -1) 599
- 3.41.21 setDataSymbol(symbol as Integer, size as Integer = 7, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 599
- 3.41.22 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 600

3.41.19 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgecolor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a custom polygon as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
polygon	(Mandatory)	An array of integers representing the coordinates the polygon vertices. See Shape Specification on how the custom shape is defined.
size	11	The nominal width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means the color of the data set will be used.
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means the edge color of the data set will be used.

See also:

- 3.41.16 setDataSymbol(area as CDDrawAreaMBS) 596
- 3.41.17 setDataSymbol(image as folderitem) 597
- 3.41.18 setDataSymbol(pic as Picture) 597
- 3.41.20 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgecolor as Integer = -1) 599
- 3.41.21 setDataSymbol(symbol as Integer, size as Integer = 7, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 599
- 3.41.22 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 600

3.41.20 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgcolor as Integer = -1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setDataSymbol method, but uses color instead of integer data type for passing color values.

See also:

- 3.41.16 setDataSymbol(area as CDDrawAreaMBS) 596
- 3.41.17 setDataSymbol(image as folderitem) 597
- 3.41.18 setDataSymbol(pic as Picture) 597
- 3.41.19 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgcolor as Integer = -1) 598
- 3.41.21 setDataSymbol(symbol as Integer, size as Integer = 7, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 599
- 3.41.22 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 600

3.41.21 setDataSymbol(symbol as Integer, size as Integer = 7, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses one of the built-in symbols as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
symbol	(Mandatory)	One of the predefined shape constants representing the symbol shape. See Shape Specification for the available built-in shapes.
size	7	The width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means the color of the data set will be used.
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means the edge color of the data set will be used.
lineWidth	1	The line width used for drawing the symbols.

See also:

- 3.41.16 setDataSymbol(area as CDDrawAreaMBS) 596
- 3.41.17 setDataSymbol(image as folderitem) 597
- 3.41.18 setDataSymbol(pic as Picture) 597

- 3.41.19 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgecolor as Integer = -1) 598
- 3.41.20 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgecolor as Integer = -1) 599
- 3.41.22 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 600

3.41.22 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setDataSymbol method, but uses color instead of integer data type for passing color values.

See also:

- 3.41.16 setDataSymbol(area as CDDrawAreaMBS) 596
- 3.41.17 setDataSymbol(image as folderitem) 597
- 3.41.18 setDataSymbol(pic as Picture) 597
- 3.41.19 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgecolor as Integer = -1) 598
- 3.41.20 setDataSymbol(polygon() as Integer, size as Integer, fillColor as color, edgecolor as Integer = -1) 599
- 3.41.21 setDataSymbol(symbol as Integer, size as Integer = 7, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 599

3.41.23 setHTMLImageMap(url as string, queryFormat as string = "", extraAttr as string = "")

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Override the default arguments used when generating HTML image map for the layer.

Notes: BaseChart.getHTMLImageMap can be used to generate HTML image map for the whole chart. When BaseChart.getHTMLImageMap is used, the image map for all layers will be generated based on the arguments supplied to BaseChart.getHTMLImageMap.

The setHTMLImageMap method can be used to override those arguments for a chart layer, so the image map for that layer can be different.

For a detail description of image maps, please refer to BaseChart.getHTMLImageMap.

Argument	Default	Description
url	(Mandatory)	The URL to be used in the "href" attribute of the image map. Parameter Substitution and Formatting is supported.

The special keyword " { default } " represents the global URL as specified in BaseChart.getHTMLImageMap. This field is useful for specifying appending to the global URL.

Note that an empty string also means to use the global URL. To specify no URL, use the special keyword " { none } ".

To disable the entire image map, use the special keyword " { disable } ".

queryFormat	""	A text string representing the template of the query parameters to be appended to the URL. Parameter Substitution and Formatting is supported.
-------------	----	--

The special keyword " { default } " represents the global query parameters as specified in BaseChart.getHTMLImageMap. This field is useful for specifying appending to the global query parameters.

Note that an empty string also means to use the global query parameters. To specify no query parameters, use the special keyword " { none } ".

extraAttr	""	A text string to specify additional attributes to add to the <area> tag. Parameter Substitution and Formatting is supported.
-----------	----	--

The special keyword " { default } " represents the global additional attributes as specified in BaseChart.getHTMLImageMap. This field is useful for specifying appending to the global additional attributes.

Note that an empty string also means to use the global additional attributes. To specify no additional attributes, use the special keyword " { none } ".

3.41.24 setImageMapWidth(width as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the effective size of a data point for producing image maps.

Notes: For the purpose of producing image maps for the data points, the sizes of the data points are assumed to be the size of the data symbols. If no data symbol is used, an effective size is assumed. The

default is 10 pixels in width and height.

Argument	Default	Description
width	(Mandatory)	The effective width and height of the data point for the purpose of producing image maps.

3.41.25 `setLineWidth(w as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the line width of lines when drawing the data on the layer.

Notes:

Argument	Default	Description
w	(Mandatory)	The width of the line in pixels.

3.41.26 `setSymbolOffset(offsetX as Integer, offsetY as Integer)`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Offset the symbols in the x and y directions in pixel unit.

Notes:

Argument	Default	Description
xOffset	(Mandatory)	The x offset in pixels. A positive value mean shifting to the right.
yOffset	(Mandatory)	The y offset in pixels. A positive value mean shifting to the bottom.

3.41.27 `setSymbolScale(data() as Double, scaleType as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the symbol for each data point (for creating bubble charts).

Notes: One common usage for this method is to draw circle symbols of different sizes at each data points, creating a bubble chart.

This method supports any valid data symbols. You can create bubble charts with square bubbles, or even custom data symbols.

ChartDirector supports specifying sizes as pixels or in axis scale. The unit is specified by using the following predefined constants.

ConstantValueDescription

PixelScale	0	The unit is measured in pixels.
RadialAxisScale	2	The unit is measured in the radial axis scale.

Argument	Default	Description
zData	(Mandatory)	The sizes of the symbols, expressed using the unit defined by the scaleType argument.
scaleType	PixelScale	The unit for zData, which must be one of the predefined constants in the table above.

3.42 class CDPolarLineLayerMBS

3.42.1 class CDPolarLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PolarLineLayer class represents polar area layers.

Notes: Subclass of the CDPolarLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Xojo Developer Magazine

- [7.5, page 32: Easy Charts and Graphs Part 2, Using the ChartDirector Plugin](#)

3.42.2 Methods

3.42.3 setCloseLoop(b as boolean)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Specifies whether the polar line should form a close loop (joining the last point to the first point) or not.

Notes:

Argument	Default	Description
b	(Mandatory)	A true value means the polar line should form a close loop. A false value means the polar line should not form a close loop.

3.42.4 setGapColor(lineColor as color, lineWidth as Integer)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setGapColor method, but uses color instead of integer data type for passing color values.

See also:

- [3.42.5 setGapColor\(lineColor as Integer, lineWidth as Integer\)](#)

604

3.42.5 setGapColor(lineColor as Integer, lineWidth as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color and style of the line used for jumping across NoValue data points.

Notes: By default, the color of the line for jumping across NoValue data points is Transparent, which means

the line will become discontinuous

Argument	Default	Description
lineColor	(Mandatory)	The line color of the line used for jumping across NoValue data points
lineWidth	-1	The line width of the line used for jumping across NoValue data points. -1 means the width will be the same as the line width of the line for drawing normal data points.

See also:

- 3.42.4 setGapColor(lineColor as color, lineWidth as Integer)

3.43 class CDPolarSplineAreaLayerMBS

3.43.1 class CDPolarSplineAreaLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PolarSplineAreaLayer class represents polar spline area layers.

Notes: Subclass of the CDPolarAreaLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.43.2 Methods

3.43.3 setTension(tension as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the tension to use when computing the spline curve.

Notes: The tension parameter should be between -1 and 1. A positive tension will make the spline tighter. The spline curve will become straight line segments when tension is 1. A negative tension will make the spline looser.

Argument	Default	Description
tension	(Mandatory)	The tension of the spline, which should be between -1 and 1.

3.44 class CDPolarSplineLineLayerMBS

3.44.1 class CDPolarSplineLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PolarSplineLineLayer class represents polar spline line layers.

Notes: Subclass of the CDPolarLineLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.44.2 Methods

3.44.3 setTension(tension as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the tension to use when computing the spline curve.

Notes: The tension parameter should be between -1 and 1. A positive tension will make the spline tighter. The spline curve will become straight line segments when tension is 1. A negative tension will make the spline looser.

Argument	Default	Description
tension	(Mandatory)	The tension of the spline, which should be between -1 and 1.

3.45 class CDPolarVectorLayerMBS

3.45.1 class CDPolarVectorLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PolarVectorLayer class represents polar vector layers.

Notes: Subclass of the CDPolarLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.45.2 Methods

3.45.3 setArrowAlignment(alignment as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the alignment of the vector relative to the data point.

Notes:

Argument	Default	Description
alignment	(Mandatory)	A BottomCenter value means the vector will point away from the data point (the default). A TopCenter value means the vector will point into the data point. A Center value means the center of the vector will be at the data point.

3.45.4 setArrowHead(polygon() as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets a custom shape to be used as the arrow head.

Notes: The custom shape is specified as an array of integers x0, y0, x1, y1, x2, y2 ... representing the coordinates of the vertices of the custom polygonal shape.

The polygon should be defined with a bounding square of 10 x 10 units, in which the x-axis is from left to right, and the y-axis from bottom to top. The origin is assumed to be the bottom center of the arrow (the point where the arrow head joins the arrow stem). The shape is assumed to represent an arrow pointing upwards.

As an example, the followings are the integer array that represents the standard ChartDirector vector arrow head:

-5, -5, 0, 0, 5, -5, 0, 5

ChartDirector will automatically scale the shape to the actual width and height as specified in `PolarVectorLayer.setArrowHead`.

Argument	Default	Description
<code>polygon</code>	(Mandatory)	An array of integers <code>x0, y0, x1, y1, x2, y2 ...</code> representing the coordinates the polygon vertices on a 10 x 10 units grid.

See also:

- 3.45.5 `setArrowHead(width as Integer, height as Integer)` 609

3.45.5 `setArrowHead(width as Integer, height as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the arrow head.

Notes:

Argument	Default	Description
<code>width</code>	(Mandatory)	The width of the arrow head in pixels. The default width is 8 pixels.
<code>height</code>	0	The height of the arrow head in pixels. The default value of 0 means the height is the same as the width.

See also:

- 3.45.4 `setArrowHead(polygon() as Integer)` 608

3.45.6 `setArrowStem(polygon() as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets a custom shape to be used as the arrow stem.

Notes: By default, the arrow stem is just a straight line, with the line width controlled using `PolarLayer.setLineWidth`. The `setArrowStem` method can specify a custom shape for the arrow stem.

The custom shape is specified as an array of integers `x0, y0, x1, y1, x2, y2 ...` representing the coordinates of the vertices of the custom polygonal shape.

The polygon should be defined with a bounding square of 10 x 100 units, in which the x-axis is from left to right, and the y-axis from bottom to top. The origin is assumed to be the starting point of the arrow stem, and the shape is assumed to represent an arrow stem pointing upwards.

ChartDirector will automatically scale the shape so that the total arrow length (head + stem) is the required length of the arrow as according to actual data, and the stem width is as specified in `PolarLayer.setLineWidth`.

Argument	Default	Description
<code>polygon</code>	(Mandatory)	An array of integers <code>x0, y0, x1, y1, x2, y2 ...</code> representing the coordinates the polygon vertices on a 10 x 100 units grid.

3.45.7 `setIconSize(height as Integer, width as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the icon to be used in legend box.

Notes: By default, if a legend box is available on the chart, ChartDirector will insert an legend entry if the `PolarVectorLayer` is named. The size of the icon will be the size of the vectors used on the chart, using a short vector length to fit the legend box.

This method can be used to override the legend box settings to specify a custom width/height for the icons of the current `PolarVectorLayer`.

Argument	Default	Description
<code>height</code>	(Mandatory)	The height of the icon in pixels.
<code>width</code>	0	The width of the icon in pixels. The default value of 0 means the width is automatically determined.

3.45.8 `setVector(lengths() as Double, directions() as Double, lengthScale as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the lengths and directions for the vectors.

Notes: ChartDirector supports specifying lengths as pixels or in axis scale. The unit is specified by using the following predefined constants.

ConstantValueDescription

<code>PixelScale</code>	0	The unit is measured in pixels.
<code>RadialAxisScale</code>	2	The unit is measured in the radial axis scale.

3.45. CLASS *CDPOLARVECTORLAYERMBS*

611

Argument	Default	Description
lengths	(Mandatory)	An array of numbers representing the lengths of the vectors, in unit as specified in the lengthScale argument.
directions	(Mandatory)	An array of numbers representing the direction of the vectors as a clockwise angle in degrees, where 0 is upward pointing direction.
lengthScale	PixelScale	The unit for the lengths, which must be one of the predefined constants in the table above.

3.45.9 setVectorMargin(startMargin as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the vector margin.

See also:

- 3.45.10 setVectorMargin(startMargin as Double, endMargin as Double)

611

3.45.10 setVectorMargin(startMargin as Double, endMargin as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the vector margin.

See also:

- 3.45.9 setVectorMargin(startMargin as Double)

611

3.46 class CDPyramidChartMBS

3.46.1 class CDPyramidChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PyramidChart class represents pyramid charts (including cone charts and funnel charts).

Notes: The PyramidChart class is a subclass of BaseChart.

Subclass of the CDBaseChartMBS class.

Blog Entries

- [Create a cone chart with MBS and ChartDirector](#)
- [Chart Diagrams with Xojo](#)

Xojo Developer Magazine

- [7.4, page 31: Easy Charts and Graphs, Using the ChartDirector Plugin](#)

3.46.2 Methods

3.46.3 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new PyramidChart object.

Notes: Arguments:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	kBackgroundColor	The background color of the chart.
edgeColor	kTransparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

See also:

- [3.46.4 Constructor\(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0\)](#) 613

3.46.4 Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other Constructor method, but uses color instead of integer data type for passing color values.

See also:

- 3.46.3 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0) 612

3.46.5 getLayer(layerNo as Integer) as CDPyramidLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the PyramidLayer object representing a single pyramid layer in the pyramid chart.

Notes: You must call CDPyramidChartMBS.setData to create the layers first before calling this method.

Arguments

Argument	Default	Description
layerNo	(Mandatory)	The layer number of the pyramid layer to retrieve, starting from 0. The first pyramid layer is 0. The nth pyramid layer is (n-1).

Return Value

The requested CDPyramidLayerMBS object.

3.46.6 setCenterLabel(labelTemplate as string = "", font as string = "", font-Size as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds labels to the center of the pyramid layers.

Notes: This method affects all pyramid layers. To apply this method to one particular pyramid layer only, use CDPyramidLayerMBS.setCenterLabel.

See Parameter Substitution and Formatting on available format parameters for the template.

See Font Specification for details on various font attributes.

Arguments:

Argument	Default	Description
template	" { skip } "	The label template. " { skip } " means to keep the existing value unchanged.
font	" { skip } "	The font style. " { skip } " means to keep the existing value unchanged.
fontSize	-1	The font size in points. -1 means to keep the existing value unchanged.
fontColor	-1	The font color. -1 means to keep the existing value unchanged.

Return Value

A CDTextBoxMBS object representing the prototype of the labels. This may be used to fine-tune the appearance of the labels.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.46.7 `setCenterLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS` 614

3.46.7 `setCenterLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setCenterLabel` method, but uses color instead of integer data type for passing color values.

See also:

- 3.46.6 `setCenterLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS` 613

3.46.8 `setConeSize(cx as Integer, cy as Integer, radius as Integer, height as Integer)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position and size of the cone in a cone chart.

Notes: Arguments:

3.46.9 `setData(data() as Double)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Argument	Default	Description
cx	(Mandatory)	The x-coordinate of the center of the cone. The center of the cone is the midpoint of the vertical axis of the cone.
cy	(Mandatory)	The y-coordinate of the center of the cone. The center of the cone is the midpoint of the vertical axis of the cone.
radius	(Mandatory)	The radius of the base of the cone.
height	(Mandatory)	The height of the cone.

Function: Sets the data used to draw the pyramid chart.

Notes: Arguments:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data values.
labels	[Empty_Array]	An array of text strings representing the labels of the layers. An empty array means no layer label.

See also:

- 3.46.10 setData(data() as Double, labels() as string)

615

3.46.10 setData(data() as Double, labels() as string)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data used to draw the pyramid chart.

Notes: Arguments:

Argument	Default	Description
data	(Mandatory)	An array of numbers representing the data values.
labels	[Empty_Array]	An array of text strings representing the labels of the layers. An empty array means no layer label.

See also:

- 3.46.9 setData(data() as Double)

614

3.46.11 setFunnelSize(cx as Integer, cy as Integer, radius as Integer, height as Integer, tubeRadius as Double = 0.2, tubeHeight as Double = 0.3)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position and size of the funnel in a funnel chart.

Notes: A funnel is similar to an inverted cone, except that the vertex of the cone is replaced by a tube.

Arguments:

Argument	Default	Description
cx	(Mandatory)	The x-coordinate of the center of the funnel. The center of the funnel is the midpoint of the vertical axis of the funnel.
cy	(Mandatory)	The y-coordinate of the center of the funnel. The center of the funnel is the midpoint of the vertical axis of the funnel.
radius	(Mandatory)	The radius of the mouth of the funnel.
height	(Mandatory)	The height of the funnel, inclusive of the tube.
tubeRadius	0.2	The radius of the tube, as a ratio to the radius of the mouth of the funnel.
tubeHeight	0.3	The height (length) of the tube, as a ratio to the total height of the funnel. It must be between 0 to 1.

3.46.12 setGradientShading(startBrightness as Double, endBrightness as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets gradient shading mode.

Notes: Gradient shading only applies to 2D and 3D pyramids, and 2D cones and funnels. 3D cones and funnels always use Phong lighting (see CDPyramidChartMBS.setLighting).

The gradient is specified with two brightness values at the gradient end points. A brightness less than 1 means the color is darkened, while a brightness greater than 1 means the color is brightened. For example, a brightness of 0.5 means the color is half as bright as the original color. If the original color is red, the color will become dark red. Conversely, a brightness of 2 means the color is twice as bright as the original color. If the original color is red, the color will become light red.

For a pyramid or cone layer, the gradient is from left edge to the right edge. For a funnel chart, the gradient is from the right edge to the left edge.

For a pyramid chart, this is the default coloring method. If this method is never called, the default brightness is from 0.75 to 2.

Arguments:

Argument	Default	Description
startBrightness	(Mandatory)	The brightness at the starting point.
endBrightness	(Mandatory)	The brightness at the ending point.

3.46.13 setJoinLine(ColorValue as color, width as Integer = -1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setJoinLine method, but uses color instead of integer data type for passing color values.

See also:

- 3.46.14 setJoinLine(ColorValue as Integer, width as Integer = -1) 617

3.46.14 setJoinLine(ColorValue as Integer, width as Integer = -1)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color and line width of the join lines that join the pyramid layers with the left and right labels.

Notes: This method affects all pyramid layers. To apply this method to one particular pyramid layer only, use PyramidLayer.setJoinLine.

Arguments:

Argument	Default	Description
color	(Mandatory)	The color of the join line. -1 means to keep the existing value unchanged.
width	-1	The width of the line join. -1 means to keep the existing value unchanged.

See also:

- 3.46.13 setJoinLine(ColorValue as color, width as Integer = -1) 617

3.46.15 setJoinLineGap(pyramidGap as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gaps of the join lines that join the pyramid layers with the left and right labels.

Notes: This method affects all pyramid layers. To apply this method to one particular pyramid layer only, use CDPyramidLayerMBS.setJoinLineGap.

By default, the starting point of the join line connects to the pyramid layer edge with a 3 pixels gap. The join line is horizontal, and its length is such that the ending point is 10 pixels outside the pyramid bounding

box. The label text box connects to the ending point with a 3 pixels gap.

This method allows the gaps and positions of the join line end points to be configured to other values.

Arguments:

Argument	Default	Description
pyramidGap	(Mandatory)	The gap in pixels between the starting point of the join line and the pyramid layer edge.
pyramidMargin	10	The distance in pixels between the ending point of the join line and the pyramid bounding box.
textGap	3	The gap in pixels between the label text box and the ending point of the join line.

See also:

- 3.46.16 `setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer)` 618
- 3.46.17 `setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer, textGap as Integer)` 619

3.46.16 `setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gaps of the join lines that join the pyramid layers with the left and right labels.

Notes: This method affects all pyramid layers. To apply this method to one particular pyramid layer only, use `CDPyramidLayerMBS.setJoinLineGap`.

By default, the starting point of the join line connects to the pyramid layer edge with a 3 pixels gap. The join line is horizontal, and its length is such that the ending point is 10 pixels outside the pyramid bounding box. The label text box connects to the ending point with a 3 pixels gap.

This method allows the gaps and positions of the join line end points to be configured to other values.

Arguments:

See also:

- 3.46.15 `setJoinLineGap(pyramidGap as Integer)` 617
- 3.46.17 `setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer, textGap as Integer)` 619

Argument	Default	Description
pyramidGap	(Mandatory)	The gap in pixels between the starting point of the join line and the pyramid layer edge.
pyramidMargin	10	The distance in pixels between the ending point of the join line and the pyramid bounding box.
textGap	3	The gap in pixels between the label text box and the ending point of the join line.

3.46.17 setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer, textGap as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gaps of the join lines that join the pyramid layers with the left and right labels.

Notes: This method affects all pyramid layers. To apply this method to one particular pyramid layer only, use `CDPyramidLayerMBS.setJoinLineGap`.

By default, the starting point of the join line connects to the pyramid layer edge with a 3 pixels gap. The join line is horizontal, and its length is such that the ending point is 10 pixels outside the pyramid bounding box. The label text box connects to the ending point with a 3 pixels gap.

This method allows the gaps and positions of the join line end points to be configured to other values.

Arguments:

Argument	Default	Description
pyramidGap	(Mandatory)	The gap in pixels between the starting point of the join line and the pyramid layer edge.
pyramidMargin	10	The distance in pixels between the ending point of the join line and the pyramid bounding box.
textGap	3	The gap in pixels between the label text box and the ending point of the join line.

See also:

- 3.46.15 `setJoinLineGap(pyramidGap as Integer)` 617
- 3.46.16 `setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer)` 618

3.46.18 setLayerBorder(ColorValue as color, width as Integer = -1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setLayerBorder` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.46.19 `setLayerBorder(ColorValue as Integer, width as Integer = -1)` 620

3.46.19 `setLayerBorder(ColorValue as Integer, width as Integer = -1)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color and line width of the layer border.

Notes: This method affects all pyramid layers. To apply this method to one particular pyramid layer only, use `CDPyramidLayerMBS.setLayerBorder`.

Arguments:

Argument	Default	Description
<code>color</code>	(Mandatory)	The color of the layer border. -1 means to keep the existing value unchanged.
<code>width</code>	-1	The width of the layer border. -1 means to keep the existing value unchanged.

See also:

- 3.46.18 `setLayerBorder(ColorValue as color, width as Integer = -1)` 619

3.46.20 `setLayerGap(layerGap as Double)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gaps between pyramid layers.

Notes: This method affects all pyramid layers. To apply this method to one particular pyramid layer only, use `CDPyramidLayerMBS.setLayerGap`.

Arguments:

Argument	Default	Description
<code>layerGap</code>	(Mandatory)	The gap between layers as a ratio to the height of the pyramid. The gap must be greater than or equals 0. The sum of all gaps must be less than 1.

3.46.21 setLeftLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds labels to the left of the pyramid layers.

Notes: This method affects all pyramid layers. To apply this method to one particular pyramid layer only, use `CDPyramidLayerMBS.setLeftLabel`.

See Parameter Substitution and Formatting on available format parameters for the template.

See Font Specification for details on various font attributes.

Arguments:

Argument	Default	Description
template	" { skip } "	The label template. " { skip } " means to keep the existing value unchanged.
font	" { skip } "	The font style. " { skip } " means to keep the existing value unchanged.
fontSize	-1	The font size in points. -1 means to keep the existing value unchanged.
fontColor	-1	The font color. -1 means to keep the existing value unchanged.

Return Value

A `CDTextBoxMBS` object representing the prototype of the labels. This may be used to fine-tune the appearance of the labels.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.46.22 `setLeftLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS` 621

3.46.22 setLeftLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setLeftLabel` method, but uses color instead of integer data type for passing color values.

See also:

- 3.46.21 `setLeftLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS` 621

3.46.23 `setLighting(ambientIntensity as Double = 0.5, diffuseIntensity as Double = 0.5, specularIntensity as Double = 1, shininess as Double = 8)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets Phong lighting mode.

Notes: This method adjusts the brightness of the surfaces as according to the Phong lighting model, in which the light source is from the viewer direction and is far away.

For a cone or funnel chart, this is the default coloring method.

For a pyramid chart, only ambient and diffuse reflections are used.

Arguments:

Argument	Default	Description
<code>ambientIntensity</code>	0.5	The ambient reflection coefficient of the Phong lighting model.
<code>diffuseIntensity</code>	0.5	The diffuse reflection coefficient of the Phong lighting model.
<code>specularIntensity</code>	1	The specular reflection coefficient of the Phong lighting model.
<code>shininess</code>	8	The shininess coefficient of the Phong lighting model.

3.46.24 `setPyramidSides(noOfSides as Integer)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the number of sides of the base polygon in a pyramid chart.

Notes: By default, the base of a pyramid is a square. This method can be used to change it to other regular polygon.

Arguments:

Argument	Default	Description
<code>noOfSides</code>	(Mandatory)	The number of sides of the base polygon in a pyramid chart.

3.46.25 `setPyramidSize(cx as Integer, cy as Integer, radius as Integer, height as Integer)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position and size of the pyramid in a pyramid chart.

Notes: Arguments:

Argument	Default	Description
cx	(Mandatory)	The x-coordinate of the center of the pyramid. The center of the pyramid is the midpoint of the vertical axis of the pyramid.
cy	(Mandatory)	The y-coordinate of the center of the pyramid. The center of the pyramid is the midpoint of the vertical axis of the pyramid.
radius	(Mandatory)	For a square pyramid, it is the length of the square. For a pyramid with other regular polygonal base, it is the radius of the circumcircle of the base.
height	(Mandatory)	The height of the pyramid.

3.46.26 setRightLabel(labelTemplate as string = "", font as string = "", font-Size as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds labels to the right of the pyramid layers.

Notes: This method affects all pyramid layers. To apply this method to one particular pyramid layer only, use `CDPyramidLayerMBS.setRightLabel`.

See Parameter Substitution and Formatting on available format parameters for the template.

See Font Specification for details on various font attributes.

Arguments:

Argument	Default	Description
template	" { skip } "	The label template. " { skip } " means to keep the existing value unchanged.
font	" { skip } "	The font style. " { skip } " means to keep the existing value unchanged.
fontSize	-1	The font size in points. -1 means to keep the existing value unchanged.
fontColor	-1	The font color. -1 means to keep the existing value unchanged.

Return Value:

A `TextBox` object representing the prototype of the labels. This may be used to fine-tune the appearance of the labels.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.46.27 `setRightLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color)` as `CDTextBoxMBS` 624

3.46.27 `setRightLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color)` as `CDTextBoxMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setRightLabel` method, but uses color instead of integer data type for passing color values.

See also:

- 3.46.26 `setRightLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1)` as `CDTextBoxMBS` 623

3.46.28 `setViewAngle(elevation as Double, rotation as Double = 0, twist as Double = 0)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the 3D view angles.

Notes: ChartDirector uses elevation, rotation and twist angles to specify the 3D view angles.

To explain the meaning of the angles, imagine the object being viewed is put at the center of a hollow sphere, and a camera (the view point) is put at the surface of the sphere, directed inwards to the center to look at the object.

The elevation angle refers to the "latitude" of the camera. An elevation angle of 0 degrees means the camera is at the "equator" pointing to the object from the side. An elevation angle of 90 degrees means the camera is at the "north pole" pointing down to the object. An angle elevation of -90 degrees means the camera is at the "south pole", pointing up to the object.

The rotation angle refers to the "longitude" of the camera. If the elevation is 0 degrees, and the rotation angle varies from 0 to 360 degrees, the camera will move around the "equator" in the easterly direction (counter-clockwise when viewed from the north pole).

The twist angle is for rotating the camera itself while still pointing to the object. For example, a twist angle of 90 degrees means you are holder the camera "vertically" instead of "horizontally". The rotation is clockwise from the view point of the person holding the camera.

Note that from the view point of the camera, the object will appear to be rotating in the opposite direction. For example, as the rotation angle rotates the camera counter-clockwise along the "equator", the object will appear to be rotating clockwise on the screen.

Arguments:

Argument	Default	Description
elevation	(Mandatory)	The elevation angle in degrees.
rotation	0	The rotation angle in degrees.
twist	0	The twist angle in degrees.

3.47 class CDPyramidLayerMBS

3.47.1 class CDPyramidLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The PyramidLayer class represents pyramid layers (including cone and funnel layers).

Notes: The PyramidLayer object is obtained using PyramidChart.getLayer.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.47.2 Methods

3.47.3 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.47.4 setCenterLabel(labelTemplate as string = "", font as string = "", font-Size as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a label to the center of the pyramid layer.

Notes: To apply this method to all pyramid layers, use CDPyramidChartMBS.setCenterLabel.

See Parameter Substitution and Formatting on available format parameters for the template.

See Font Specification for details on various font attributes.

Arguments:

Argument	Default	Description
template	" { skip } "	The label template. " { skip } " means to keep the existing value unchanged.
font	" { skip } "	The font style. " { skip } " means to keep the existing value unchanged.
fontSize	-1	The font size in points. -1 means to keep the existing value unchanged.
fontColor	-1	The font color. -1 means to keep the existing value unchanged.

Return Value

A CDTextBoxMBS object representing the prototype of the label. This may be used to fine-tune the ap-

pearance of the label.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.47.5 setCenterLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS 627

3.47.5 setCenterLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setCenterLabel method, but uses color instead of integer data type for passing color values.

See also:

- 3.47.4 setCenterLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS 626

3.47.6 setColor(ColorValue as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.47.7 setColor(ColorValue as Integer) 627

3.47.7 setColor(ColorValue as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color of the pyramid layer.

See also:

- 3.47.6 setColor(ColorValue as color) 627

3.47.8 setJoinLine(ColorValue as color, width as Integer = -1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setJoinLine` method, but uses color instead of integer data type for passing color values.

See also:

- 3.47.9 `setJoinLine(ColorValue as Integer, width as Integer = -1)` 628

3.47.9 `setJoinLine(ColorValue as Integer, width as Integer = -1)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color and line width of the join line that joins the pyramid layer with the left and right labels.

Notes: To apply this method to all pyramid layers, use `CDPyramidChartMBS.setJoinLine`.

Arguments:

Argument	Default	Description
color	(Mandatory)	The color of the join line. -1 means to keep the existing value unchanged.
width	-1	The width of the line join. -1 means to keep the existing value unchanged.

See also:

- 3.47.8 `setJoinLine(ColorValue as color, width as Integer = -1)` 627

3.47.10 `setJoinLineGap(pyramidGap as Integer)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gaps of the join line that joins the pyramid layer with the left and right labels.

Notes: To apply this method to all pyramid layers, use `CDPyramidChartMBS.setJoinLineGap`.

By default, the starting point of the join line connects to the pyramid layer edge with a 3 pixels gap. The join line is horizontal, and its length is such that the ending point is 10 pixels outside the pyramid bounding box. The label text box connects to the ending point with a 3 pixels gap.

This method allows the gaps and positions of the join line end points to be configured to other values.

Arguments:

See also:

- 3.47.11 `setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer)` 629

Argument	Default	Description
pyramidGap	(Mandatory)	The gap in pixels between the starting point of the join line and the pyramid layer edge.
pyramidMargin	10	The distance in pixels between the ending point of the join line and the pyramid bounding box.
textGap	3	The gap in pixels between the label text box and the ending point of the join line.

- 3.47.12 setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer, textGap as Integer) 629

3.47.11 setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gaps of the join line that joins the pyramid layer with the left and right labels.

Notes: To apply this method to all pyramid layers, use CDPyramidChartMBS.setJoinLineGap.

By default, the starting point of the join line connects to the pyramid layer edge with a 3 pixels gap. The join line is horizontal, and its length is such that the ending point is 10 pixels outside the pyramid bounding box. The label text box connects to the ending point with a 3 pixels gap.

This method allows the gaps and positions of the join line end points to be configured to other values.

Arguments:

Argument	Default	Description
pyramidGap	(Mandatory)	The gap in pixels between the starting point of the join line and the pyramid layer edge.
pyramidMargin	10	The distance in pixels between the ending point of the join line and the pyramid bounding box.
textGap	3	The gap in pixels between the label text box and the ending point of the join line.

See also:

- 3.47.10 setJoinLineGap(pyramidGap as Integer) 628
- 3.47.12 setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer, textGap as Integer) 629

3.47.12 setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer, textGap as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gaps of the join line that joins the pyramid layer with the left and right labels.

Notes: To apply this method to all pyramid layers, use `CDPyramidChartMBS.setJoinLineGap`.

By default, the starting point of the join line connects to the pyramid layer edge with a 3 pixels gap. The join line is horizontal, and its length is such that the ending point is 10 pixels outside the pyramid bounding box. The label text box connects to the ending point with a 3 pixels gap.

This method allows the gaps and positions of the join line end points to be configured to other values.

Arguments:

Argument	Default	Description
<code>pyramidGap</code>	(Mandatory)	The gap in pixels between the starting point of the join line and the pyramid layer edge.
<code>pyramidMargin</code>	10	The distance in pixels between the ending point of the join line and the pyramid bounding box.
<code>textGap</code>	3	The gap in pixels between the label text box and the ending point of the join line.

See also:

- 3.47.10 `setJoinLineGap(pyramidGap as Integer)` 628
- 3.47.11 `setJoinLineGap(pyramidGap as Integer, pyramidMargin as Integer)` 629

3.47.13 `setLayerBorder(ColorValue as color, width as Integer = -1)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setLayerBorder` method, but uses color instead of integer data type for passing color values.

See also:

- 3.47.14 `setLayerBorder(ColorValue as Integer, width as Integer = -1)` 630

3.47.14 `setLayerBorder(ColorValue as Integer, width as Integer = -1)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color and line width of the layer border.

Notes: To apply this method to all pyramid layers, use `CDPyramidChartMBS.setLayerBorder`.

Arguments:

Argument	Default	Description
color	(Mandatory)	The color of the layer border. -1 means to keep the existing value unchanged.
width	-1	The width of the layer border. -1 means to keep the existing value unchanged.

See also:

- 3.47.13 `setLayerBorder(ColorValue as color, width as Integer = -1)` 630

3.47.15 `setLayerGap(layerGap as Double)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the gap between this pyramid layer and the next pyramid layer.

Notes: To apply this method to all pyramid layers, use `CDPyramidChartMBS.setLayerGap`.

Arguments:

Argument	Default	Description
layerGap	(Mandatory)	The gap between layers as a ratio to the height of the pyramid. The gap must be greater than or equals 0. The sum of all gaps must be less than 1.

3.47.16 `setLeftLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a label to the left of the pyramid layer.

Notes: To apply this method to all pyramid layers, use `CDPyramidChartMBS.setLeftLabel`.

See Parameter Substitution and Formatting on available format parameters for the template.

See Font Specification for details on various font attributes.

Arguments:

Return Value

A `CDTextBoxMBS` object representing the prototype of the label. This may be used to fine-tune the appearance of the label.

Argument	Default	Description
template	" { skip } "	The label template. " { skip } " means to keep the existing value unchanged.
font	" { skip } "	The font style. " { skip } " means to keep the existing value unchanged.
fontSize	-1	The font size in points. -1 means to keep the existing value unchanged.
fontColor	-1	The font color. -1 means to keep the existing value unchanged.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.47.17 `setLeftLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color)` as `CDTextBoxMBS` 632

3.47.17 `setLeftLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color)` as `CDTextBoxMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setLeftLabel` method, but uses color instead of integer data type for passing color values.

See also:

- 3.47.16 `setLeftLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1)` as `CDTextBoxMBS` 631

3.47.18 `setRightLabel(labelTemplate as string = "", font as string = "", font-Size as Double = 8, fontColor as Integer = -1)` as `CDTextBoxMBS`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a label to the right of the pyramid layer.

Notes: To apply this method to all pyramid layers, use `CDPyramidChartMBS.setRightLabel`.

See Parameter Substitution and Formatting on available format parameters for the template.

See Font Specification for details on various font attributes.

Arguments:

Return Value

A `CDTextBoxMBS` object representing the prototype of the label. This may be used to fine-tune the appearance of the label.

Argument	Default	Description
template	" { skip } "	The label template. " { skip } " means to keep the existing value unchanged.
font	" { skip } "	The font style. " { skip } " means to keep the existing value unchanged.
fontSize	-1	The font size in points. -1 means to keep the existing value unchanged.
fontColor	-1	The font color. -1 means to keep the existing value unchanged.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.47.19 `setRightLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS` 633

3.47.19 `setRightLabel(labelTemplate as string, font as string, fontSize as Double, fontColor as color) as CDTextBoxMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setRightLabel` method, but uses color instead of integer data type for passing color values.

See also:

- 3.47.18 `setRightLabel(labelTemplate as string = "", font as string = "", fontSize as Double = 8, fontColor as Integer = -1) as CDTextBoxMBS` 632

3.48 class CDRadialAxisMBS

3.48.1 class CDRadialAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The RadialAxis class represents radial axes in polar charts.

Notes: In the current version of ChartDirector, RadialAxis is implemented as a special configuration of Axis.

Subclass of the CDAxisMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.49 class CDRanSeriesMBS

3.49.1 class CDRanSeriesMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: RanSeries is a utility class to produce random series.

Notes: It facilitates testing and demonstrating ChartDirector programs without needing a real data source.

Blog Entries

- [MBS Xojo Plugins, version 23.6pr1](#)
- [MBS Real Studio Plugins, version 12.3pr16](#)

Xojo Developer Magazine

- [18.5, page 69: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.49.2 Methods

3.49.3 Constructor(seed as Integer)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a RanSeries object.

Notes:

Argument	Default	Description
seed	(Mandatory)	The seed to be used in the random number generator.

3.49.4 create(seed as Integer) as CDRanSeriesMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a RanSeries object.

Notes:

Argument	Default	Description
seed	(Mandatory)	The seed to be used in the random number generator.

3.49.5 fillDateSeries(Values() as Double, startTime as double, tickInc as double, weekDayOnly as Boolean = false)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill an array with date/time values.

Notes:

Argument	Default	Description
ret	(Mandatory)	The array to be filled.
len	(Mandatory)	The number of values to generate.
startTime	(Mandatory)	The first date/time.
tickInc	(Mandatory)	The change in date/time for subsequent records in seconds. The value 30 * 86400 is assumed to mean 1 month (which is not a constant interval), and its multiples are assumed to mean multiple months. In particular, the value 360 * 86400 is assumed to mean 12 months, or 1 year.
weekDayOnly	false	A true value means skipping dates that are not weekdays (Saturday and Sunday). A false value means no skipping.

3.49.6 fillSeries(Values() as Double, minValue as double, maxValue as double)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill an array with independent random numbers.

Notes:

Argument	Default	Description
ret	(Mandatory)	The array to be filled.
len	(Mandatory)	The number of random numbers to generate.
minValue	(Mandatory)	The minimum value of the random number.
maxValue	(Mandatory)	The maximum value of the random number.

See also:

- 3.49.7 fillSeries(Values() as Double, startValue as double, minDelta as double, maxDelta as double, lowerLimit as double = -1E+308, upperLimit as double = 1E+308) 636

3.49.7 fillSeries(Values() as Double, startValue as double, minDelta as double, maxDelta as double, lowerLimit as double = -1E+308, upperLimit as double = 1E+308)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill an array with numbers in which the difference between adjacent numbers is random.

Notes:

Argument	Default	Description
ret	(Mandatory)	The array to be filled.
len	(Mandatory)	The number of random values to generate.
startValue	(Mandatory)	The first value in the sequence.
minDelta	(Mandatory)	The minimum change between two consecutive numbers.
maxDelta	(Mandatory)	The maximum change between two consecutive numbers.
lowerLimit	[-Infinity]	The minimum allowed value of the numbers.
upperLimit	[+Infinity]	The maximum allowed value of the numbers.

See also:

- 3.49.6 fillSeries(Values() as Double, minValue as double, maxValue as double)

636

3.49.8 get2DSeries(xLen as Integer, yLen as Integer, minValue as Double, maxValue as Double) as Double()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets an array of random numbers representing points on a 2D grid in which the difference between neighbouring points is random.

Notes:

Argument	Default	Description
xLen	(Mandatory)	The number of points in the grid in the x direction.
yLen	(Mandatory)	The number of points in the grid in the y direction.
minValue	(Mandatory)	The minimum value of the random number.
maxValue	(Mandatory)	The maximum value of the random number.

Returns an array of random numbers representing points on a 2D grid in which the difference between neighbouring points is random. The array will contain (xLen * yLen) elements. The element at index (y * xLen + x) represents the point at (x, y), in which x and y are integers such that $0 \leq x < xLen$ and $0 \leq y < yLen$.

3.49.9 getDateSeries(len as Integer, startTime as Double, tickInc as Double, weekdayOnly as boolean = false) as Double()

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets an array of date/time values.

Notes:

Returns an array of date/time values.

Argument	Default	Description
len	(Mandatory)	The number of random values to get.
startTime	(Mandatory)	The date/time of the first record in the column.
tickInc	(Mandatory)	The change in date/time for subsequent records in seconds. The value 30 * 86400 is assumed to mean 1 month (which is not a constant interval), and its multiples are assumed to mean multiple months. In particular, the value 360 * 86400 is assumed to mean 12 months, or 1 year.
weekDayOnly	false	A true value means skipping dates that are not weekdays (Saturday and Sunday). A %F value means no skipping.

3.49.10 `getSeries(len as Integer, minValue as Double, maxValue as Double) as Double()`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets an array of numbers that fluctuate randomly.

Notes:

Argument	Default	Description
len	(Mandatory)	The number of random values to get.
startValue	(Mandatory)	The first value in the sequence.
minDelta	(Mandatory)	The minimum change between two consecutive numbers.
maxDelta	(Mandatory)	The maximum change between two consecutive numbers.
lowerLimit	[-Infinity]	The minimum allowed value of the numbers.
upperLimit	[+Infinity]	The maximum allowed value of the numbers.

Returns an array of numbers that fluctuate randomly.

See also:

- 3.49.11 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double) as Double()` 638
- 3.49.12 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double) as Double()` 639
- 3.49.13 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double) as Double()` 640

3.49.11 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double) as Double()`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets an array of numbers that fluctuate randomly.

Notes:

Argument	Default	Description
len	(Mandatory)	The number of random values to get.
startValue	(Mandatory)	The first value in the sequence.
minDelta	(Mandatory)	The minimum change between two consecutive numbers.
maxDelta	(Mandatory)	The maximum change between two consecutive numbers.
lowerLimit	[-Infinity]	The minimum allowed value of the numbers.
upperLimit	[+Infinity]	The maximum allowed value of the numbers.

Returns an array of numbers that fluctuate randomly.

See also:

- 3.49.10 `getSeries(len as Integer, minValue as Double, maxValue as Double) as Double()` 638
- 3.49.12 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double) as Double()` 639
- 3.49.13 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double) as Double()` 640

3.49.12 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double) as Double()`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets an array of numbers that fluctuate randomly.

Notes:

Argument	Default	Description
len	(Mandatory)	The number of random values to get.
startValue	(Mandatory)	The first value in the sequence.
minDelta	(Mandatory)	The minimum change between two consecutive numbers.
maxDelta	(Mandatory)	The maximum change between two consecutive numbers.
lowerLimit	[-Infinity]	The minimum allowed value of the numbers.
upperLimit	[+Infinity]	The maximum allowed value of the numbers.

Returns an array of numbers that fluctuate randomly.

See also:

- 3.49.10 `getSeries(len as Integer, minValue as Double, maxValue as Double) as Double()` 638
- 3.49.11 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double) as Double()` 638
- 3.49.13 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double) as Double()` 640

3.49.13 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double) as Double()`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets an array of numbers that fluctuate randomly.

Notes:

Argument	Default	Description
<code>len</code>	(Mandatory)	The number of random values to get.
<code>startValue</code>	(Mandatory)	The first value in the sequence.
<code>minDelta</code>	(Mandatory)	The minimum change between two consecutive numbers.
<code>maxDelta</code>	(Mandatory)	The maximum change between two consecutive numbers.
<code>lowerLimit</code>	[-Infinity]	The minimum allowed value of the numbers.
<code>upperLimit</code>	[+Infinity]	The maximum allowed value of the numbers.

Returns an array of numbers that fluctuate randomly.

See also:

- 3.49.10 `getSeries(len as Integer, minValue as Double, maxValue as Double) as Double()` 638
- 3.49.11 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double) as Double()` 638
- 3.49.12 `getSeries(len as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double) as Double()` 639

3.50 class CDRanTableMBS

3.50.1 class CDRanTableMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: RanTable is a utility class to produce tables with random numbers.

Notes: It facilitates testing and demonstrating ChartDirector programs without needing a real database table.

Xojo Developer Magazine

- [7.5, pages 34 to 35: Easy Charts and Graphs Part 2, Using the ChartDirector Plugin](#)

3.50.2 Methods

3.50.3 Constructor(seed as Integer, noOfCols as Integer, noOfRows as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a RanTable object.

Notes:

Argument	Default	Description
seed	(Mandatory)	The seed to be used in the random number generator.
noOfCols	(Mandatory)	The number of columns in the random number table.
noOfRows	(Mandatory)	The number of rows in the random number table.

3.50.4 getCol(colNo as Integer) as CDArrayMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the request column as an array of numbers.

Notes:

Argument	Default	Description
i	(Mandatory)	The column to get. The first column is 0. The nth column is (n - 1).

Return Value

An array containing numbers from the requested column.

3.50.5 selectDate(colNo as Integer, minDate as Double, maxDate as Double) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Selects the rows within a certain date range.

Notes: After executing this method, the RanTable will be reduced to only contain rows that are within minDate and maxDate (inclusive of both dates).

Argument	Default	Description
colNo	(Mandatory)	The date column to be used as selection criteria.
minDate	(Mandatory)	The minimum date (earliest date).
maxDate	(Mandatory)	The maximum date (latest date).

3.50.6 setCol(colNo as Integer, minValue as Double, maxValue as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill the given column in the RanTable with random numbers in the given range.

Notes:

Argument	Default	Description
colNo	(Mandatory)	The column to fill. The first column is 0. The nth column is (n - 1).
minValue	(Mandatory)	The minimum value of the random number.
maxValue	(Mandatory)	The maximum value of the random number.

See also:

- 3.50.7 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double) 642
- 3.50.8 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double) 643
- 3.50.9 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double) 644

3.50.7 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill the given column in the RanTable with random numbers that fluctuates within a given speed.

Notes:

Argument	Default	Description
colNo	(Mandatory)	The column to fill. The first column is 0. The nth column is (n - 1).
startValue	(Mandatory)	The value of the first record in the column.
minDelta	(Mandatory)	The minimum change allowed for between the current random number and the previous random number. This parameter is usually negative.
maxDelta	(Mandatory)	The maximum change allowed for between the current random number and the previous random number.
lowerLimit	[-Infinity]	The minimum value of the random number.
upperLimit	[+Infinity]	The maximum value of the random number.

See also:

- 3.50.6 setCol(colNo as Integer, minValue as Double, maxValue as Double) 642
- 3.50.8 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double) 643
- 3.50.9 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double) 644

3.50.8 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill the given column in the RanTable with random numbers that fluctuates within a given speed.

Notes:

Argument	Default	Description
colNo	(Mandatory)	The column to fill. The first column is 0. The nth column is (n - 1).
startValue	(Mandatory)	The value of the first record in the column.
minDelta	(Mandatory)	The minimum change allowed for between the current random number and the previous random number. This parameter is usually negative.
maxDelta	(Mandatory)	The maximum change allowed for between the current random number and the previous random number.
lowerLimit	[-Infinity]	The minimum value of the random number.
upperLimit	[+Infinity]	The maximum value of the random number.

See also:

- 3.50.6 setCol(colNo as Integer, minValue as Double, maxValue as Double) 642
- 3.50.7 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double) 642
- 3.50.9 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double) 644

3.50.9 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill the given column in the RanTable with random numbers that fluctuates within a given speed.

Notes:

Argument	Default	Description
colNo	(Mandatory)	The column to fill. The first column is 0. The nth column is (n - 1).
startValue	(Mandatory)	The value of the first record in the column.
minDelta	(Mandatory)	The minimum change allowed for between the current random number and the previous random number. This parameter is usually negative.
maxDelta	(Mandatory)	The maximum change allowed for between the current random number and the previous random number.
lowerLimit	[-Infinity]	The minimum value of the random number.
upperLimit	[+Infinity]	The maximum value of the random number.

See also:

- 3.50.6 setCol(colNo as Integer, minValue as Double, maxValue as Double) 642
- 3.50.7 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double) 642
- 3.50.8 setCol(colNo as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double) 643

3.50.10 setDateCol(i as Integer, startTime as Double, tickInc as Double, weekDayOnly as boolean=false)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill the given column in the RanTable with a date/time series.

Notes:

Argument	Default	Description
i	(Mandatory)	The column to fill. The first column is 0. The nth column is (n - 1).
startTime	(Mandatory)	The date/time of the first record in the column.
tickInc	(Mandatory)	The change in date/time for subsequent records in seconds. The value 30 * 86400 is assumed to mean 1 month (which is not a constant interval), and its multiplies are assumed to mean multiple months. In particular, the value 360 * 86400 is assumed to mean 12 months, or 1 year.
weekDayOnly	false	A true value means skipping dates that are not weekdays (Saturday and Sunday). A %F value means no skipping.

3.50.11 setHLOCCols(i as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double = 0.0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill 4 columns in the RanTable with random high, low, open and close records.

Notes: The 4 columns will meet the constraints that for each record, the high value will be greater than or equal to the low value, and both the open and close values will be in between the high and low values.

Argument	Default	Description
i	(Mandatory)	The column number of the "high" column. The next column will be assumed to be "low", followed by "open" and "close". The column number starts at 0. The first column is 0. The nth column is (n - 1).
startValue	(Mandatory)	The initial "open" value.
minDelta	(Mandatory)	The minimum change allowed for between the current "open" and the previous "close" value, as well as between the current "close" value and the current "open" value. This parameter is usually negative.
maxDelta	(Mandatory)	The maximum change allowed for between the current "open" and the previous "close" value, as well as between the current "close" value and the current "open" value.
lowerLimit	0	The minimum value for "high", "low", "open" and "close".
upperLimit	[+Infinity]	The maximum value for "high", "low", "open" and "close".

See also:

- 3.50.12 setHLOCCols(i as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double) 645

3.50.12 setHLOCCols(i as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double, upperLimit as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fill 4 columns in the RanTable with random high, low, open and close records.

Notes: The 4 columns will meet the constraints that for each record, the high value will be greater than or equal to the low value, and both the open and close values will be in between the high and low values.

See also:

- 3.50.11 setHLOCCols(i as Integer, startValue as Double, minDelta as Double, maxDelta as Double, lowerLimit as Double = 0.0) 645

Argument	Default	Description
i	(Mandatory)	The column number of the "high" column. The next column will be assumed to be "low", followed by "open" and "close". The column number starts at 0. The first column is 0. The nth column is (n - 1).
startValue	(Mandatory)	The initial "open" value.
minDelta	(Mandatory)	The minimum change allowed for between the current "open" and the previous "close" value, as well as between the current "close" value and the current "open" value. This parameter is usually negative.
maxDelta	(Mandatory)	The maximum change allowed for between the current "open" and the previous "close" value, as well as between the current "close" value and the current "open" value.
lowerLimit	0	The minimum value for "high", "low", "open" and "close".
upperLimit	[+Infinity]	The maximum value for "high", "low", "open" and "close".

3.51 class CDScatterLayerMBS

3.51.1 class CDScatterLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The ScatterLayer class represents scatter layers.

Notes: Subclass of the CDLineLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.52 class CDSectorMBS

3.52.1 class CDSectorMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Sector class represents sectors.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.52.2 Methods

3.52.3 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.52.4 getImageCoor(offsetX as Integer = 0, offsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the image map coordinates of the sector as HTML image map attributes.

Notes: The image map coordinates will be in the following format:

shape="poly" cords=" [x1] , [y1] , [x2] , [y2] ..."

where (x1, y1), (x2, y2) ... are vertices of a polygon that approximate the sector. The format is specially designed so that it can easily be incorporated into HTML image maps.

This method should be called only after creating the chart image (eg. using BaseChart.makeChart, BaseChart.makeChart2 or BaseChart.makeChart3). The image map cannot be determined without creating the chart image first.

Argument	Default	Description
offsetX	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
offsetY	0	An offset to be added to all y coordinates in the image map. See offsetX above for description.

Return Value

A text string representing the coordinates of the sector in HTML image map attribute format.

3.52.5 getLabelCoor(offsetX as Integer = 0, offsetY as Integer = 0) as string

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the image map coordinates of the sector label as HTML image map attributes.

Notes: The image map coordinates will be in the following format:

```
shape="rect" cords=" [ x1 ] , [ y1 ] , [ x2 ] , [ y2 ] "
```

where (x1, y1) and (x2, y2) are opposite corners of the box that encloses the sector label. The format is specially designed so that it can easily be incorporated into HTML image maps.

This method should be called only after creating the chart image (eg. using BaseChart.makeChart, BaseChart.makeChart2 or BaseChart.makeChart3). The image map cannot be determined without creating the chart image first.

Argument	Default	Description
offsetX	0	An offset to be added to all x coordinates in the image map. This is useful if the current image will be shifted and inserted into another image. In this case, the image map will need to be shifted by the same offset.
offsetY	0	An offset to be added to all y coordinates in the image map. See offsetX above for description.

Return Value

A text string representing the coordinates of the sector label in HTML image map attribute format.

3.52.6 setColor(colorvalue as color, edgeColor as color, joinLineColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.52.7 setColor(colorvalue as Integer, edgeColor as Integer = -1, joinLineColor as Integer = -1) 649

3.52.7 setColor(colorvalue as Integer, edgeColor as Integer = -1, joinLineColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the fill color, edge color and join line color of the sector.

Notes: By default, the edge color is SameAsMainColor. For the join line color, if the circular label layout method is used, the default is Transparent, otherwise the default is SameAsMainColor.

Argument	Default	Description
color	(Mandatory)	The color of the sector.
edgeColor	-1	The edge color of the sector. -1 means the color is unchanged.
joinLineColor	-1	The color of the line that join the sector perimeter with the sector label. -1 means the color is unchanged.

See also:

- 3.52.6 setColor(colorvalue as color, edgeColor as color, joinLineColor as color) 649

3.52.8 setExplode(distance as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Explode the sector.

Notes:

Argument	Default	Description
distance	-1	The explosion distance in pixels. -1 means the distance is automatically determined.

3.52.9 setJoinLine(joinLineColor as color, joinLineWidth as Integer = 1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setJoinLine method, but uses color instead of integer data type for passing color values.

See also:

- 3.52.10 setJoinLine(joinLineColor as Integer, joinLineWidth as Integer = 1) 650

3.52.10 setJoinLine(joinLineColor as Integer, joinLineWidth as Integer = 1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color and width of the join line used to connect the sector label to the sector perimeter.

Notes: By default, for circular label layout, the join line color is Transparent. For side label layout, the join line color is SameAsMainColor.

Argument	Default	Description
joinLineColor	(Mandatory)	The color of the line that joins the sector perimeter with the sector label.
joinLineWidth	1	The line width of the join line.

See also:

- 3.52.9 setJoinLine(joinLineColor as color, joinLineWidth as Integer = 1)

650

3.52.11 setLabelFormat(formatString as string = "")

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the format of the sector label.

Example:

```
dim c as CDSectorMBS
```

```
// you can use label formats like this:
```

```
c.setLabelFormat("<*block,halign=left*><*font=timesbi.ttf,size=12,underline=1*>{ label } <*/font*><*br*>US$
{ value } K ( { percent } %)")
```

```
// we can reduce that to this:
```

```
c.setLabelFormat("< { label } { value } { percent } %")
```

```
// and it shows 3 numbers. With | 1 after the variable name, we define the decimals after dot:
```

```
c.setLabelFormat("< { label } { value | 1 } { percent | 1 } %")
```

```
// and
```

```
c.setLabelFormat("< { label } { value | 1., } { percent | 1., } %")
```

```
// uses dot for thousands and comma for decimal separator.
```

Notes: To set the label format for all sectors, use PieChart.setLabelFormat.

Argument	Default	Description
formatString	(Mandatory)	The format string. See PieChart.setLabelFormat for details.

3.52.12 setLabelLayout(layoutMethod as Integer, pos as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the layout method and location of the sector label.

Notes: To set the sector label layout method and location for all sectors, use PieChart.setLabelLayout.

ChartDirector supports two sector label layout methods - circular layout and side layout. For detail descriptions, please refer to PieChart.setLabelLayout.

Argument	Default	Description
layoutMethod	(Mandatory)	Specify the layout method. Must be one of the predefined constants CircleLayout or SideLayout.
pos	-1	For circular layout, it is the distance between the sector perimeter and the sector label. A negative value (but not -1) means the sector label will be drawn in the interior of the sector.

For side layout, it is the distance between the pie perimeter and the left or right edges of the invisible containing rectangle (equal to the width of the rectangle minus the pie diameter and then divided by 2).

In either case, -1 means the distance is automatically determined.

3.52.13 setLabelPos(pos as Integer, joinLineColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setLabelPos method, but uses color instead of integer data type for passing color values.

See also:

- 3.52.14 setLabelPos(pos as Integer, joinLineColor as Integer = -1) 652

3.52.14 setLabelPos(pos as Integer, joinLineColor as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the circular label layout method, and configure the join lines used to connect the sector label to the sector perimeter.

Notes: To set the sector label position or join line color for all sectors, use PieChart.setLabelPos.

See also:

- 3.52.13 setLabelPos(pos as Integer, joinLineColor as color) 652

Argument	Default	Description
pos	(Mandatory)	The distance between the sector perimeter and the sector label. A negative value means the sector label will be drawn in the interior of the sector.
joinLineColor	-1	The color of the line that joins the sector perimeter with the sector label. The default is Transparent. The join line is ignored if the sector label is inside the sector.

3.52.15 setLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002) as CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the style used to draw the sector label.

Notes: To set the label style for all sector labels, use PieChart.setLabelStyle.

See Font Specification for details on various font attributes.

Argument	Default	Description
font	""	The font used to draw the sector label.
fontSize	8	The font size in points.
fontColor	TextColor	The text color for the sector label.

Return Value

A TextBox object representing the prototype of the obj. This may be used to fine-tune the appearance of the obj.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

See also:

- 3.52.16 setLabelStyle(font as string, fontsize as Double, fontcolor as color) as CDTextBoxMBS 653

3.52.16 setLabelStyle(font as string, fontsize as Double, fontcolor as color) as CDTextBoxMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setLabelStyle method, but uses color instead of integer data type for passing color values.

See also:

- 3.52.15 setLabelStyle(font as string = "", fontsize as Double = 8, fontcolor as Integer = &hfff0002) as CDTextBoxMBS 653

3.52.17 `setStyle(shadingMethod as Integer, edgeColor as color, edgeWidth as Integer = -1)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setStyle` method, but uses color instead of integer data type for passing color values.

See also:

- 3.52.18 `setStyle(shadingMethod as Integer, edgeColor as Integer = -1, edgeWidth as Integer = -1)` 654

3.52.18 `setStyle(shadingMethod as Integer, edgeColor as Integer = -1, edgeWidth as Integer = -1)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the sector shading style, edge color and edge width.

Notes: This method is the same as `CDPieChartMBS.setSectorStyle`, except it only affects one sector. Please refer to `CDPieChartMBS.setSectorStyle` for the detail explanation of this method.

Arguments:

Argument	Default	Description
<code>shadingMethod</code>	(Mandatory)	The sector shading style to use. -1 means to keep the existing value unchanged.
<code>edgeColor</code>	-1	The edge color of the sector. -1 means to keep the existing value unchanged.
<code>edgeWidth</code>	-1	The edge width of the sector. -1 means to keep the existing value unchanged.

See also:

- 3.52.17 `setStyle(shadingMethod as Integer, edgeColor as color, edgeWidth as Integer = -1)` 654

3.53 class CDSplineLayerMBS

3.53.1 class CDSplineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The SplineLayer class represents spline layers.

Notes: In the current version of ChartDirector, SplineLayer is implemented as a special configuration of LineLayer.

Subclass of the CDLineLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Xojo Developer Magazine

- [7.4, page 34: Easy Charts and Graphs, Using the ChartDirector Plugin](#)

3.53.2 Methods

3.53.3 setMonotonicity(m as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the monotonicity constraint when computing the spline curve.

Notes: Monotonicity means a sequence is increasing or decreasing in one direction. For example, the sequence 1, 2, 3, 4 is monotonically increasing, while the sequence 1, 3, 2, 5, 4 is not monotonic.

A standard spline curve can overshoot or undershoot if the underlying points have "sharp corners". That means without any constrain, even if the data points are monotonic, the spline curve joining the points is not necessarily monotonic. It can overshoot and then "fall back".

In many charts, it may be desirable to constraint the spline curve to flow in a certain direction. For example, in a time based chart, it may be desirable to constraint the curve always flow from left to right.

ChartDirector supports the following monotonic modes.

Arguments:

3.53.4 setTension(tension as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the tension to use when computing the spline curve.

Notes: The tension parameter should be between -1 and 1. A positive tension will make the spline tighter.

Constant	Value	Description
MonotonicNone	0	The spline curve is not constraint to flow in any direction. This results in the smoothest spline curve.
MonotonicX	1	The spline curve is constraint to not overshooting or undershooting in the x-axis direction. This means if the data points are monotonic in the x-axis direction, the spline curve will also be monotonic in the x-axis direction.
MonotonicY	2	The spline curve is constraint to not overshooting or undershooting in the y-axis direction. This means if the data points are monotonic in the y-axis direction, the spline curve will also be monotonic in the y-axis direction.
MonotonicXY	3	The spline curve is constraint to not overshooting or undershooting in both the x-axis and the y-axis directions. This means if the data points are monotonic in the x-axis direction, the spline curve will also be monotonic in the x-axis direction. If the data points are monotonic in the y-axis direction, the spline curve will also be monotonic in the y-axis direction.
MonotonicAuto	4	Automatically choose between MonotonicX, MonotonicY or MonotonicXY, depending on whether the data points are monotonic in the x-axis direction, or y-axis direction, or both.

Argument	Default	Description
monotonicMode	(Mandatory)	The monotonic constraint of the spline curve. Must be one of the constants in the above table. If this method is never called for a spline layer, the default is MonotonicAuto.

The spline curve will become straight line segments when tension is 1. A negative tension will make the spline looser.

Argument	Default	Description
tension	(Mandatory)	The tension of the spline, which should be between -1 and 1.

3.54 class CDStepLineLayerMBS

3.54.1 class CDStepLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The StepLineLayer class represents step line layers.

Notes: Subclass of the CDLineLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.54.2 Methods

3.54.3 setAlignment(a as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the alignment of the steps relative to the data points.

Notes:

Argument	Default	Description
a	(Mandatory)	The alignment of the steps relative to the data points. Must be one of the predefined constants Left, Center or Right.

3.55 class CDSurfaceChartMBS

3.55.1 class CDSurfaceChartMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The SurfaceChart class represents surface charts.

Notes: Subclass of the CDThreeDChartMBS class.

Xojo Developer Magazine

- [7.5, page 33: Easy Charts and Graphs Part 2, Using the ChartDirector Plugin](#)

3.55.2 Methods

3.55.3 addSurfaceLine(x() as double, y() as double, colorValue as Integer, lineWidth as Integer = -1, side as Integer = 0)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-segment line to the surface.

Notes:

Argument	Default	Description
x	(Mandatory)	An array of containing the x coordinates of the data points that define the line.
y	(Mandatory)	An array of containing the y coordinates of the data points that define the line.
color	(Mandatory)	The color of the line.
lineWidth	1	The line width (thickness).
side	0	A value of 1 means the line is drawn on the front side of the surface. A value of 0 means the line is drawn on the back side of the surface. The default value of 0 means the line is drawn on both sides of the surface.

See also:

- [3.55.4 addSurfaceLine\(x1 as double, y1 as double, x2 as double, y2 as double, colorValue as Integer, lineWidth as Integer = -1, side as Integer = 0\)](#) 658

3.55.4 addSurfaceLine(x1 as double, y1 as double, x2 as double, y2 as double, colorValue as Integer, lineWidth as Integer = -1, side as Integer = 0)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a line segment to the surface.

Notes:

See also:

Argument	Default	Description
x1	(Mandatory)	The x data coordinate of the first end-point of the line.
y1	(Mandatory)	The y data coordinate of the first end-point of the line.
x2	(Mandatory)	The x data coordinate of the second end-point of the line.
y2	(Mandatory)	The y data coordinate of the second end-point of the line.
color	(Mandatory)	The color of the line.
lineWidth	1	The line width (thickness).
side	0	A value of 1 means the line is drawn on the front side of the surface. A value of 0 means the line is drawn on the back side of the surface. The default value of 0 means the line is drawn on both sides of the surface.

- 3.55.3 addSurfaceLine(x() as double, y() as double, colorValue as Integer, lineWidth as Integer = -1, side as Integer = 0) 658

3.55.5 addSurfaceZone(x1 as double, y1 as double, x2 as double, y2 as double, fillColor as Integer, edgeColor as Integer = &hff000000, edgeWidth as Integer = 1)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a rectangular zone to the surface.

Notes:

Argument	Default	Description
x1	(Mandatory)	The x data coordinate for the first corner of the rectangle.
y1	(Mandatory)	The y data coordinate for the first corner of the rectangle.
x2	(Mandatory)	The x data coordinate for the corner opposite to the first corner.
y2	(Mandatory)	The y data coordinate for the corner opposite to the first corner.
fillColor	(Mandatory)	The fill color of the rectangle.
edgeColor	Transparent	The border color of the rectangle.
edgeWidth	1	The border width in pixels.

3.55.6 addXYProjection(offset as Integer = 0)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a projection of the surface on the XY wall.

Notes: The XY wall is the wall containing the x axis and y axis. It is usually under the plotted surface. Because of this, the surface may block the projection, making it hard to see. To address this issue, it is common to add a bottom margin to the z-axis (see Axis.setMargin). This will move the z-axis scale and therefore the surface upwards, leaving more space between the surface and the XY wall to make the projection more visible.

This method accepts an optional offset argument that can be used to move the projection vertically. A

positive offset moves the it upwards. A negative offset moves it under the XY wall. Because of this, for negative offsets, the XY wall must be hidden using `ThreeDChart.setWallVisibility`, otherwise the chart will not be drawn correctly.

Argument	Default	Description
offset	0	The vertical offset of the XY projection in 3D pixel coordinates. A positive offset moves it upwards. A negative offset moves it under the XY wall.

3.55.7 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new SurfaceChart object.

Notes: Arguments:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	kBackgroundColor	The background color of the chart.
edgeColor	kTransparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

See also:

- 3.55.8 Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0) 660

3.55.8 Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other Constructor method, but uses color instead of integer data type for passing color values.

See also:

- 3.55.7 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0) 660

3.55.9 getValuesAtPixel(x as Integer, y as Integer) as Double()

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the (x, y, z, w) values of the surface at the specified image pixel coordinates.

Notes: This method is valid only after the chart image has been drawn (eg. using BaseChart.makeChart, BaseChart.makeChart2 or BaseChart.makeChart3). It is typically used to provide user feedback when the mouse moves on the chart.

If w values are not provided to the surface chart, the returned w value will be set to the same as the z value. If the pixel is not on the plotted surface, the returned array will be empty or null.

Argument	Default	Description
x	(Mandatory)	The x pixel coordinate.
y	(Mandatory)	The y pixel coordinate.

Returns an array of 4 numbers representing the x, y, z and w values at the image pixel location. If the pixel is not on the plotted surface, the returned array will be empty or null.

3.55.10 setBackSideBrightness(brightness as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the back side brightness as a ratio of the front side brightness.

Notes: The surface of a surface chart has two sides. The side that is facing the positive z direction of the plot region is the front side. The side that is facing the negative z direction of the plot region is the back side.

By default, the back side brightness is half the front side brightness. This method can be used to adjust the ratio.

Arguments:

Argument	Default	Description
brightness	(Mandatory)	The ratio of the back side brightness to the front side brightness.

3.55.11 setBackSideColor(ColorValue as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setBackSideColor` method, but uses `color` instead of integer data type for passing color values.

See also:

- 3.55.12 `setBackSideColor(ColorValue as Integer)` 662

3.55.12 `setBackSideColor(ColorValue as Integer)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color of the back side of the surface.

Notes: The surface of a surface chart has two sides. The side that is facing the positive z direction of the plot region is the front side. The side that is facing the negative z direction of the plot region is the back side.

By default, the back side is colored similar to the front side, but with different brightness of lighting parameters. This method can be used to color the back side with a fixed color instead.

Arguments:

Argument	Default	Description
<code>color</code>	(Mandatory)	The color used for the back side.

See also:

- 3.55.11 `setBackSideColor(ColorValue as color)` 661

3.55.13 `setBackSideLighting(ambientLight as Double, diffuseLight as Double, specularLight as Double, shininess as Double)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the Phong lighting parameters for the back side of the surface.

Notes: The surface of a surface chart has two sides. The side that is facing the positive z direction of the plot region is the front side. The side that is facing the negative z direction of the plot region is the back side.

ChartDirector uses the Phong lighting model to adjust the brightness of the surface to make it look realistic. The default lighting parameters for the back side is half the brightness of those for the front side. This method may be used to set alternative lighting parameters for the back side.

Arguments:

Argument	Default	Description
ambientIntensity	(Mandatory)	The ambient reflection coefficient of the Phong lighting model.
diffuseIntensity	(Mandatory)	The diffuse reflection coefficient of the Phong lighting model.
specularIntensity	(Mandatory)	The specular reflection coefficient of the Phong lighting model.
shininess	(Mandatory)	The shininess coefficient of the Phong lighting model.

3.55.14 setContourColor(contourColor as color, minorContourColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setContourColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.55.15 setContourColor(contourColor as Integer, minorContourColor as Integer = -1) 663

3.55.15 setContourColor(contourColor as Integer, minorContourColor as Integer = -1)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors of the contour lines on the surface.

Notes: The contour lines are lines that join regions of the same z value on the surface. The contour levels are determined by the ticks on the z-axis. Major ticks associate with major contour lines. Minor ticks associate with minor contour lines.

Arguments:

Argument	Default	Description
contourColor	(Mandatory)	The color of the major contour lines.
minorContourColor	-1	The color of the minor contour lines. -1 means it is the same as the contourColor.

See also:

- 3.55.14 setContourColor(contourColor as color, minorContourColor as color) 663

3.55.16 setData(xData() as Double, yData() as Double, zData() as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data for the surface chart.

Notes: ChartDirector supports both gridded and scattered data. If the data points are on a rectangular grid

will no missing points, they will be handled as gridded data. Otherwise, they will be handled as scattered data.

For gridded data, you may provide the x and y values of the grid, and the z values of the data points. For a 10 x 15 grid, that means the x data series should have 10 values, the y data series should have 15 values, and the z data series should have 150 values. The x and y data series should be strictly monotonic (either strictly increasing or strictly decreasing).

For both gridded and scattered data, you may also provide the (x, y, z) values of the data points. For example, for 150 data points, the x, y and z data series should each have 150 values. ChartDirector will automatically detect if the data points are gridded or scattered.

Arguments:

Argument	Default	Description
xData	(Mandatory)	An array of numbers representing the x data series.
yData	(Mandatory)	An array of numbers representing the y data series.
zData	(Mandatory)	An array of numbers representing the z data series.

See also:

- 3.55.17 setData(xData() as double, yData() as double, zData() as double, wData() as double) 664

3.55.17 setData(xData() as double, yData() as double, zData() as double, wData() as double)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the data for the surface chart.

See also:

- 3.55.16 setData(xData() as Double, yData() as Double, zData() as Double) 663

3.55.18 setInterpolation(xSamples as Integer, ySamples as Integer = -1, isSmooth as Boolean = true)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables and sets the degree and method of interpolation of the data points.

Notes: A surface chart consists of planar patches, drawn using the data points as the vertices. To create visually smooth surfaces, we may use a large number of small planar patches. However, this requires a large number of data points.

For example, consider a plot region of which the x and y dimensions are both 400 pixels in length. To create a smooth surface, if gridded data are used, the distance between grid lines should be no more than a few pixels. Suppose a grid spacing of 8 pixels is used. The grid size will be 51 x 51, and 2601 data points will be needed.

If the number of available data points are too small to create a smooth surface, This method can be used to generate a denser grid of data points from the original data points using interpolation.

Two types of interpolation are supported. In spline surface interpolation, ChartDirector will compute a smooth surface that passes through the original data points, and sample the spline surface for the new data points. In bilinear/linear interpolation, ChartDirector will apply bilinear/linear interpolation to its vertices of the original patches to compute the new data points.

Arguments:

Argument	Default	Description
xSamples	(Mandatory)	The number of samples on the x-dimension to interpolate to.
ySamples	-1	The number of samples on the y-dimension to interpolate to. -1 means it is the same as xSamples.
isSmooth	true	A true value means to use spline surface interpolation. A false value means to use bilinear/linear interpolation.

See also:

- 3.55.19 `setInterpolation(xSamples as Integer, ySamples as Integer, isSmooth as Boolean, isColorSmooth as Boolean)` 665

3.55.19 `setInterpolation(xSamples as Integer, ySamples as Integer, isSmooth as Boolean, isColorSmooth as Boolean)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables and sets the degree and method of interpolation of the data points.

Notes: A surface chart consists of planar patches, drawn using the data points as the vertices. To create visually smooth surfaces, we may use a large number of small planar patches. However, this requires a large number of data points.

For example, consider a plot region of which the x and y dimensions are both 400 pixels in length. To create a smooth surface, if gridded data are used, the distance between grid lines should be no more than a few pixels. Suppose a grid spacing of 8 pixels is used. The grid size will be 51 x 51, and 2601 data points will be needed.

If the number of available data points are too small to create a smooth surface, This method can be used to generate a denser grid of data points from the original data points using interpolation.

Two types of interpolation are supported. In spline surface interpolation, ChartDirector will compute a smooth surface that passes through the original data points, and sample the spline surface for the new data points. In bilinear/linear interpolation, ChartDirector will apply bilinear/linear interpolation to its vertices of the original patches to compute the new data points.

Argument	Default	Description
xSamples	(Mandatory)	The number of samples on the x-dimension to interpolate to.
ySamples	-1	The number of samples on the y-dimension to interpolate to. -1 means it is the same as xSamples.
isSmooth	true	A true value means to use spline surface interpolation for the z values. A false value means to use bilinear/linear interpolation.
isColorSmooth	(Optional)	A true value means to use spline surface interpolation for the w values. A false value means to use bilinear/linear interpolation. If this argument is not specified, the default is to use the same interpolation method as the z values.

See also:

- 3.55.18 `setInterpolation(xSamples as Integer, ySamples as Integer = -1, isSmooth as Boolean = true)`
664

3.55.20 `setLighting(ambientIntensity as Double, diffuseIntensity as Double, specularIntensity as Double, shininess as Double)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the Phong lighting parameters for the front side of the surface.

Notes: The surface of a surface chart has two sides. The side that is facing the positive z direction of the plot region is the front side. The side that is facing the negative z direction of the plot region is the back side.

ChartDirector uses the Phong lighting model to adjust the brightness of the surface to make it look realistic. The light source is assumed to be from the direction of the viewer and is far away. The default lighting parameters for the front side are 0.5 for both ambient and diffuse reflections, 1 for specular reflection and 8 for shininess.

Arguments:

3.55.21 `setShadingMode(shadingMode as Integer, wireWidth as Integer = 1)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Argument	Default	Description
ambientIntensity	(Mandatory)	The ambient reflection coefficient of the Phong lighting model.
diffuseIntensity	(Mandatory)	The diffuse reflection coefficient of the Phong lighting model.
specularIntensity	(Mandatory)	The specular reflection coefficient of the Phong lighting model.
shininess	(Mandatory)	The shininess coefficient of the Phong lighting model.

Function: Sets the shading mode for the surface.

Notes: ChartDirector supports the following methods for shading the surface.

Constant	Value	Description
kSmoothShading	0	The brightness of the surface varies smoothly.
kTriangularShading	0	The surface is divided into triangular patches. The brightness of the patches varies according to their orientation. Within a patch, the brightness is constant.
kRectangularShading	0	The surface is divided into rectangular patches. The brightness of the patches varies according to their orientation. Within a patch, the brightness is constant.
kTriangularFrame	0	The surface is divided into triangular patches. Only the edges of the patches are drawn.
kRectangularFrame	0	The surface is divided into rectangular patches. Only the edges of the patches are drawn.

Arguments:

Argument	Default	Description
shadingMode	(Mandatory)	The shading mode, which must be one of the above constants.
wireWidth	1	The wire width for use with wireframe shading modes.

3.55.22 setSurfaceAxisGrid(majorXGridColor as color, majorYGridColor as color, minorXGridColor as color, minorYGridColor as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setSurfaceAxisGrid method, but uses color instead of integer data type for passing color values.

See also:

- 3.55.23 setSurfaceAxisGrid(majorXGridColor as Integer, majorYGridColor as Integer = -1, minorXGridColor as Integer = -1, minorYGridColor as Integer = -1) 668

3.55.23 `setSurfaceAxisGrid(majorXGridColor as Integer, majorYGridColor as Integer = -1, minorXGridColor as Integer = -1, minorYGridColor as Integer = -1)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors of the axis grid lines on the surface.

Notes: Axis grid lines are grid lines associated with the ticks on the x and y axes. Major grid lines associate with major ticks. Minor grid lines associate with minor ticks. They can be drawn on the surface of the surface chart, and on the plot region walls. This method is for the surface of the surface chart. For the plot region walls, see `CDSurfaceChartMBS.setWallGrid`.

Arguments:

Argument	Default	Description
<code>majorXGridColor</code>	(Mandatory)	The color of the major x-axis grid lines on the surface.
<code>majorYGridColor</code>	-1	The color of the major y-axis grid lines on the surface. -1 means it is the same as the <code>majorXGridColor</code> .
<code>minorXGridColor</code>	-1	The color of the minor x-axis grid lines on the surface. -1 means it is the same as the <code>majorXGridColor</code> .
<code>minorYGridColor</code>	-1	The color of the minor y-axis grid lines on the surface. -1 means it is the same as the <code>majorYGridColor</code> .

See also:

- 3.55.22 `setSurfaceAxisGrid(majorXGridColor as color, majorYGridColor as color, minorXGridColor as color, minorYGridColor as color)` 667

3.55.24 `setSurfaceDataGrid(xGridColor as color, yGridColor as color)`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setSurfaceDataGrid` method, but uses color instead of integer data type for passing color values.

See also:

- 3.55.25 `setSurfaceDataGrid(xGridColor as Integer, yGridColor as Integer = -1)` 668

3.55.25 `setSurfaceDataGrid(xGridColor as Integer, yGridColor as Integer = -1)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors of the data grid lines on the surface.

Notes: Data grid lines are grid lines that define the positions of the data points. They can be drawn on the

surface of the surface chart.

Arguments:

Argument	Default	Description
xGridColor	(Mandatory)	The color of the x data grid lines.
yGridColor	-1	The color of the y data grid lines. -1 means it is the same as the yGridColor.

See also:

- 3.55.24 setSurfaceDataGrid(xGridColor as color, yGridColor as color) 668

3.55.26 setSurfaceTexture(patternColor as color)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a texture image to the surface.

Notes: A texture is a 2D image pattern for mapping onto a 3D surface. The pattern can be defined by using BaseChart.patternColor or BaseChart.patternColor2.

One common usage of this method is to apply an image the same size as the plot region xy plane to the surface. To do this, simply use an image with width and height equals to the plot region width and depth.

Argument	Default	Description
patternColor	(Mandatory)	A color (normally a pattern color) that contains the texture image to be applied to the surface.

See also:

- 3.55.27 setSurfaceTexture(patternColor as integer) 669

3.55.27 setSurfaceTexture(patternColor as integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a texture image to the surface.

Notes: A texture is a 2D image pattern for mapping onto a 3D surface. The pattern can be defined by using BaseChart.patternColor or BaseChart.patternColor2.

One common usage of this method is to apply an image the same size as the plot region xy plane to the surface. To do this, simply use an image with width and height equals to the plot region width and depth.

Argument	Default	Description
patternColor	(Mandatory)	A color (normally a pattern color) that contains the texture image to be applied to the surface.

See also:

- 3.55.26 `setSurfaceTexture(patternColor as color)` 669

3.55.28 `setWContourColor(wContourColor as color, wMinorContourColor as color)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors of the w contour lines on the surface.

Notes: The w contour lines are lines that join regions of the same w value on the surface. The contour levels are determined by the ticks on the color axis. Major ticks associate with major contour lines. Minor ticks associate with minor contour lines.

Argument	Default	Description
wContourColor	(Mandatory)	The color of the major contour lines.
wMinorContourColor	-1	The color of the minor contour lines. -1 means it is the same as the contourColor.

See also:

- 3.55.29 `setWContourColor(wContourColor as integer, wMinorContourColor as Integer = -1)` 670

3.55.29 `setWContourColor(wContourColor as integer, wMinorContourColor as Integer = -1)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the colors of the w contour lines on the surface.

Notes: The w contour lines are lines that join regions of the same w value on the surface. The contour levels are determined by the ticks on the color axis. Major ticks associate with major contour lines. Minor ticks associate with minor contour lines.

Argument	Default	Description
wContourColor	(Mandatory)	The color of the major contour lines.
wMinorContourColor	-1	The color of the minor contour lines. -1 means it is the same as the contourColor.

See also:

3.55. CLASS CDSURFACECHARTMBS

671

- 3.55.28 setWContourColor(wContourColor as color, wMinorContourColor as color)

670

3.56 class CDTextBoxMBS

3.56.1 class CDTextBoxMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The TextBox class represents text boxes.

Notes: Subclass of the CDBoxMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [Chart Diagrams with Xojo](#)

Xojo Developer Magazine

- [7.4, pages 34 to 35: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [7.4, page 29: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [20.4, page 39: PDF Pie Charts, Adding Xojo Charts to Your PDFs by Stefanie Juchmes](#)
- [17.2, page 35: More Beyond JSON, Develop APIs That Generate Barcodes, Charts, and More by Timothy Dietrich](#)

3.56.2 Methods

3.56.3 setAlignment(a as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the alignment of the text relative to the container box.

Notes:

Argument	Default	Description
a	(Mandatory)	The alignment specification. See Alignment Specification for supported alignment types.

3.56.4 setFontAngle(angle as Double, vertical as boolean=false)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the rotation angle and layout direction of the text.

Notes: See Font Specification for details on various font attributes.

Argument	Default	Description
angle	(Mandatory)	The text rotation angle. Rotation is measured in counter-clockwise direction in degrees.
vertical	false	Determine if the font is layout horizontally (from left to right) or vertically (from top to down). Vertical layout is common for Oriental languages such as Chinese, Japanese and Korean. A true value means vertical layout. A false value means horizontal layout.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

3.56.5 setFontColor(colorvalue as color)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other setFontColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.56.6 setFontColor(colorvalue as Integer)

673

3.56.6 setFontColor(colorvalue as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color of the text.

Notes: If this method is not called, TextColor will be used to draw the text.

Argument	Default	Description
color	(Mandatory)	The font color.

See also:

- 3.56.5 setFontColor(colorvalue as color)

673

3.56.7 setFontSize(fontHeight as Double, fontWidth as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the font height and width in points.

Notes: See Font Specification for details on various font attributes.

Argument	Default	Description
fontHeight	(Mandatory)	The font height in points.
fontWidth	0	The font width in points. If the font width is zero, it is assumed to be the same as the font height.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

3.56.8 setFontStyle(font as string, fontIndex as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the font of the text by specifying the file that contains the font.

Notes: See Font Specification for details on various font attributes.

Argument	Default	Description
font	(Mandatory)	The font name.
fontIndex	0	The font index in case the font name refers to a font collection. An index of 0 means the first font.

See font specification here:

<http://www.monkeybreadsoftware.net/faq-chartdirectorfontspecification.shtml>

3.56.9 setHeight(height as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the height of the text box.

Notes: By default, the height of the text box is automatically determined to be just enough to hold the text. The setHeight method can be used to specified a fix height.

Argument	Default	Description
height	(Mandatory)	The height of the text box in pixels.

3.56.10 setMargin(leftMargin as Integer, rightMargin as Integer, topMargin as Integer, bottomMargin as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the margins of the text box in pixels.

Notes: The margins of a text box are the distances between the borders of the text box to the text inside. By default, the left and right margins are approximately half the font width, and the top and bottom margins are approximately 1/4 of the font height.

Argument	Default	Description
leftMargin	(Mandatory)	The left margin in pixels.
rightMargin	(Mandatory)	The right margin in pixels.
topMargin	(Mandatory)	The top margin in pixels.
bottomMargin	(Mandatory)	The bottom margin in pixels.

See also:

- 3.56.11 setMargin(m as Integer)

675

3.56.11 setMargin(m as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets all margins (left, right, top, and bottom) of the text box to the same value.

Notes: The margins of a text box are the distances between the borders of the text box to the text inside. By default, the left and right margins are approximately half the font width, and the top and bottom margins are approximately 1/4 of the font height.

Argument	Default	Description
m	(Mandatory)	The left, right, top and bottom margins in pixels.

See also:

- 3.56.10 setMargin(leftMargin as Integer, rightMargin as Integer, topMargin as Integer, bottomMargin as Integer)

674

3.56.12 setMaxWidth(width as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the maximum width of the text box and wraps text if necessary.

Notes: By default, the width of the text box is automatically determined to be the length of the text. This method can be used to limit the width of the text box. If the text requires a longer width for display, it will be wrapped into multiple lines.

Argument	Default	Description
maxWidth	(Mandatory)	The maximum width of the text box in pixels.

3.56.13 setText(text as string)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the text to be shown in the text box.

Notes: See ChartDirector Mark Up Language on how to embed special tags in the text for sophisticated formatting.

Argument	Default	Description
text	(Mandatory)	The text to be displayed in the text box.

3.56.14 setTruncate(maxWidth as Integer, maxLines as Integer = 1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the maximum number of lines in the text box, and truncate the text if it exceeds the line count.

Notes: The maxWidth parameter of this method specifies the maximum width of the text box. If the text requires a longer width for display, it will be wrapped into multiple lines. However, if the number of lines exceed maxLines, the text will be truncated, with the last visible characters replaced with "...".

If maxLines is 1 (the default), the text will never be wrapped and will be truncated directly if it needs more than maxWidth for display.

Argument	Default	Description
maxWidth	(Mandatory)	The maximum width of the text box in pixels.
maxLines	1	The maximum number of lines that the text box can have. The text will be truncated if it exceeds the maximum number of lines.

3.56.15 setWidth(width as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the width of the text box and wraps text if necessary.

Notes: By default, the width of the text box is automatically determined to be the length of the text. The setWidth method can be used to specified a fix width. If the text requires a longer width for display, the text will be wrapped into multiple lines.

Argument	Default	Description
width	(Mandatory)	The width of the text box in pixels.

3.57 class CDThreeDChartMBS

3.57.1 class CDThreeDChartMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The ThreeDChart class is an abstract class containing methods that are common to all true 3D chart types.

Notes: ThreeDChart is a subclass of BaseChart.

Subclass of the CDBaseChartMBS class.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Real Studio Plugins, version 12.3pr16](#)

3.57.2 Methods

3.57.3 colorAxis as CDColorAxisMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the ColorAxis object representing the color axis (color legend).

Notes: Returns the ColorAxis object representing the color axis.

3.57.4 Constructor

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

Notes: The ThreeDChart class is an abstract class containing methods that are common to all true 3D chart types.

ThreeDChart is a subclass of BaseChart.

This constructor is private to make sure you don't create an object from this class by error. Please use designated functions to create objects.

3.57.5 getElevationAngle as double

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the elevation angle.

3.57.6 getPlotRegionDepth as integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the plot region depth.

3.57.7 getPlotRegionHeight as integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the plot region height.

3.57.8 getPlotRegionWidth as integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the plot region width.

3.57.9 getRotationAngle as double

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the rotation angle.

3.57.10 getXCoord(xValue as Double, yValue as Double, zValue as Double) as Integer

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x pixel coordinate of a point given its x, y and z data values.

Notes: Note: You must call BaseChart.layout first before calling this method. ChartDirector needs to perform auto-scaling and layout the axis before it can compute pixel coordinates from data values.

Argument	Default	Description
x	(Mandatory)	The x data value of the point.
y	(Mandatory)	The y data value of the point.
z	(Mandatory)	The z data value of the point.

Returns the x pixel coordinate of the point.

3.57.11 `getYCoord(xValue as Double, yValue as Double, zValue as Double) as Integer`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y pixel coordinate of a point given its x, y and z data values.

Notes: You must call `BaseChart.layout` first before calling this method. `ChartDirector` needs to perform auto-scaling and layout the axis before it can compute pixel coordinates from data values.

Argument	Default	Description
x	(Mandatory)	The x data value of the point.
y	(Mandatory)	The y data value of the point.
z	(Mandatory)	The z data value of the point.

Returns the y pixel coordinate of the point.

3.57.12 `setColorAxis(x as Integer, y as Integer, alignment as Integer, length as Integer, orientation as Integer) as CDCColorAxisMBS`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position, length and orientation of the color axis (color legend).

Notes:

Argument	Default	Description
x	(Mandatory)	The x-coordinate of the reference point used to position the color axis.
y	(Mandatory)	The y-coordinate of the reference point used to position the color axis.
alignment	(Mandatory)	The alignment of the color axis with respect to the reference point. For example, a value of <code>TopLeft</code> means the reference point is the top- left corner of the color axis. See <code>Alignment Specification</code> for supported alignment types..
length	(Mandatory)	The length of the color axis in pixels.
orientation	(Mandatory)	The orientation of the color axis. A value of <code>Top/Bottom</code> means the axis is horizontal, and the axis labels are at top/bottom side of the axis. A value of <code>Left/Right</code> means the axis is vertical, and the axis labels are at the left/right side of the axis.

3.57.13 `setPerspective(perspective as Double)`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the strength of the perspective effect.

Notes: When taking the photograph of a scene, things closer to the camera are larger in the photograph, while things farther from the camera are smaller. For example, two trees of exactly the same height but at

different distances from the camera will have different heights as measured in the photograph. This is known as the perspective effect.

This method configures the strength of the perspective effect for a ThreeDChart object. The nearer part of the plot region will be magnified, while the farther part will be reduced. The average size of the plot region will be approximately unchanged.

Argument	Default	Description
perspective	(Mandatory)	The strength of the perspective effect. It approximately represents the maximum percentage magnification of the plot region, which is the same as the maximum percentage reduction. A value of 0 means no perspective effect. This value is usually between 0 to 100. If this method is never called, the default is 12.

3.57.14 setPlotRegion(cx as Integer, cy as Integer, xWidth as Integer, yDepth as Integer, zHeight as Integer)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position of the plot region.

Notes: The plot region is a 3D box, with x, y, and z axes along its border, using a right-hand cartesian coordinate system.

To create the chart image, you may imagine a camera taking a photograph of the box. The photograph can be taken at different angles, configurable with ThreeDChart.setViewAngle. The position of the plot region in the photograph is such that the center of the plot region is at a given point (cx, cy) on the image.

Argument	Default	Description
cx	(Mandatory)	The x pixel coordinate of the point on the image at which the center of the plot region is located.
cy	(Mandatory)	The y pixel coordinate of the point on the image at which the center of the plot region is located.
xWidth	(Mandatory)	The length of the x-dimension of the plot region in pixels.
yDepth	(Mandatory)	The length of the y-dimension of the plot region in pixels.
zHeight	(Mandatory)	The length of the z-dimension of the plot region in pixels.

3.57.15 setViewAngle(elevation as Double, rotation as Double = 0, twist as Double = 0)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the 3D view angles.

Notes: ChartDirector uses elevation, rotation and twist angles to specify the 3D view angles.

To explain the meaning of the angles, imagine the object being viewed is put at the center of a hollow sphere, and a camera (the view point) is put at the surface of the sphere, directed inwards to the center to look at the object.

The elevation angle refers to the "latitude" of the camera. An elevation angle of 0 degrees means the camera is at the "equator" pointing to the object from the side. An elevation angle of 90 degrees means the camera is at the "north pole" pointing down to the object. An angle elevation of -90 degrees means the camera is at the "south pole", pointing up to the object.

The rotation angle refers to the "longitude" of the camera. If the elevation is 0 degrees, and the rotation angle varies from 0 to 360 degrees, the camera will move around the "equator" in the easterly direction (counter-clockwise when viewed from the north pole).

The twist angle is for rotating the camera itself while still pointing to the object. For example, a twist angle of 90 degrees means you are holder the camera "vertically" instead of "horizontally". The rotation is clockwise from the view point of the person holding the camera.

Note that from the view point of the camera, the object will appear to be rotating in the opposite direction. For example, as the rotation angle rotates the camera counter-clockwise along the "equator", the object will appear to be rotating clockwise on the screen.

Note: The twist angle is not supported in the current version of ChartDirector. It should always be 0, and should be considered as a reserved argument for future use.

Argument	Default	Description
elevation	(Mandatory)	The elevation angle in degrees.
rotation	0	The rotation angle in degrees.
twist	0	The twist angle in degrees.

3.57.16 `setWallColor(xyColor as color, yzColor as color, zxColor as color, borderColor as color)`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color of the plot region walls.

Notes: By default, the plot region walls are light grey (eeeeee) in color, with a grey (888888) border . This method can be used to modify the colors.

Argument	Default	Description
xyColor	(Mandatory)	The color of the xy wall.
yzColor	-1	The color of the yz wall. -1 means it is the same as xyColor.
zxColor	-1	The color of the zx wall. -1 means it is the same as xyColor.
borderColor	-1	The color of the wall border. -1 means to keep the existing value unchanged.

See also:

- 3.57.17 setWallColor(xyColor as Integer, yzColor as Integer = -1, zxColor as Integer = -1, borderColor as Integer = -1) 683

3.57.17 setWallColor(xyColor as Integer, yzColor as Integer = -1, zxColor as Integer = -1, borderColor as Integer = -1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the color of the plot region walls.

Notes: By default, the plot region walls are light grey (eeeeee) in color, with a grey (888888) border . This method can be used to modify the colors.

Argument	Default	Description
xyColor	(Mandatory)	The color of the xy wall.
yzColor	-1	The color of the yz wall. -1 means it is the same as xyColor.
zxColor	-1	The color of the zx wall. -1 means it is the same as xyColor.
borderColor	-1	The color of the wall border. -1 means to keep the existing value unchanged.

See also:

- 3.57.16 setWallColor(xyColor as color, yzColor as color, zxColor as color, borderColor as color) 682

3.57.18 setWallGrid(majorXGridColor as color, majorYGridColor as color, majorZGridColor as color, minorXGridColor as color, minorYGridColor as color, minorZGridColor as color)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the grid line colors on plot region walls.

Notes: The grid lines on the walls are associated with the ticks on the x, y and z axes. Major ticks associate with major grid lines. Minor ticks associate with minor grid lines. By default, the major grid lines are light grey (cccccc), and the minor grid lines are lighter grey (dddddd). This method can be used to modify the grid line colors.

See also:

Argument	Default	Description
majorXGridColor	(Mandatory)	The color of the major x-axis grid lines on the walls.
majorYGridColor	-1	The color of the major y-axis grid lines on the walls. -1 means it is the same as the majorXGridColor.
majorZGridColor	-1	The color of the major z-axis grid lines on the walls. -1 means it is the same as the majorXGridColor.
minorXGridColor	-1	The color of the minor x-axis grid lines on the walls. -1 means it is the same as the majorXGridColor.
minorYGridColor	-1	The color of the minor y-axis grid lines on the walls. -1 means it is the same as the majorYGridColor.
minorZGridColor	-1	The color of the minor z-axis grid lines on the walls. -1 means it is the same as the majorZGridColor.

- 3.57.19 setWallGrid(majorXGridColor as Integer, majorYGridColor as Integer = -1, majorZGridColor as Integer = -1, minorXGridColor as Integer = -1, minorYGridColor as Integer = -1, minorZGridColor as Integer = -1) 684

3.57.19 setWallGrid(majorXGridColor as Integer, majorYGridColor as Integer = -1, majorZGridColor as Integer = -1, minorXGridColor as Integer = -1, minorYGridColor as Integer = -1, minorZGridColor as Integer = -1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the grid line colors on plot region walls.

Notes: The grid lines on the walls are associated with the ticks on the x, y and z axes. Major ticks associate with major grid lines. Minor ticks associate with minor grid lines. By default, the major grid lines are light grey (cccccc), and the minor grid lines are lighter grey (dddddd). This method can be used to modify the grid line colors.

Argument	Default	Description
majorXGridColor	(Mandatory)	The color of the major x-axis grid lines on the walls.
majorYGridColor	-1	The color of the major y-axis grid lines on the walls. -1 means it is the same as the majorXGridColor.
majorZGridColor	-1	The color of the major z-axis grid lines on the walls. -1 means it is the same as the majorXGridColor.
minorXGridColor	-1	The color of the minor x-axis grid lines on the walls. -1 means it is the same as the majorXGridColor.
minorYGridColor	-1	The color of the minor y-axis grid lines on the walls. -1 means it is the same as the majorYGridColor.
minorZGridColor	-1	The color of the minor z-axis grid lines on the walls. -1 means it is the same as the majorZGridColor.

See also:

- 3.57.18 setWallGrid(majorXGridColor as color, majorYGridColor as color, majorZGridColor as color,

minorXGridColor as color, minorYGridColor as color, minorZGridColor as color)

3.57.20 setWallThickness(xyThickness as Integer, yzThickness as Integer = -1, zxThickness as Integer = -1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the thickness of the plot region walls.

Notes: By default, the plot region walls are 10 pixels thick. This method can be used to modify the thickness.

Argument	Default	Description
xyThickness	(Mandatory)	The thickness of the xy wall in pixels.
yzThickness	-1	The thickness of the yz wall in pixels. -1 means it is the same as xyThickness.
zxThickness	-1	The thickness of the zx wall in pixels. -1 means it is the same as xyThickness.

3.57.21 setWallVisibility(xyVisible as boolean, yzVisible as boolean, zxVisible as boolean)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shows or hides the plot region walls.

Notes: By default, all walls are visible. This method can be used to hide some or all of the walls.

Argument	Default	Description
xyVisible	(Mandatory)	A true value means the xy wall is visible. A false value means the xy wall is hidden.
yzVisible	(Mandatory)	A true value means the yz wall is visible. A false value means the yz wall is hidden.
zxVisible	(Mandatory)	A true value means the zx wall is visible. A false value means the zx wall is hidden.

3.57.22 setZAxisPos(pos as Integer)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position of the z-axis.

Notes: In a ThreeDChart object, the z-axis can be drawn either on the left side or the right side. By default, ChartDirector will draw the z-axis on the side that is nearer to the viewer.

This method can be used to specify which side the z-axis should be drawn.

Argument	Default	Description
pos	(Mandatory)	The position of the z-axis, which must be either Left or Right.

3.57.23 Properties

3.57.24 xAxis as CDAxisMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the x-axis of the ThreeDChart object.

Notes: Returns the Axis object representing the x-axis of the ThreeDChart object.
(Read only property)

3.57.25 yAxis as CDAxisMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the y-axis of the ThreeDChart object.

Notes: Returns the Axis object representing the y-axis of the ThreeDChart object.
(Read only property)

3.57.26 zAxis as CDAxisMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the z-axis of the ThreeDChart object.

Notes: Returns the Axis object representing the z-axis of the ThreeDChart object.
(Read only property)

3.58 class CDThreeDScatterChartMBS

3.58.1 class CDThreeDScatterChartMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The ThreeDScatterChart class represents 3D scatter charts.

Notes: ThreeDScatterChart is a subclass of ThreeDChart.

Subclass of the CDThreeDChartMBS class.

Blog Entries

- [3D Scatter Charts](#)

3.58.2 Methods

3.58.3 addScatterGroup(xData() as Double, yData() as Double, zData() as Double, name as string = "", symbol as Integer = 7, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDThreeDScatterGroupMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a group of scatter symbols to the ThreeDScatterChart.

Notes: A scatter chart can be considered as a special configuration of a line chart, in the data symbols are enabled and the line width is set to zero. Therefore only the data symbols are visible and the chart appears as scattered.

Argument	Default	Description
xData	(Mandatory)	An array of numbers representing the x values of the data points.
yData	(Mandatory)	An array of numbers representing the y values of the data points.
zData	(Mandatory)	An array of numbers representing the z values of the data points.
name	""	The name of the symbol group. The name will be used in the legend box, if one is available. An empty string means the symbol group has no name.
symbol	CircleShape	One of the predefined symbol constants to specify the symbol to use. (See Shape Specification for available built-in shapes.)
symbolSize	5	The width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette. SameAsMainColor means the color is based on the z value of the symbol as according to the ColorAxis (accessible via ThreeDChart.colorAxis).
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means using LineColor as the edge color.

Returns a ThreeDScatterGroup object representing the symbol group created.

See also:

- 3.58.4 `addScatterGroup(xData() as Double, yData() as Double, zData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDThreeDScatterGroupMBS` 688

3.58.4 `addScatterGroup(xData() as Double, yData() as Double, zData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDThreeDScatterGroupMBS`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a group of scatter symbols to the `ThreeDScatterChart`.

Notes: A scatter chart can be considered as a special configuration of a line chart, in the data symbols are enabled and the line width is set to zero. Therefore only the data symbols are visible and the chart appears as scattered.

Argument	Default	Description
<code>xData</code>	(Mandatory)	An array of numbers representing the x values of the data points.
<code>yData</code>	(Mandatory)	An array of numbers representing the y values of the data points.
<code>zData</code>	(Mandatory)	An array of numbers representing the z values of the data points.
<code>name</code>	""	The name of the symbol group. The name will be used in the legend box, if one is available. An empty string means the symbol group has no name.
<code>symbol</code>	<code>CircleShape</code>	One of the predefined symbol constants to specify the symbol to use. (See Shape Specification for available built-in shapes.)
<code>symbolSize</code>	5	The width and height of the symbol in pixels.
<code>fillColor</code>	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette. <code>SameAsMainColor</code> means the color is based on the z value of the symbol as according to the <code>ColorAxis</code> (accessible via <code>ThreeDChart.colorAxis</code>).
<code>edgeColor</code>	-1	The edge color used to draw the edge of the symbol. -1 means using <code>LineColor</code> as the edge color.

Returns a `ThreeDScatterGroup` object representing the symbol group created.

See also:

- 3.58.3 `addScatterGroup(xData() as Double, yData() as Double, zData() as Double, name as string = "", symbol as Integer = 7, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDThreeDScatterGroupMBS` 687

3.58.5 `Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0)`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new ThreeDScatterChart object.

Notes:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	BackgroundColor	The background color of the chart.
edgeColor	Transparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

See also:

- 3.58.6 Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0) 689

3.58.6 Constructor(width as Integer, height as Integer, bgColor as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new ThreeDScatterChart object.

Notes:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	BackgroundColor	The background color of the chart.
edgeColor	Transparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

See also:

- 3.58.5 Constructor(width as Integer = 640, height as Integer = 480, bgColor as Integer = &hfff0000, edgeColor as Integer = &hff000000, raisedEffect as Integer = 0) 688

3.59 class CDThreeDScatterGroupMBS

3.59.1 class CDThreeDScatterGroupMBS

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The ThreeDScatterGroup class represents a group of scatter symbols in a ThreeDScatterChart object.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Xojo Plugins, version 18.5pr7](#)
- [MBS Real Studio Plugins, version 12.4pr3](#)
- [MBS Real Studio Plugins, version 12.3pr16](#)

3.59.2 Methods

3.59.3 Constructor

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.59.4 setDataSymbol(DrawArea as CDDrawAreaMBS)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a DrawArea object as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
obj	(Mandatory)	A DrawArea object to be used as the symbol.

See also:

- 3.59.5 setDataSymbol(file as folderitem) 691
- 3.59.6 setDataSymbol(ImageFilePath as string) 692
- 3.59.7 setDataSymbol(pic as Picture) 692
- 3.59.8 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color) 693

- 3.59. CLASS CDTHREEDSCATTERGROUPMBS 691
- 3.59.9 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 694
- 3.59.10 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 695
- 3.59.11 setDataSymbol(symbol as Integer, size as Integer, fillColor as color) 695
- 3.59.12 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 696

3.59.5 setDataSymbol(file as folderitem)

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load an image from a file and use it as the graphics symbol to plot the data points.

Notes: ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to BaseChart.setSearchPath on the directory that ChartDirector will search for the file.

Argument	Default	Description
image	(Mandatory)	The filename of the image file. The image type is determined based on file extension, which must be png, jpg/jpeg, gif or wbmp/wmp.

See also:

- 3.59.4 setDataSymbol(DrawArea as CDDrawAreaMBS) 690
- 3.59.6 setDataSymbol(ImageFilePath as string) 692
- 3.59.7 setDataSymbol(pic as Picture) 692
- 3.59.8 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color) 693
- 3.59.9 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 694
- 3.59.10 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 695
- 3.59.11 setDataSymbol(symbol as Integer, size as Integer, fillColor as color) 695
- 3.59.12 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 696

3.59.6 setDataSymbol(ImageFilePath as string)

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Load an image from a file and use it as the graphics symbol to plot the data points.

Notes: ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to BaseChart.setSearchPath on the directory that ChartDirector will search for the file.

Argument	Default	Description
image	(Mandatory)	The filename of the image file. The image type is determined based on file extension, which must be png, jpg/jpeg, gif or wbmp/wmp.

See also:

- 3.59.4 setDataSymbol(DrawArea as CDDrawAreaMBS) 690
- 3.59.5 setDataSymbol(file as folderitem) 691
- 3.59.7 setDataSymbol(pic as Picture) 692
- 3.59.8 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color) 693
- 3.59.9 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 694
- 3.59.10 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 695
- 3.59.11 setDataSymbol(symbol as Integer, size as Integer, fillColor as color) 695
- 3.59.12 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 696

3.59.7 setDataSymbol(pic as Picture)

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a picture object as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
obj	(Mandatory)	A picture object to be used as the symbol.

See also:

3.59. CLASS CDTHREEDSCATTERGROUPMBS	693
• 3.59.4 setDataSymbol(DrawArea as CDDrawAreaMBS)	690
• 3.59.5 setDataSymbol(file as folderitem)	691
• 3.59.6 setDataSymbol(ImageFilePath as string)	692
• 3.59.8 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color)	693
• 3.59.9 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1)	694
• 3.59.10 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1)	695
• 3.59.11 setDataSymbol(symbol as Integer, size as Integer, fillColor as color)	695
• 3.59.12 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1)	696

3.59.8 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a custom polygon as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
polygon	(Mandatory)	An array of integers representing the coordinates the polygon vertices. See Shape Specification on how the custom shape is defined.
size	11	The nominal width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette. SameAsMainColor means the color is based on the z value of the symbol as according to the ColorAxis (accessible via ThreeD-Chart.colorAxis).
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means using LineColor as the edge color.

See also:

• 3.59.4 setDataSymbol(DrawArea as CDDrawAreaMBS)	690
• 3.59.5 setDataSymbol(file as folderitem)	691
• 3.59.6 setDataSymbol(ImageFilePath as string)	692
• 3.59.7 setDataSymbol(pic as Picture)	692
• 3.59.9 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1)	694

- 3.59.10 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 695
- 3.59.11 setDataSymbol(symbol as Integer, size as Integer, fillColor as color) 695
- 3.59.12 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 696

3.59.9 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a custom polygon as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
polygon	(Mandatory)	An array of integers representing the coordinates the polygon vertices. See Shape Specification on how the custom shape is defined.
size	11	The nominal width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette. SameAsMainColor means the color is based on the z value of the symbol as according to the ColorAxis (accessible via ThreeD-Chart.colorAxis).
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means using LineColor as the edge color.

See also:

- 3.59.4 setDataSymbol(DrawArea as CDDrawAreaMBS) 690
- 3.59.5 setDataSymbol(file as folderitem) 691
- 3.59.6 setDataSymbol(ImageFilePath as string) 692
- 3.59.7 setDataSymbol(pic as Picture) 692
- 3.59.8 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color) 693
- 3.59.10 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 695
- 3.59.11 setDataSymbol(symbol as Integer, size as Integer, fillColor as color) 695
- 3.59.12 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 696

3.59.10 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a custom polygon as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
polygon	(Mandatory)	An array of integers representing the coordinates the polygon vertices. See Shape Specification on how the custom shape is defined.
size	11	The nominal width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette. SameAsMainColor means the color is based on the z value of the symbol as according to the ColorAxis (accessible via ThreeD-Chart.colorAxis).
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means using LineColor as the edge color.

See also:

- 3.59.4 setDataSymbol(DrawArea as CDDrawAreaMBS) 690
- 3.59.5 setDataSymbol(file as folderitem) 691
- 3.59.6 setDataSymbol(ImageFilePath as string) 692
- 3.59.7 setDataSymbol(pic as Picture) 692
- 3.59.8 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color) 693
- 3.59.9 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 694
- 3.59.11 setDataSymbol(symbol as Integer, size as Integer, fillColor as color) 695
- 3.59.12 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 696

3.59.11 setDataSymbol(symbol as Integer, size as Integer, fillColor as color)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a custom polygon as the graphics symbol to plot the data points.

Notes:

See also:

- 3.59.4 setDataSymbol(DrawArea as CDDrawAreaMBS) 690

Argument	Default	Description
polygon	(Mandatory)	An array of integers representing the coordinates the polygon vertices. See Shape Specification on how the custom shape is defined.
size	11	The nominal width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette. SameAsMainColor means the color is based on the z value of the symbol as according to the ColorAxis (accessible via ThreeD-Chart.colorAxis).
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means using LineColor as the edge color.

- 3.59.5 setDataSymbol(file as folderitem) 691
- 3.59.6 setDataSymbol(ImageFilePath as string) 692
- 3.59.7 setDataSymbol(pic as Picture) 692
- 3.59.8 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color) 693
- 3.59.9 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 694
- 3.59.10 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 695
- 3.59.12 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1) 696

3.59.12 setDataSymbol(symbol as Integer, size as Integer, fillColor as color, edgeColor as color, lineWidth as Integer = 1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses a custom polygon as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
polygon	(Mandatory)	An array of integers representing the coordinates the polygon vertices. See Shape Specification on how the custom shape is defined.
size	11	The nominal width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette. SameAsMainColor means the color is based on the z value of the symbol as according to the ColorAxis (accessible via ThreeD-Chart.colorAxis).
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means using LineColor as the edge color.

3.59. CLASS CDTHREEDSCATTERGROUPMBS 697

See also:

- 3.59.4 setDataSymbol(DrawArea as CDDrawAreaMBS) 690
- 3.59.5 setDataSymbol(file as folderitem) 691
- 3.59.6 setDataSymbol(ImageFilePath as string) 692
- 3.59.7 setDataSymbol(pic as Picture) 692
- 3.59.8 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as color, edgeColor as color) 693
- 3.59.9 setDataSymbol(polygon() as Integer, size as Integer = 11, fillColor as Integer = -1, edgeColor as Integer = -1) 694
- 3.59.10 setDataSymbol(symbol as Integer, size as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1, lineWidth as Integer = 1) 695
- 3.59.11 setDataSymbol(symbol as Integer, size as Integer, fillColor as color) 695

3.59.13 setDropLine

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the drop line color and width.

Notes: Drop lines are vertical lines that join the data points to the bottom of the plot region. It helps in visualizing the height of the data points.

Argument	Default	Description
dropLineColor	(Mandatory)	The color of the drop line.
dropLineWidth	1	The width of the drop line.

See also:

- 3.59.14 setDropLine(dropLineColor as color, dropLineWidth as Integer = 1) 697
- 3.59.15 setDropLine(dropLineColor as Integer, dropLineWidth as Integer = 1) 698

3.59.14 setDropLine(dropLineColor as color, dropLineWidth as Integer = 1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the drop line color and width.

Notes: Drop lines are vertical lines that join the data points to the bottom of the plot region. It helps in visualizing the height of the data points.

See also:

Argument	Default	Description
dropLineColor	(Mandatory)	The color of the drop line.
dropLineWidth	1	The width of the drop line.

- 3.59.13 setDropLine 697
- 3.59.15 setDropLine(dropLineColor as Integer, dropLineWidth as Integer = 1) 698

3.59.15 setDropLine(dropLineColor as Integer, dropLineWidth as Integer = 1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the drop line color and width.

Notes: Drop lines are vertical lines that join the data points to the bottom of the plot region. It helps in visualizing the height of the data points.

Argument	Default	Description
dropLineColor	(Mandatory)	The color of the drop line.
dropLineWidth	1	The width of the drop line.

See also:

- 3.59.13 setDropLine 697
- 3.59.14 setDropLine(dropLineColor as color, dropLineWidth as Integer = 1) 697

3.59.16 setLegendIcon(width as Integer, height as Integer = -1, color as Integer = -1)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses one of the built-in symbols as the graphics symbol to plot the data points.

Notes:

See also:

- 3.59.17 setLegendIcon(width as Integer, height as Integer, color as color) 698

3.59.17 setLegendIcon(width as Integer, height as Integer, color as color)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses one of the built-in symbols as the graphics symbol to plot the data points.

Notes:

Argument	Default	Description
symbol	(Mandatory)	One of the predefined shape constants representing the symbol shape. See Shape Specification for the available built-in shapes.
size	5	The width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette. kSameAsMainColor means the color is based on the z value of the symbol as according to the ColorAxis (accessible via ThreeD-Chart.colorAxis).
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means using LineColor as the edge color.
lineWidth	1	The line width used for drawing the symbols.

Argument	Default	Description
symbol	(Mandatory)	One of the predefined shape constants representing the symbol shape. See Shape Specification for the available built-in shapes.
size	5	The width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette. kSameAsMainColor means the color is based on the z value of the symbol as according to the ColorAxis (accessible via ThreeD-Chart.colorAxis).
edgeColor	-1	The edge color used to draw the edge of the symbol. -1 means using LineColor as the edge color.
lineWidth	1	The line width used for drawing the symbols.

See also:

- 3.59.16 setLegendIcon(width as Integer, height as Integer = -1, color as Integer = -1) 698

3.59.18 setSymbolOffset(offsetX as Integer, offsetY as Integer)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Offset the symbols in the x and y directions in pixel unit.

Notes:

Argument	Default	Description
xOffset	(Mandatory)	The x offset in pixels. A positive value mean shifting to the right.
yOffset	(Mandatory)	The y offset in pixels. A positive value mean shifting to the bottom.

3.60 class CDTreeMapChartMBS

3.60.1 class CDTreeMapChartMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The TreeMapChart class represents tree map charts.

Notes: Subclass of the CDBaseChartMBS class.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.2](#)

3.60.2 Methods

3.60.3 Constructor(width as integer = 640, height as integer = 480, bgColor as integer = &hfff0000, edgeColor as integer = &hff000000, raisedEffect as integer = 0)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Creates a new TreeMapChart object.

Notes:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	BackgroundColor	The background color of the chart.
edgeColor	Transparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

Returns the TreeMapChart object created.

See also:

- [3.60.4 Constructor\(width as integer, height as integer, bgColor as color, edgeColor as color, raisedEffect as integer = 0\)](#) 700

3.60.4 Constructor(width as integer, height as integer, bgColor as color, edgeColor as color, raisedEffect as integer = 0)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Creates a new TreeMapChart object.

Notes:

Argument	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	BackgroundColor	The background color of the chart.
edgeColor	Transparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

Returns the TreeMapChart object created.

See also:

- 3.60.3 Constructor(width as integer = 640, height as integer = 480, bgColor as integer = &hfff0000, edgeColor as integer = &hff000000, raisedEffect as integer = 0) 700

3.60.5 getLevelPrototype(index as integer) as CDTreeMapNodeMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the prototype node for the specified level.

Notes: A tree map starts with the root node. The root node contains level 1 child nodes. The level 1 child nodes can in turn contain level 2 child nodes, and so on.

A prototype node is a node that acts as the template for nodes at a certain level. For example, if the level 1 prototype node is configured to use a certain font for labels, all level 1 nodes will use that font as the default for labels.

Argument	Default	Description
level	(Mandatory)	The node level to which the prototype node applies.

Returns TreeMapNode object representing the prototype node for the specified level.

3.60.6 setMapLevel(n as integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the node level used to generate the image map.

Notes: By default, the image map will be based on the deepest node level. For example, if the tree map has 2 levels of nodes, the image map will be based on the level 2 nodes. The setMapLevel method can be used to specify an alternative node level for the image map.

Argument	Default	Description
level	(Mandatory)	The node level used to generate the image map.

3.60.7 setPlotArea(x as integer, y as integer, width as integer, height as integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the position and size of the tree map plot area.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of the top-left corner of the plot area.
y	(Mandatory)	The y coordinate of the top-left corner of the plot area.
width	(Mandatory)	The width of the plot area.
height	(Mandatory)	The height of the plot area.

3.60.8 Properties

3.60.9 RootNode as CDTreeMapNodeMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the root node of the tree map.

Notes: The root node is the starting node of the tree map. Child nodes can be added to the root node, and a second level of child nodes can be added to the first level child nodes, and so on.

(Read only property)

3.61 class CDTreeMapNodeMBS

3.61.1 class CDTreeMapNodeMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The TreeMapNode class represents tree map nodes.

Notes: The TreeMapChart.getRootNode method can be used to obtain the root node of a tree map chart. TreeMapNode.setData can then be used to add child nodes.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.5](#)
- [MBS Xojo Plugins, version 23.5pr7](#)
- [News from the MBS Xojo Plugins Version 21.2](#)

Xojo Developer Magazine

- [22.1, page 9: News](#)

3.61.2 Methods

3.61.3 addExtraField(numbers() as double)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds an array of numbers/dates to be used as an extra field in various places.

Notes: This method merely stores the data inside the node object. The Parameter Substitution and Formatting mechanism will determine how the data are to be used.

A common use for extra fields is to specify extra information (such as a custom serial number for the data points) to be displayed on data labels or on tool tips, or to supply extra query parameters in clickable charts. All these are achieved by specifying the extra field on the data label template or image map templates during parameter substitution.

Argument	Default	Description
numbers	(Mandatory)	An array of numbers/dates an extra field of the child nodes.

See also:

- [3.61.4 addExtraField\(texts\(\) as string\)](#)

3.61.4 addExtraField(texts() as string)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Adds an array of text to be used as an extra field of the child nodes.

Notes: This method merely stores the data inside the node object. The Parameter Substitution and Formatting mechanism will determine how the data are to be used.

A common use for extra fields is to specify extra information (such as a custom serial number for the data points) to be displayed on data labels or on tool tips, or to supply extra query parameters in clickable charts. All these are achieved by specifying the extra field on the data label template or image map templates during parameter substitution.

Argument	Default	Description
texts	(Mandatory)	An array of text to be an extra field of the child nodes.

See also:

- 3.61.3 addExtraField(numbers() as double)

703

3.61.5 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The private constructor.

3.61.6 Node(Index as Integer) as CDTreeMapNodeMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the child node at the specified index.

Notes:

Argument	Default	Description
i	(Mandatory)	The index of the child node.

Returns the TreeMapNode object representing the child node at the specified index.

3.61.7 setColors(fillColor as color, edgeColor as color, raisedEffect as integer = -2147483647)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the fill and border colors and 3D border effect of the node.

Notes:

Argument	Default	Description
fillColor	(Mandatory)	The fill color of the node.
edgeColor	(Optional)	The border color of the node.
raisedEffect	(Optional)	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument can also be used to specify CDBaseChartMBS.glassEffect, CDBaseChartMBS.softLighting, CDBaseChartMBS.cylinderEffect or CDBaseChartMBS.flatBorder effects.

See also:

- 3.61.8 setColors(fillColor as integer, edgeColor as integer = -1, raisedEffect as integer = -2147483647) 705

3.61.8 setColors(fillColor as integer, edgeColor as integer = -1, raisedEffect as integer = -2147483647)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the fill and border colors and 3D border effect of the node.

Notes:

Argument	Default	Description
fillColor	(Mandatory)	The fill color of the node.
edgeColor	(Optional)	The border color of the node.
raisedEffect	(Optional)	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat. This argument can also be used to specify CDBaseChartMBS.glassEffect, CDBaseChartMBS.softLighting, CDBaseChartMBS.cylinderEffect or CDBaseChartMBS.flatBorder effects.

See also:

- 3.61.7 setColors(fillColor as color, edgeColor as color, raisedEffect as integer = -2147483647) 705

3.61.9 setData(data() as double)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the data for the child nodes.

Notes:

Argument	Default	Description
data	(Mandatory)	An array of numbers for the values of the child node. This array can be empty if the child nodes contains other child nodes. In this case, the value of a node will be determined as the sum of the values of its child nodes.

See also:

- 3.61.10 setData(data() as double, labels() as string) 706
- 3.61.11 setData(data() as double, labels() as string, colors() as Color) 706
- 3.61.12 setData(data() as double, labels() as string, colors() as Integer) 707

3.61.10 setData(data() as double, labels() as string)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the data for the child nodes.

Notes:

Argument	Default	Description
data	(Mandatory)	An array of numbers for the values of the child node. This array can be empty if the child nodes contains other child nodes. In this case, the value of a node will be determined as the sum of the values of its child nodes.
labels	[Empty_Array]	An array of text strings for the node labels.

See also:

- 3.61.9 setData(data() as double) 705
- 3.61.11 setData(data() as double, labels() as string, colors() as Color) 706
- 3.61.12 setData(data() as double, labels() as string, colors() as Integer) 707

3.61.11 setData(data() as double, labels() as string, colors() as Color)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the data for the child nodes.

Notes:

See also:

- 3.61.9 setData(data() as double) 705

Argument	Default	Description
data	(Mandatory)	An array of numbers for the values of the child node. This array can be empty if the child nodes contains other child nodes. In this case, the value of a node will be determined as the sum of the values of its child nodes.
labels	[Empty_Array]	An array of text strings for the node labels.
colors	[Empty_Array]	An array of colors to be used as the fill colors of the child nodes. An empty array means the colors are determined by the prototype node (See CDTreeMapChartMBS.getLevelPrototype).

- 3.61.10 setData(data() as double, labels() as string) 706
- 3.61.12 setData(data() as double, labels() as string, colors() as Integer) 707

3.61.12 setData(data() as double, labels() as string, colors() as Integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the data for the child nodes.

Notes:

Argument	Default	Description
data	(Mandatory)	An array of numbers for the values of the child node. This array can be empty if the child nodes contains other child nodes. In this case, the value of a node will be determined as the sum of the values of its child nodes.
labels	[Empty_Array]	An array of text strings for the node labels.
colors	[Empty_Array]	An array of colors to be used as the fill colors of the child nodes. An empty array means the colors are determined by the prototype node (See CDTreeMapChartMBS.getLevelPrototype).

See also:

- 3.61.9 setData(data() as double) 705
- 3.61.10 setData(data() as double, labels() as string) 706
- 3.61.11 setData(data() as double, labels() as string, colors() as Color) 706

3.61.13 setLabelFormat(formatString as string = " { label } ", font as string = "normal", fontSize as Integer = 10, fontColor as Integer = &hfff0002, alignment as Integer = 7)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the node label format and font style.

Notes: Please refer to Parameter Substitution and Formatting in FAQ on all available format parameters.

Argument	Default	Description
format	(Mandatory)	The format string.
font	"normal"	The font used to draw the label.
fontSize	10	The font size used to draw the labels in points.
fontColor	TextColor	The color used to draw the labels.
alignment	TopLeft	The alignment of the label inside the node. See Alignment Specification for supported alignment types.

See also:

- 3.61.14 `setLabelFormat(formatString as string, font as string, fontSize as Integer, fontColor as Color, alignment as Integer = 7)` 708

3.61.14 `setLabelFormat(formatString as string, font as string, fontSize as Integer, fontColor as Color, alignment as Integer = 7)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the node label format and font style.

Notes: Please refer to Parameter Substitution and Formatting in FAQ on all available format parameters.

Argument	Default	Description
format	(Mandatory)	The format string.
font	"normal"	The font used to draw the label.
fontSize	10	The font size used to draw the labels in points.
fontColor	TextColor	The color used to draw the labels.
alignment	TopLeft	The alignment of the label inside the node. See Alignment Specification for supported alignment types.

See also:

- 3.61.13 `setLabelFormat(formatString as string = " { label } ", font as string = "normal", fontSize as Integer = 10, fontColor as Integer = &hffff0002, alignment as Integer = 7)` 707

3.61.15 `setLayoutAspectMultiplier(multiplier as Double)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets layout aspect multiplier.

3.61.16 setLayoutAspectRatio(ratio as Double)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Set the layout aspect ratio.

3.61.17 setLayoutMethod(layoutMethod as Integer, layoutDirection as Integer = -1)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the layout method used to layout the child nodes.

Notes: The layout method is specified by using the following predefined constants as the layoutMethod argument.

If a layout method argument is not specified, it will be obtained from the prototype node (see `TreeMapChart.getLevel-Prototype`). If it is still not specified, the parent node will be used. If the node is the root node with no parent, the default is to use `TreeMapSquarify` with `kTopLeft` layout direction and without `swapXY`.

Argument	Default	Description
layoutMethod	(Mandatory)	The layout method used to layout the child nodes. A value of 0 means the layout method is not specified.
layoutDirection	(Optional)	The layout direction. Must be one of <code>kTopLeft</code> , <code>kTopRight</code> , <code>kBottomLeft</code> and <code>kBottomRight</code> . By default, the horizontal direction will be applied first, followed by the vertical direction.
swapXY	(Optional)	If set to true, the vertical direction will be applied first, followed by the horizontal direction. A false value means the opposite.

See also:

- 3.61.18 `setLayoutMethod(layoutMethod as Integer, layoutDirection as Integer, swapXY as Boolean)`
709

3.61.18 setLayoutMethod(layoutMethod as Integer, layoutDirection as Integer, swapXY as Boolean)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the layout method used to layout the child nodes.

Notes: The layout method is specified by using the following predefined constants as the layoutMethod argument.

If a layout method argument is not specified, it will be obtained from the prototype node (see `TreeMapChart.getLevel-Prototype`). If it is still not specified, the parent node will be used. If the node is the root node with no parent, the default is to use `TreeMapSquarify` with `kTopLeft` layout direction and without `swapXY`.

Argument	Default	Description
<code>layoutMethod</code>	(Mandatory)	The layout method used to layout the child nodes. A value of 0 means the layout method is not specified.
<code>layoutDirection</code>	(Optional)	The layout direction. Must be one of <code>kTopLeft</code> , <code>kTopRight</code> , <code>kBottomLeft</code> and <code>kBottomRight</code> . By default, the horizontal direction will be applied first, followed by the vertical direction.
<code>swapXY</code>	(Optional)	If set to true, the vertical direction will be applied first, followed by the horizontal direction. A false value means the opposite.

See also:

- 3.61.17 `setLayoutMethod(layoutMethod as Integer, layoutDirection as Integer = -1)` 709

3.61.19 `setPos(x as Integer, y as Integer, w as Integer, h as Integer)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the position of the node.

3.61.20 `setSorting(Mode as Integer)`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Layout the nodes in sorted order.

Notes: `ChartDirector` supports a number of layout methods (see `TreeMapNode.setLayoutMethod`), most of them will layout the nodes in the order they appear in the data array. The `setSorting` can be used to layout the nodes in sorted order instead.

Argument	Default	Description
<code>mode</code>	(Mandatory)	The values -1, 0, 1 refer to descending, no sorting and ascending order.

3.61.21 Properties

3.61.22 `BottomY as Integer`

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the bottom y coordinate of the node.

Notes: This method should be used only after calling BaseChart.layout or after ChartDirector has output the chart image.

(Read only property)

3.61.23 Handle as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The internal object reference.

Notes: (Read only property)

3.61.24 Height as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the height of the node.

Notes: This method should be used only after calling BaseChart.layout or after ChartDirector has output the chart image.

(Read only property)

3.61.25 Label as String

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the the label value of the node.

Notes: (Read only property)

3.61.26 LeftX as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the left x coordinate of the node.

Notes: This method should be used only after calling BaseChart.layout or after ChartDirector has output the chart image.

(Read only property)

3.61.27 NodeCount as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the the number of child nodes.

Notes: (Read only property)

3.61.28 RightX as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the right x coordinate of the node.

Notes: This method should be used only after calling BaseChart.layout or after ChartDirector has output the chart image.

(Read only property)

3.61.29 TopY as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the top y coordinate of the node.

Notes: This method should be used only after calling BaseChart.layout or after ChartDirector has output the chart image.

(Read only property)

3.61.30 Value as Double

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the the data value of the node.

Notes: (Read only property)

3.61.31 Width as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the width of the node.

Notes: This method should be used only after calling BaseChart.layout or after ChartDirector has output the chart image.

(Read only property)

3.62 class CDTrendLayerMBS

3.62.1 class CDTrendLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The TrendLayer class represents trend layers.

Notes: The trend layer performs linear regression analysis on the data points, and represents the result as a best fit straight line with optional confidence and prediction bands.

In linear regression analysis, the data points are assumed to be related by:

$$y = m * x + c + \text{err}$$

where m and c are constants, and err is a random variable.

Linear regression analysis estimates m , c and err based on available data using the least square method. Using estimated values of m and c , the line $y = m * x + c$ are plotted as the best fit straight line based on available data.

However, as m and c are estimations based on available data, it may not be exactly equal to the "real" m and c . In ChartDirector, the uncertainties are represented visually as a confidence band around the regression line. For example, the 95% confidence band means there are 95% probability that the "real" line is in that band.

To predict a data point (infer y given x), we can use the formula:

$$y = m * x + c + \text{err}$$

based on estimated values of m , c and err .

The uncertainties of the data point is contributed by the uncertainties in m and c , as well as err . In ChartDirector, the uncertainties of the data points are represented visually as a prediction band around the regression line. For example, a 95% prediction band means there are 95% probability that a data point will be in that band.

The prediction band is always wider than the confidence band. It is because the uncertainties of the regression line is contributed by m and c , while the uncertainties of the data points are contributed by m , c and err . The err term makes the data points less certain that the regression line.

Subclass of the CDLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [Chart Diagrams with Xojo](#)

3.62.2 Methods

3.62.3 addConfidenceBand(confidence as Double, upperFillColor as color, upperEdgeColor as color, upperLineWidth as Integer, lowerFillColor as color, lowerEdgeColor as color, lowerLineWidth as Integer = -1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addConfidenceBand method, but uses color instead of integer data type for passing color values.

See also:

- 3.62.4 addConfidenceBand(confidence as Double, upperFillColor as Integer, upperEdgeColor as Integer = &hFF000000, upperLineWidth as Integer = -1, lowerFillColor as Integer = -1, lowerEdgeColor as Integer = -1, lowerLineWidth as Integer = -1) 714

3.62.4 addConfidenceBand(confidence as Double, upperFillColor as Integer, upperEdgeColor as Integer = &hFF000000, upperLineWidth as Integer = -1, lowerFillColor as Integer = -1, lowerEdgeColor as Integer = -1, lowerLineWidth as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a confidence band to the trend layer.

Notes: Please refer to the description of TrendLayer on what is a confidence band.

Argument	Default	Description
confidence	(Mandatory)	The confidence level - must be between 0 - 1.
upperFillColor	(Mandatory)	The fill color for the upper side of the confidence band (the portion that is above the regression line).
upperEdgeColor	Transparent	The border color for the upper side of the confidence band (the portion that is above the regression line).
upperLineWidth	1	The border width for the upper side of the confidence band (the portion that is above the regression line).
lowerFillColor	-1	The fill color for the lower side of the confidence band (the portion that is below the regression line). -1 means the color is the same as upperFillColor.
lowerEdgeColor	-1	The border color for the lower side of the confidence band (the portion that is below the regression line). -1 means the color is the same as upperEdgeColor.
lowerLineWidth	-1	The border width for the lower side of the confidence band (the portion that is below the regression line). -1 means the color is the same as upperLineWidth.

See also:

- 3.62.3 addConfidenceBand(confidence as Double, upperFillColor as color, upperEdgeColor as color, upperLineWidth as Integer, lowerFillColor as color, lowerEdgeColor as color, lowerLineWidth as Integer = -1) 714

3.62.5 addPredictionBand(confidence as Double, upperFillColor as color, upperEdgeColor as color, upperLineWidth as Integer, lowerFillColor as color, lowerEdgeColor as color, lowerLineWidth as Integer = -1)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addPredictionBand method, but uses color instead of integer data type for passing color values.

See also:

- 3.62.6 addPredictionBand(confidence as Double, upperFillColor as Integer, upperEdgeColor as Integer = &hFF000000, upperLineWidth as Integer = -1, lowerFillColor as Integer = -1, lowerEdgeColor as Integer = -1, lowerLineWidth as Integer = -1) 715

3.62.6 addPredictionBand(confidence as Double, upperFillColor as Integer, upperEdgeColor as Integer = &hFF000000, upperLineWidth as Integer = -1, lowerFillColor as Integer = -1, lowerEdgeColor as Integer = -1, lowerLineWidth as Integer = -1)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a prediction band to the trend layer.

Notes: Please refer to the description of TrendLayer on what is a prediction band.

Argument	Default	Description
confidence	(Mandatory)	The confidence level - must be between 0 - 1.
upperFillColor	(Mandatory)	The fill color for the upper side of the prediction band (the portion that is above the regression line).
upperEdgeColor	Transparent	The border color for the upper side of the prediction band (the portion that is above the regression line).
upperLineWidth	1	The border width for the upper side of the prediction band (the portion that is above the regression line).
lowerFillColor	-1	The fill color for the lower side of the prediction band (the portion that is below the regression line). -1 means the color is the same as upperFillColor.
lowerEdgeColor	-1	The border color for the lower side of the prediction band (the portion that is below the regression line). -1 means the color is the same as upperEdgeColor.
lowerLineWidth	-1	The border width for the lower side of the prediction band (the portion that is below the regression line). -1 means the color is the same as upperLineWidth.

See also:

- 3.62.5 addPredictionBand(confidence as Double, upperFillColor as color, upperEdgeColor as color, upperLineWidth as Integer, lowerFillColor as color, lowerEdgeColor as color, lowerLineWidth as Integer = -1) 715

3.62.7 getCoefficient(index as Integer) as Double

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the coefficients of the regression function.

Notes: The coefficients depend on the regression type. They are indexed as a0, a1, a2, ... in the regression type table published on TrendLayer.

3.62.8 getCorrelation as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the correlation coefficient of the trend line.

3.62.9 getIntercept as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y-axis intercept of the trend line.

3.62.10 getLine as CDLineObjMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves an opaque LineObj representing the trend line. The opaque LineObj is to be used in XYChart.addInterLineLayer for adding coloring between lines.

3.62.11 getSlope as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the slope of the trend line.

3.62.12 getStdError as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the standard error of the trend line.

3.62.13 setImageMapWidth(w as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the effective width of the line used for producing image maps.

Notes: For thin lines, it is hard to click on the lines. So for the purpose of producing image maps for a line chart, ChartDirector can assume the line is very thick. The default is 10 pixels.

Argument	Default	Description
width	(Mandatory)	The effective width of the line used for producing image maps.

3.62.14 setRegressionType(regressionType as Integer)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the regression type to be used.

Notes: RegressionType:

Must be one of the constants kLinearRegression, kConstrainedLinearRegression, kExponentialRegression and kLogarithmicRegression, or the return value of BaseChartMBS.PolynomialRegression. They represent linear regression, constrained linear regression, exponential regression, logarithmic regression, and polynomial regression of configurable degree.

3.63 class CDTTFTextMBS

3.63.1 class CDTTFTextMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The TTFText class represents text blocks.

Notes: This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Real Studio Plugins, version 12.5pr8](#)

Xojo Developer Magazine

- [18.5, page 76: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)

3.63.2 Methods

3.63.3 Constructor

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

3.63.4 destroy

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Destroys the text object.

Notes: Normally you don't need to call this.

3.63.5 draw(x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other draw method, but uses color instead of integer data type for passing color values.

See also:

- [3.63.6 draw\(x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7\)](#)

3.63.6 draw(x as Integer, y as Integer, colorvalue as Integer, alignment as Integer = 7)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Draws the text block.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of a reference point to align the text.
y	(Mandatory)	The y coordinate of a reference point to align the text.
color	(Mandatory)	The color of the text.
alignment	TopLeft	The location of the text relative to the reference point. See Alignment Specification for supported alignment types.

See also:

- 3.63.5 draw(x as Integer, y as Integer, colorvalue as color, alignment as Integer = 7) 718

3.63.7 getHeight as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the height of the text block.

3.63.8 getLineDistance as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the distance between two lines in the text block.

Notes: Return Value

The distance between two lines in the text block in pixels.

3.63.9 getLineHeight as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the height of a typical line in the text block.

3.63.10 getWidth as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the width of the text block.

3.64 class CDVectorLayerMBS

3.64.1 class CDVectorLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The VectorLayer class represents vector layers.

Notes: Subclass of the CDLayerMBS class.

This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

3.64.2 Methods

3.64.3 setArrowAlignment(alignment as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the alignment of the vector relative to the data point.

Notes:

Argument	Default	Description
alignment	(Mandatory)	A BottomCenter value means the vector will point away from the data point (the default). A TopCenter value means the vector will point into the data point. A Center value means the center of the vector will be at the data point.

3.64.4 setArrowHead(polygon() as Integer)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets a custom shape to be used as the arrow head.

Notes: The custom shape is specified as an array of integers x0, y0, x1, y1, x2, y2 ... representing the coordinates of the vertices of the custom polygonal shape.

The polygon should be defined with a bounding square of 10 x 10 units, in which the x-axis is from left to right, and the y-axis from bottom to top. The origin is assumed to be the bottom center of the arrow (the point where the arrow head joins the arrow stem). The shape is assumed to represent an arrow pointing upwards.

As an example, the followings are the integer array that represents the standard ChartDirector vector arrow head:

-5, -5, 0, 0, 5, -5, 0, 5

ChartDirector will automatically scale the shape to the actual width and height as specified in `VectorLayer.setArrowHead`.

Argument	Default	Description
<code>polygon</code>	(Mandatory)	An array of integers <code>x0, y0, x1, y1, x2, y2 ...</code> representing the coordinates the polygon vertices on a 10 x 10 units grid.

See also:

- 3.64.5 `setArrowHead(width as Integer, height as Integer = 0)` 722

3.64.5 `setArrowHead(width as Integer, height as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the arrow head.

Notes:

Argument	Default	Description
<code>width</code>	(Mandatory)	The width of the arrow head in pixels. The default width is 8 pixels.
<code>height</code>	0	The height of the arrow head in pixels. The default value of 0 means the height is the same as the width.

See also:

- 3.64.4 `setArrowHead(polygon() as Integer)` 721

3.64.6 `setArrowStem(polygon() as Integer)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets a custom shape to be used as the arrow stem.

Notes: By default, the arrow stem is just a straight line, with the line width controlled using `Layer.setLineWidth`. The `setArrowStem` method can specify a custom shape for the arrow stem.

The custom shape is specified as an array of integers `x0, y0, x1, y1, x2, y2 ...` representing the coordinates of the vertices of the custom polygonal shape.

The polygon should be defined with a bounding square of 10 x 100 units, in which the x-axis is from left to right, and the y-axis from bottom to top. The origin is assumed to be the starting point of the arrow stem, and the shape is assumed to represent an arrow stem pointing upwards.

ChartDirector will automatically scale the shape so that the total arrow length (head + stem) is the required length of the arrow as according to actual data, and the stem width is as specified in `Layer.setLineWidth`.

Argument	Default	Description
<code>polygon</code>	(Mandatory)	An array of integers <code>x0, y0, x1, y1, x2, y2 ...</code> representing the coordinates the polygon vertices on a 10 x 100 units grid.

3.64.7 `setIconSize(height as Integer, width as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the size of the icon to be used in legend box.

Notes: By default, if a legend box is available on the chart, ChartDirector will insert an legend entry if the `VectorLayer` is named. The size of the icon will be the size of the vectors used on the chart, using a short vector length to fit the legend box.

This method can be used to override the legend box settings to specify a custom width/height for the icons of the current `VectorLayer`.

Argument	Default	Description
<code>height</code>	(Mandatory)	The height of the icon in pixels.
<code>width</code>	0	The width of the icon in pixels. The default value of 0 means the width is automatically determined.

3.64.8 `setVector(lengths() as Double, directions() as Double, lengthScale as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the lengths and directions for the vectors.

Notes: ChartDirector supports specifying lengths as pixels or in axis scale. The unit is specified by using the following predefined constants.

Constant	Value	Description
<code>PixelScale</code>	0	The unit is measured in pixels.
<code>XAxisScale</code>	1	The unit is measured in x-axis scale.
<code>YAxisScale</code>	2	The unit is measured in y-axis scale.

Argument	Default	Description
lengths	(Mandatory)	An array of numbers representing the lengths of the vectors, in unit as specified in the lengthScale argument.
directions	(Mandatory)	An array of numbers representing the direction of the vectors as a clockwise angle in degrees, where 0 is upward pointing direction.
lengthScale	PixelScale	The unit for the lengths, which must be one of the predefined constants in the table above.

3.64.9 setVectorMargin(startMargin as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the margins to shorten the vectors.

Notes: By default, vectors will be drawn from the given start points to the given end points.

In some cases, it may be desirable to shorten the vectors so that they do not start exactly at the start points and/or end exactly at the end points. For example, the vectors may be used to point to some circular symbols created by a ScatterLayer. The vectors may need to point to the perimeter of the circles, rather than their centers. This can be achieved by shortening the vectors by the radius of the circles.

Arguments:

Argument	Default	Description
startMargin	(Mandatory)	The length to shorten at the start of the vector in pixels.
endMargin	NoValue	The length to shorten at the end of the vector in pixels. NoValue means the length is the same as startMargin.

See also:

- 3.64.10 setVectorMargin(startMargin as Double, endMargin as Double)

724

3.64.10 setVectorMargin(startMargin as Double, endMargin as Double)

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the margins to shorten the vectors.

Notes: By default, vectors will be drawn from the given start points to the given end points.

In some cases, it may be desirable to shorten the vectors so that they do not start exactly at the start points and/or end exactly at the end points. For example, the vectors may be used to point to some circular symbols created by a ScatterLayer. The vectors may need to point to the perimeter of the circles, rather than their centers. This can be achieved by shortening the vectors by the radius of the circles.

Arguments:

Argument	Default	Description
startMargin	(Mandatory)	The length to shorten at the start of the vector in pixels.
endMargin	NoValue	The length to shorten at the end of the vector in pixels. NoValue means the length is the same as startMargin.

See also:

- 3.64.9 setVectorMargin(startMargin as Double) 724

3.65 class CDViewPortControlBaseMBS

3.65.1 class CDViewPortControlBaseMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: A viewport control is a user interface element to let the user visualize and manipulate the viewport managed by the CDViewPortManagerMBS.

Notes: In typical usage, the viewport control displays an "overall chart" that shows the full data range, and connects to the CDViewPortManagerMBS using ViewPortControlBase.setViewPortManager. It can then include a rectangle on the overall chart to represent the viewport in the CDViewPortManagerMBS. The region outside the rectangle can be dimmed out to highlight the viewport region. If the viewport is updated (such as if the user uses the mouse wheel to zoom in the chart), the rectangle will also update automatically.

To manipulate the viewport, the user can drag the rectangle to move the viewport (equivalent to scrolling). The user can also resize the viewport by dragging the border of the rectangle (equivalent to zooming), or drag a new rectangular region on the overall chart to be used as the new viewport, or click on a point on the chart to center the viewport at that point.

In ChartDirector, the ViewPortControlBase implements the main logic of the viewport control without the GUI framework specific functions. The actual GUI framework specific control can derive from ViewPortControlBase and implement the GUI specific part. This greatly simplifies development of the actual control.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.2](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.2](#)
- [MBS Xojo Plugins, version 21.2pr1](#)
- [RealTimeViewPort in ChartDirector](#)

Videos

- [RealTimeViewPort in ChartDirector](#)

3.65.2 Methods

3.65.3 Constructor

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The constructor.

3.65.4 getProperty(attr as integer) as integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Queries a property by ID.

3.65.5 handleMouseDown(x as double, y as double)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Handles the mouse down.

Notes:

Argument	Default	Description
x	(Mandatory)	The x pixel coordinate of the mouse.
y	(Mandatory)	The y pixel coordinate of the mouse.

3.65.6 handleMouseMove(x as double, y as double, isDragging as boolean)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Handles the mouse move.

Notes:

Argument	Default	Description
x	(Mandatory)	The x pixel coordinate of the mouse.
y	(Mandatory)	The y pixel coordinate of the mouse.

3.65.7 handleMouseUp(x as double, y as double)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Handles the mouse up.

Notes:

Argument	Default	Description
x	(Mandatory)	The x pixel coordinate of the mouse.
y	(Mandatory)	The y pixel coordinate of the mouse.

3.65.8 isOnPlotArea(x as double, y as double) as boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Determines if a point is on the plot area.

Notes:

Argument	Default	Description
x	(Mandatory)	The x pixel coordinate of the point.
y	(Mandatory)	The y pixel coordinate of the point.

3.65.9 paintViewPort

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Updates the viewport rectangle on the chart to reflect the state of the viewport.

3.65.10 setMouseMargin(mouseMargin as Integer, cornerMargin as Integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the margin around the viewport border for mouse drag.

Notes: The viewport border can be as thin as 1 pixel, and it is hard to position the mouse over the border to drag it. To make it easier to drag the border, dragging can start if the mouse is within a certain margin from the border.

If both vertical and horizontal zooming is allowed, the viewport corners can be dragged diagonally to resize the viewport in both directions at the same time. To distinguish between dragging the corners and the borders, if the mouse is within a certain margin from the corner when the dragging starts, it is considered to be dragging the corner.

In either case, the mouse cursor will change to let the user know if the border or corner can be dragged.

Argument	Default	Description
mouseMargin	(Mandatory)	The margin around the viewport border within which dragging of the border can start. The margin is in pixel unit. The default is 4 pixels.
cornerMargin	(Mandatory)	The margin around the viewport corner within which dragging of the corner can start. The margin is in pixel unit. The default is 8 pixels.

3.65.11 setProperty(attr as integer, value as integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets a property by ID.

3.65.12 setZoomScrollDirection(zoomDirection as integer, scrollDirection as integer)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the directions in which zooming and scrolling are allowed.

Notes: The viewport control uses this information to determine which type of dragging is allowed.

Arguments

Argument	Default	Description
zoomDirection	(Mandatory)	The allowable zoom direction. Must be one of the predefined constants kDirectionHorizontal, kDirectionVertical and kDirectionHorizontalVertical for horizontal, vertical and bi-directional zooming.
scrollDirection	(Mandatory)	The allowable scroll direction. Must be one of the predefined constants kDirectionHorizontal, kDirectionVertical and kDirectionHorizontalVertical for horizontal, vertical and bi-directional scrolling.

3.65.13 Properties**3.65.14 Chart as CDBaseChartMBS**

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The BaseChart object to be displayed in the viewport control.

Notes: (Read and Write property)

3.65.15 ClickToCenter as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Enables or disables centering the viewport at the clicked point.

Notes: True to enable the behaviour, false to disable it.

(Read and Write property)

3.65.16 CornerMargin as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The corner margin.

Notes: See setMouseMargin.

(Read and Write property)

3.65.17 Cursor as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Gets the mouse cursor to be used for user feedback.

Notes: Returns kLeft, kRight, kTop or kBottom if the mouse can initiate dragging the corresponding side of the viewport. Returns kTopLeft, kTopRight, kBottomLeft or kBottomRight if the mouse can initiate dragging the corresponding corner of the viewport. Returns kCenter if the mouse is within the viewport.

(Read only property)

3.65.18 DragBorderToResize as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Enables or disables dragging the viewport border to resize it.

Notes: True to enable the behaviour, false to disable it.

(Read and Write property)

3.65.19 DragInsideToMove as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Enables or disables dragging inside the viewport to move it.

Notes: (Read and Write property)

3.65.20 DragOutsideToSelect as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Enables or disables dragging outside the viewport to select a new viewport.

Notes: True to enable the behaviour, false to disable it.

(Read and Write property)

3.65.21 MouseMargin as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The mouse margin.

Notes: See setMouseMargin.

(Read and Write property)

3.65.22 needUpdateChart as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Checks if need to update the chart in a viewport changed event.

Notes: True if need to update the chart in a viewport changed event, otherwise false.

(Read only property)

3.65.23 needUpdateDisplay as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Checks if need to update the viewport control screen display.

Notes: True if need to update the viewport control screen display, otherwise false.

(Read only property)

3.65.24 needUpdateImageMap as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Checks if need to update the image map in a viewport changed event.

Notes: True if need to update the image map in a viewport changed event, otherwise false.

(Read only property)

3.65.25 SelectionBorderColor as Color

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the border color of the selection box.

Notes:

(Read and Write property)

Argument	Default	Description
color	(Mandatory)	The border color of the selection box. The default is semi-transparent black (&c7F000000).

3.65.26 SelectionBorderWidth as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the border width of the selection box.

Notes:

Argument	Default	Description
width	(Mandatory)	The border width of the selection box. The default is 1 pixel.

(Read and Write property)

3.65.27 ViewPortBorderColor as Color

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the viewport border color.

Notes:

Argument	Default	Description
color	(Mandatory)	The viewport border color. The default is semi-transparent black (&c7F000000).

(Read and Write property)

3.65.28 ViewPortBorderWidth as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the viewport border width.

Notes:

(Read and Write property)

Argument	Default	Description
width	(Mandatory)	The viewport border width. The default is 1 pixel.

3.65.29 ViewPortExternalColor as Color

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the fill color for the region outside the viewport.

Notes:

Argument	Default	Description
color	(Mandatory)	The fill color for the region outside the viewport. The default is semi-transparent white (&c7FFFFFFF) to dim out the region.

(Read and Write property)

3.65.30 ViewPortFillColor as Color

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets the fill color for the region inside the viewport.

Notes:

Argument	Default	Description
color	(Mandatory)	The fill color for the region inside the viewport. The default is transparent.

(Read and Write property)

3.65.31 ViewPortManager as CDViewPortManagerMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The ViewPortManager object to be associated with the viewport control.

Notes: (Read and Write property)

3.66 class CDViewPortManagerMBS

3.66.1 class CDViewPortManagerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: A view port can be imagined as a rectangular window of an underlying rectangular surface.

Notes: For example, a chart that has 10 years of data can be imagined as a very long chart. If one only displays one of the year, we can say the view port covers only 10% of the underlying chart.

With the view port concept, scrolling can be handled as moving the view port, while zooming in and out can be handled as changing the view port size.

ViewPortManager is a utility class for handling view ports. It manages mapping of the mouse and display pixel coordinates to view port coordinates, and supports various user interface constraints that limits how the view port may be changed.

View port coordinates are represented as fractions of the width or height of the underlying surface. For example, the width of a view port is represented as a fraction of the width of the underlying surface, so it must be between 0 to 1. A value of 0.1 means the view port width is 10% of the underlying surface width.

If you are using MFC, there is no need to create a ViewPortManager object directly. You may simply use the MFC CChartViewer control, which is a derived class of ViewPortManager and contains all its functions.

If you are using other GUI framework, and would like to perform zooming and scrolling functions by mouse drag, the ViewPortManager may be useful to you. The section Using ChartDirector with Other GUI Frameworks contains outlines on how the ViewPortManager may be used in general GUI frameworks.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.2](#)
- [MBS Xojo Plugins, version 21.2pr1](#)

3.66.2 Methods

3.66.3 canZoomIn(zoomDirection as Integer) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if it is possible to zoom in further at a certain direction without violating zoom in limits.

Notes: See ViewPortManager.setZoomInWidthLimit and ViewPortManager.setZoomInHeightLimit on how to configure zoom in limits.

Argument	Default	Description
zoomDirection	(Mandatory)	The zoom direction to check. Must be one of the predefined constants DirectionHorizontal, DirectionVertical and DirectionHorizontalVertical for horizontal, vertical and bi-directional zooming.

Return Value

True if can zoom in further, otherwise false.

3.66.4 canZoomOut(zoomDirection as Integer) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if it is possible to zoom out further at a certain direction without violating zoom out limits.

Notes: See ViewPortManager.setZoomOutWidthLimit and ViewPortManager.setZoomOutHeightLimit on how to configure zoom out limits.

Argument	Default	Description
zoomDirection	(Mandatory)	The zoom direction to check. Must be one of the predefined constants DirectionHorizontal, DirectionVertical and DirectionHorizontalVertical for horizontal, vertical and bi-directional zooming.

Return Value

True if can zoom out further, otherwise false.

3.66.5 clearAllRanges

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A convenience method to clear all the ranges configured using setFullRange.

3.66.6 commitPendingSyncAxis(baseChart as CDBaseChartMBS)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The commitPendingSyncAxis is a method to actually perform the function in syncLinearAxisWithViewPort, syncLogAxisWithViewPort and syncDateAxisWithViewPort.

3.66.7 Constructor

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new viewport.

3.66.8 dragTo(scrollDirection as integer, x as double, y as double) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Scrolls the view port to reflect dragging of the underlying rectangular surface.

Notes: The amount of drag is measured as changed in mouse cursor coordinates since the call to ViewPortManager.startDrag. A positive change means the dragging is to the right or bottom. A negative change means the dragging is to the left or top.

The drag is considered as applying to the underlying surface. The view port moves in the opposite direction to the drag. For example, dragging the underlying surface to the right is equivalent to moving the view port to the left.

The view port may not change at all if it has reached the borders of the underlying surface.

Argument	Default	Description
scrollDirection	(Mandatory)	The allowed scroll direction. Must be one of the predefined constants DirectionHorizontal, DirectionVertical and DirectionHorizontalVertical for horizontal, vertical and bi-directional scrolling.
deltaX	(Mandatory)	The change in mouse x-coordinates. A positive change means the drag is to the right. A negative change means the drag is to the left.
deltaY	(Mandatory)	The change in mouse y-coordinates. A positive change means the drag is to the bottom. A negative change means the drag is to the top.

3.66.9 getPlotAreaHeight as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the height of the plot area in pixels.

Notes: Return Value

The height of the plot area in pixels.

3.66.10 getPlotAreaLeft as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x-coordinate of the left side of the plot area in pixels.

Notes: Return Value

The x-coordinate of the left side of the plot area in pixels.

3.66.11 getPlotAreaTop as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y-coordinate of the top side of the plot area in pixels.

3.66.12 getPlotAreaWidth as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the width of the plot area in pixels.

3.66.13 getValueAtViewPort(id as string, ratio as Double, isLogScale as boolean = false) as Double

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts a view port coordinate to a value of the specified data scale.

Notes: Please refer to ViewPortManager.setFullRange on how to define a data scale.

Argument	Default	Description
id	(Mandatory)	The name of the data scale.
vpCoor	(Mandatory)	The view port coordinate.
isLogScale	false	true if the conversion is based on a logarithmic scale. false if the conversion is based on a non-logarithmic scale.

Returns the value of the specified data scale at the view port coordinate.

3.66.14 getViewPortAtValue(id as string, ratio as Double, isLogScale as boolean = false) as Double

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Converts a value of the specified data scale to a view port coordinate.

Notes: Please refer to ViewPortManager.setFullRange on how to define a data scale.

Argument	Default	Description
id	(Mandatory)	The name of the data scale.
value	(Mandatory)	The value to be converted.
isLogScale	false	true if the conversion is based on a logarithmic scale. false if the conversion is based on a non-logarithmic scale.

Returns the view port coordinate at the value of the specified data scale.

3.66.15 getViewPortHeight as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the height of the view port.

Notes: The view port height is expressed as a fraction of the height of the underlying surface. It should be between 0 and 1.

3.66.16 getViewPortLeft as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the position of the left side of the view port.

Notes: The position of the view port left side is its distance from the left side of the underlying surface, as a fraction of the width of the underlying surface. It should be between 0 and 1.

3.66.17 getViewPortTop as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the position of the top side of the view port.

Notes: The position of the view port top side is its distance from the top side of the underlying surface, as a fraction of the height of the underlying surface. It should be between 0 and 1.

3.66.18 getViewPortWidth as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the width of the view port.

Notes: The view port width is expressed as a fraction of the width of the underlying surface. It should be between 0 and 1.

3.66.19 getZoomInHeightLimit as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the view port height at maximum zoom in for mouse zoom in actions.

3.66.20 getZoomInWidthLimit as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the view port width at maximum zoom in for mouse zoom in actions.

3.66.21 getZoomOutHeightLimit as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the view port height at maximum zoom out for mouse zoom out actions.

3.66.22 getZoomOutWidthLimit as Double

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the view port width at maximum zoom out for mouse zoom out actions.

3.66.23 getZoomXYRatio as double

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the enforced aspect ratio of the viewport.

Notes: The ViewPortManager.setKeepAspectRatio can be used to keep the aspect ratio unchanged. This method gets that aspect ratio.

Returns the enforced aspect ratio, or 0 if the aspect ratio is not enforced.

3.66.24 inExtendedPlotArea(x as Integer, y as Integer) as boolean

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Determines if a given (x, y) coordinate is within the extended plot area (the plot area plus the extra margin sets up using `setPlotAreaMouseMargin`).

3.66.25 `inPlotArea(x as double, y as double)` as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Determines if a given point is within the plot area.

Notes:

Argument	Default	Description
x	(Mandatory)	The x coordinate of the point in pixel unit.
y	(Mandatory)	The y coordinate of the point in pixel unit.

Return Value

True if the point is within the plot area, otherwise false.

3.66.26 `setChartMetrics(metrics as string)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the chart metrics to the `ViewPortManager` so it knows the positions of the necessary chart objects for supporting view ports.

Notes:

Argument	Default	Description
metrics	(Mandatory)	A text string obtained from <code>BaseChart.getChartMetrics</code> that represents the metrics of the chart.

3.66.27 `setFullRange(ID as string, minValue as Double, maxValue as Double)`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Defines the full range of a view port data scale.

Notes: In the `ChartDirector` zooming and scrolling framework, the view port coordinates are expressed as the visible fractions of the "full data scale". For example, if the "full data scale" is 10 years horizontally, a view port width of 0.1 means that 1 year of data are visible.

The `setFullRange` method defines the full range of a data scale and gives it a name. As `ChartDirector` supports multiple x-axes and y-axes, so there can be multiple data scales with different names. The name can be used in other `ChartDirector` APIs to convert between the view port coordinates and data scale (see

ViewPortManager.getValueAtViewPort, ViewPortManager.getViewPortAtValue), and to configure an Axis to reflect the visible data scale (see ViewPortManager.syncLinearAxisWithViewPort, ViewPortManager.syncLogAxisWithViewPort and ViewPortManager.syncDateAxisWithViewPort).

Argument	Default	Description
id	(Mandatory)	The name of the data scale.
minValue	(Mandatory)	The minimum value of the data scale.
maxValue	(Mandatory)	The maximum value of the data scale.

3.66.28 setKeepAspectRatio(value as boolean)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Keeps the current viewport aspect ratio.

Notes: If the viewport aspect ratio is keep unchanged, ViewPortManager will ensure the vertical and horizontal directions zoom by the same ratio.

Argument	Default	Description
b	(Mandatory)	A true value ensures the vertical and horizontal directions zoom by the same ratio. A false value allows them to zoom independently.

3.66.29 setPlotAreaMouseMargin(leftMargin as Integer, rightMargin as Integer, topMargin as Integer, bottomMargin as Integer)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Configures area for mouse tracking.

Notes: SetPlotAreaMouseMargin configures some margins outside the plot area, so that a mouse cursor in that region (and therefore just outside the plot area) will still be considered to be exactly at the boundary of the plot area for the purpose of triggering plot area mouse events. Without this mechanism, it would be difficult to put the mouse exactly at the edge of the plot area (as it is too easy to "overshoot" the edge). Putting the mouse exactly at the edge is very useful for "track cursors" as this tracks the first or last data point.

3.66.30 setViewPortHeight(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the height of the view port.

Notes: The view port height is expressed as a fraction of the height of the underlying surface. It should be

between 0 and 1.

Argument	Default	Description
height	(Mandatory)	The height of the view port.

3.66.31 setViewPortLeft(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position of the left side of the view port.

Notes: The position of the view port left side is its distance from the left side of the underlying surface, as a fraction of the width of the underlying surface. It should be between 0 and 1.

Argument	Default	Description
left	(Mandatory)	The position of the left side of the view port.

3.66.32 setViewPortTop(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position of the top side of the view port.

Notes: The position of the view port top side is its distance from the top side of the underlying surface, as a fraction of the height of the underlying surface. It should be between 0 and 1.

Argument	Default	Description
top	(Mandatory)	The position of the top side of the view port.

3.66.33 setViewPortWidth(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the width of the view port.

Notes: The view port width is expressed as a fraction of the width of the underlying surface. It should be between 0 and 1.

3.66.34 setZoomInHeightLimit(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Argument	Default	Description
width	(Mandatory)	The width of the view port.

Function: Sets the view port height at maximum zoom in.

Notes: In many applications, it is desirable to set a maximum zoom in level, instead of allowing the user to zoom in indefinitely.

This method determines the minimum allowed view port height. It should be between 0 and 1. The default is 0.01, which means a maximum zoom in of 100x (the view port see only 1% of the underlying surface).

Argument	Default	Description
height	(Mandatory)	The minimum allowed view port height.

3.66.35 setZoomInWidthLimit(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the view port width at maximum zoom in.

Notes: In many applications, it is desirable to set a maximum zoom in level, instead of allowing the user to zoom in indefinitely.

This method determines the minimum allowed view port width. It should be between 0 and 1. The default is 0.01, which means a maximum zoom in of 100x (the view port see only 1% of the underlying surface).

Argument	Default	Description
width	(Mandatory)	The minimum allowed view port width.

3.66.36 setZoomOutHeightLimit(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the view port height at maximum zoom in.

Notes: In many applications, it is desirable to set a maximum zoom in level, instead of allowing the user to zoom in indefinitely.

This method determines the minimum allowed view port height. It should be between 0 and 1. The default is 0.01, which means a maximum zoom in of 100x (the view port see only 1% of the underlying surface).

Argument	Default	Description
height	(Mandatory)	The minimum allowed view port height.

3.66.37 setZoomOutWidthLimit(value as Double)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the view port width at maximum zoom out for mouse zoom out actions.

3.66.38 startDrag

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Takes a snapshot of the view port to prepare for dragging.

Notes: This method must be called before calling ViewPortManager.dragTo.

3.66.39 syncDateAxisWithViewPort(id as string, axis as CDAxisMBS)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Synchronizes a date/time Axis with the part of the data scale in view port.

Notes: If the data scale has already been defined (see ViewPortManager.setFullRange on how to define a data scale), this method will compute the visible data scale based on the view port coordinates using date/time interpolation, and then sets the axis to that date/time scale.

If the data scale has not been defined, this method will define the data scale based on the axis scale and the the view port coordinates using date/time extrapolation. Because the axis scale may not be known at the time of calling this method (the axis could be auto-scaled by ChartDirector, which may not occur until the chart is rendered), the definition of the data scale will not occur immediately, but will be pending until the chart is to be displayed.

If your charting code calls this method but never defines the full data scale or the view port coordinates (in which case the view port defaults to showing the complete data scale), then the first time the code is executed, the data scale will be defined to be equal to the axis scale. Subsequently, when the user zooms into the chart (which means the view port coordinates are changed), the same charting code will set up the axis scale to the range that the user has zoomed to.

Argument	Default	Description
id	(Mandatory)	The name of the data scale.
axis	(Mandatory)	The Axis object to synchronize with.

3.66.40 syncLinearAxisWithViewPort(id as string, axis as CDAxisMBS)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Synchronizes a linear Axis with the part of the data scale in view port.

Notes: If the data scale has already been defined (see ViewPortManager.setFullRange on how to define a data scale), this method will compute the visible data scale based on the view port coordinates using linear interpolation, and then sets the axis to that linear scale.

If the data scale has not been defined, this method will define the data scale based on the axis scale and the view port coordinates using linear extrapolation. Because the axis scale may not be known at the time of calling this method (the axis could be auto-scaled by ChartDirector, which may not occur until the chart is rendered), the definition of the data scale will not occur immediately, but will be pending until the chart is to be displayed.

If your charting code calls this method but never defines the full data scale or the view port coordinates (in which case the view port defaults to showing the complete data scale), then the first time the code is executed, the data scale will be defined to be equal to the axis scale. Subsequently, when the user zooms into the chart (which means the view port coordinates are changed), the same charting code will set up the axis scale to the range that the user has zoomed to.

Argument	Default	Description
id	(Mandatory)	The name of the data scale.
axis	(Mandatory)	The Axis object to synchronize with.

3.66.41 syncLogAxisWithViewPort(id as string, axis as CDAxisMBS)

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Synchronizes a logarithmic Axis with the part of the data scale in view port.

Notes: If the data scale has already been defined (see ViewPortManager.setFullRange on how to define a data scale), this method will compute the visible data scale based on the view port coordinates using logarithmic interpolation, and then sets the axis to that logarithmic scale.

If the data scale has not been defined, this method will define the data scale based on the axis scale and the view port coordinates using logarithmic extrapolation. Because the axis scale may not be known at the time of calling this method (the axis could be auto-scaled by ChartDirector, which may not occur until the chart is rendered), the definition of the data scale will not occur immediately, but will be pending until the chart is to be displayed.

If your charting code calls this method but never defines the full data scale or the view port coordinates (in which case the view port defaults to showing the complete data scale), then the first time the code is executed, the data scale will be defined to be equal to the axis scale. Subsequently, when the user zooms

into the chart (which means the view port coordinates are changed), the same charting code will set up the axis scale to the range that the user has zoomed to.

Argument	Default	Description
id	(Mandatory)	The name of the data scale.
axis	(Mandatory)	The Axis object to synchronize with.

3.66.42 `updateFullRangeH(id as string, minValue as Double, maxValue as Double, updateType as Integer) as boolean`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Updates the full range of a horizontal viewport data scale.

Notes:

Argument	Default	Description
id	(Mandatory)	The name of the horizontal data scale.
minValue	(Mandatory)	The new minimum value of the data scale.
maxValue	(Mandatory)	The new maximum value of the data scale.
updateType	(Mandatory)	The method to update the viewport. Must be one of ViewPortNoUpdate, KeepVisibleRange, ScrollWithMax or ScrollWithMin.

3.66.43 `updateFullRangeV(id as string, minValue as Double, maxValue as Double, updateType as Integer) as boolean`

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Updates the full range of a vertical viewport data scale.

Notes:

Argument	Default	Description
id	(Mandatory)	The name of the vertical data scale.
minValue	(Mandatory)	The new minimum value of the data scale.
maxValue	(Mandatory)	The new maximum value of the data scale.
updateType	(Mandatory)	The method to update the viewport. Must be one of ViewPortNoUpdate, KeepVisibleRange, ScrollWithMax or ScrollWithMin.

3.66.44 `validateViewPort`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Ensures the view port left, top, width and height are within valid ranges and adjusts them if necessary.

Notes: The valid ranges of the view port width and height should be 0 to 1. The view port left should be in between 0 and (1 - view port width). The view port top should be in between 0 and (1 - view port height).

3.66.45 zoomAround(x as double, y as double, xZoomRatio as double, yZoomRatio as double) as boolean

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Zoom at a given point using the zoom ratio.

3.66.46 zoomAt(zoomDirection as integer, x as double, y as double, zoomRatio as double) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Zooms in/out around a certain point.

Notes: This method adjusts the view port position and size to reflect zooming in/out around the given point. If possible, the given point will be at the center of the view port after zooming in/out.

Argument	Default	Description
zoomDirection	(Mandatory)	The allowed zoom direction. Must be one of the predefined constants DirectionHorizontal, DirectionVertical and DirectionHorizontalVertical for horizontal, vertical and bi-directional zooming.
x	(Mandatory)	The x-coordinate of the point to zoom around.
y	(Mandatory)	The y-coordinate of the point to zoom around.
zoomRatio	(Mandatory)	The zoom magnification factor. For example, a value of 2 means zooming in by 2x. A value of 0.5 means zooming out by 2x.

Return Value

True if view port is changed, otherwise false. It is possible for no zooming to occur if zoom limits are reached. See ViewPortManager.setZoomInWidthLimit, ViewPortManager.setZoomOutWidthLimit, ViewPortManager.setZoomInHeightLimit and ViewPortManager.setZoomOutHeightLimit for the meaning of zoom limits.

3.66.47 zoomTo(zoomDirection as integer, x1 as double, y1 as double, x2 as double, y2 as double) as boolean

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Zooms to the selected rectangular region.

Notes: This method updates the view port position and size to reflect a rectangular region on the chart. (x1, y1) and (x2, y2) are opposite vertices of the rectangular region in pixel coordinates.

The final view port position may not exactly reflect the rectangular region because of zoom limits. See `ViewPortManager.setZoomInWidthLimit`, `ViewPortManager.setZoomOutWidthLimit`, `ViewPortManager.setZoomInHeightLimit` and `ViewPortManager.setZoomOutHeightLimit` for the meaning of zoom limits.

Argument	Default	Description
<code>zoomDirection</code>	(Mandatory)	The allowed zoom direction. Must be one of the predefined constants <code>DirectionHorizontal</code> , <code>DirectionVertical</code> and <code>DirectionHorizontalVertical</code> for horizontal, vertical and bi-directional zooming.
<code>x1</code>	(Mandatory)	The x-coordinate of one vertices of the selected rectangular region.
<code>y1</code>	(Mandatory)	The y-coordinate of one vertices of the selected rectangular region.
<code>x2</code>	(Mandatory)	The x-coordinate of the vertex that is opposite to the vertex (x1, y1).
<code>y2</code>	(Mandatory)	The y-coordinate of the vertex that is opposite to the vertex (x1, y1).

Return Value

True if view port is changed, otherwise false.

3.67 class CDXYChartMBS

3.67.1 class CDXYChartMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The XYChart class represents XY charts.

Notes: Subclass of the CDBaseChartMBS class.

Blog Entries

- [Charts with more than one x or y axis](#)
- [Create a Bar Chart in Xojo with the ChartDirector](#)
- [Several ways for picture to PDF in MBS Plugins](#)
- [Chart Diagrams with Xojo](#)
- [ChartDirector 6 PDF example](#)
- [MBS Real Studio Plugins, version 12.5pr8](#)
- [A special chart with our MBS REALbasic ChartDirector Plugin](#)

Xojo Developer Magazine

- [7.4, pages 31 to 34: Easy Charts and Graphs, Using the ChartDirector Plugin](#)
- [20.1, page 43: Cool Charts and Heatmaps, Using Monkeybread Software's ChartDirector Plugin by Stefanie Juchmes](#)
- [18.5, page 69: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)
- [18.5, pages 61 to 62: Xojo Charts, Charts in Xojo with the MBS Xojo ChartDirector plugin by Stefanie Juchmes](#)
- [17.2, page 35: More Beyond JSON, Develop APIs That Generate Barcodes, Charts, and More by Timothy Dietrich](#)
- [17.2, page 33: More Beyond JSON, Develop APIs That Generate Barcodes, Charts, and More by Timothy Dietrich](#)

3.67.2 Methods

3.67.3 `addAreaLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addAreaLayer` method, but uses `color` instead of `integer` data type for passing color values.

See also:

- 3.67.4 `addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS` 750
- 3.67.5 `addAreaLayer(dataCombineMethod as Integer = 1, depth as Integer = 0) as CDAreaLayerMBS` 751
- 3.67.6 `addAreaLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS` 752
- 3.67.7 `addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS` 752
- 3.67.8 `addAreaLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDAreaLayerMBS` 753
- 3.67.9 `addAreaLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDAreaLayerMBS` 754

3.67.4 `addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an area chart layer to the `XYChart`, and specify the data set to use for drawing the area.

Notes:

Parameter	Default	Description
<code>data</code>	(Mandatory)	An array of numbers representing the data set.
<code>color</code>	-1	The color to draw the area. -1 means that the color is automatically selected from the color palette.
<code>name</code>	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
<code>depth</code>	0	The 3D depth of the area layer.

Return Value

An `AreaLayer` object representing the area layer created.

See also:

- 3.67.3 `addAreaLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS` 749
- 3.67.5 `addAreaLayer(dataCombineMethod as Integer = 1, depth as Integer = 0) as CDAreaLayerMBS` 751

- 3.67. CLASS CDXYCHARTMBS 751
- 3.67.6 addAreaLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 752
 - 3.67.7 addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 752
 - 3.67.8 addAreaLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDAreaLayerMBS 753
 - 3.67.9 addAreaLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDAreaLayerMBS 754

3.67.5 addAreaLayer(dataCombineMethod as Integer = 1, depth as Integer = 0) as CDAreaLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an empty area chart layer to the XYChart.

Notes: This method is typically used to add multiple data sets to a single bar layer. First an empty area chart layer is created, then the data sets can be added using Layer.addDataSet.

The dataCombineMethod parameter specifies how to combine the data sets together in the area layer. The followings methods are supported:

Constant	Value	Description
Stack	1	The data sets are combined by stacking up the areas.
Percentage	4	The data sets are combined similar to stacked area, except that the data are scaled so that the area always stacked up to 100. An area strip within the stacked area therefore represents the percentage of the data item relative to sum of all the data items in the stacked area.

Parameter	Default	Description
dataCombineMethod	Stack	The method to combine the data sets together in the area layer.
depth	0	The 3D depth of the area layer.

Return Value

An AreaLayer object representing the area layer created.

See also:

- 3.67.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 749
- 3.67.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 750

- 3.67.6 `addAreaLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0)` as `CDAreaLayerMBS` 752
- 3.67.7 `addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0)` as `CDAreaLayerMBS` 752
- 3.67.8 `addAreaLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0)` as `CDAreaLayerMBS` 753
- 3.67.9 `addAreaLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0)` as `CDAreaLayerMBS` 754

3.67.6 `addAreaLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0)` as `CDAreaLayerMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Same as the other `addAreaLayer` method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.3 `addAreaLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0)` as `CDAreaLayerMBS` 749
- 3.67.4 `addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0)` as `CDAreaLayerMBS` 750
- 3.67.5 `addAreaLayer(dataCombineMethod as Integer = 1, depth as Integer = 0)` as `CDAreaLayerMBS` 751
- 3.67.7 `addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0)` as `CDAreaLayerMBS` 752
- 3.67.8 `addAreaLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0)` as `CDAreaLayerMBS` 753
- 3.67.9 `addAreaLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0)` as `CDAreaLayerMBS` 754

3.67.7 `addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0)` as `CDAreaLayerMBS`

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds an area chart layer to the `XYChart`, and specify the data set to use for drawing the area.

Notes:

Return Value

An `AreaLayer` object representing the area layer created.

See also:

Parameter	Default	Description
data	(Mandatory)	An array of numbers representing the data set.
color	-1	The color to draw the area. -1 means that the color is automatically selected from the color palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
depth	0	The 3D depth of the area layer.

- 3.67.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 749
- 3.67.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 750
- 3.67.5 addAreaLayer(dataCombineMethod as Integer = 1, depth as Integer = 0) as CDAreaLayerMBS 751
- 3.67.6 addAreaLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 752
- 3.67.8 addAreaLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDAreaLayerMBS 753
- 3.67.9 addAreaLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDAreaLayerMBS 754

3.67.8 addAreaLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDAreaLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an area chart layer to the XYChart, and specify the data set to use for drawing the area.
See also:

- 3.67.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 749
- 3.67.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 750
- 3.67.5 addAreaLayer(dataCombineMethod as Integer = 1, depth as Integer = 0) as CDAreaLayerMBS 751
- 3.67.6 addAreaLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 752
- 3.67.7 addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 752
- 3.67.9 addAreaLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDAreaLayerMBS 754

3.67.9 addAreaLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDAreaLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an area chart layer to the XYChart, and specify the data set to use for drawing the area.
See also:

- 3.67.3 addAreaLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 749
- 3.67.4 addAreaLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 750
- 3.67.5 addAreaLayer(dataCombineMethod as Integer = 1, depth as Integer = 0) as CDAreaLayerMBS 751
- 3.67.6 addAreaLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 752
- 3.67.7 addAreaLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDAreaLayerMBS 752
- 3.67.8 addAreaLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDAreaLayerMBS 753

3.67.10 addAxis(align as Integer, offset as Integer) as CDAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an additional axis to the chart.

Notes: By default, ChartDirector XY charts have 2 x-axes and 2 y-axes at the 4 borders of the plot area. The addAxis method can be used to create additional axis.

The new axis will be put at one of the 4 borders of the plot area. To avoid overlapping with existing axis, the new axis will not be put exactly as the plot area border, but at an offset from it.

Parameter	Default	Description
align	(Mandatory)	Specifies which side of the plot area is the primary side of the axis. Must be one of the constants Left, Right, Top and Bottom for the 4 sides of the plot area.
offset	(Mandatory)	An offset in pixels to move the axis away from the plot area. If a negative value is used, it will mean to move the axis into the plot area.

3.67.11 addBarLayer(data() as Double, colors() as color, depth as Integer = 0) as CDBarLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBarLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.12 addBarLayer(data() as Double, colors() as color, names() as string, depth as Integer = 0) as CDBarLayerMBS 755
- 3.67.13 addBarLayer(data() as Double, colors() as Integer, depth as Integer = 0) as CDBarLayerMBS 756
- 3.67.14 addBarLayer(data() as Double, colors() as Integer, names() as string, depth as Integer = 0) as CDBarLayerMBS 757
- 3.67.15 addBarLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDBarLayerMBS 758
- 3.67.16 addBarLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDBarLayerMBS 759
- 3.67.17 addBarLayer(dataCombineMethod as Integer = 3, depth as Integer = 0) as CDBarLayerMBS 759

3.67.12 addBarLayer(data() as Double, colors() as color, names() as string, depth as Integer = 0) as CDBarLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBarLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.11 addBarLayer(data() as Double, colors() as color, depth as Integer = 0) as CDBarLayerMBS 755
- 3.67.13 addBarLayer(data() as Double, colors() as Integer, depth as Integer = 0) as CDBarLayerMBS 756
- 3.67.14 addBarLayer(data() as Double, colors() as Integer, names() as string, depth as Integer = 0) as CDBarLayerMBS 757
- 3.67.15 addBarLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDBarLayerMBS 758
- 3.67.16 addBarLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDBarLayerMBS 759

- 3.67.17 `addBarLayer(dataCombineMethod as Integer = 3, depth as Integer = 0) as CDBarLayerMBS`
759

3.67.13 `addBarLayer(data() as Double, colors() as Integer, depth as Integer = 0) as CDBarLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an empty bar layer to the XYChart.

Example:

```
// Create a XYChart object of size 250 x 250 pixels
dim c as new CDXYChartMBS(250, 250)

// Add a bar chart layer using the given data
dim t as CDTextBoxMBS = c.addBarLayer(data)
t.setPos(0, -10) // move 10 up
```

Notes: This method is typically used to add multiple data sets to a single bar layer. First an empty bar layer is created, then the data sets can be added using `Layer.addDataSet`.

The `dataCombineMethod` parameter specifies how to combine the data sets together in the bar layer. The following methods are supported:

Constant	Value	Description
Side	3	The data sets are combined by plotting the bars side by side.
Stack	1	The data sets are combined by stacking up the bar segments.
Overlay	0	The data sets are combined similar to stacked bars. However, in Overlay, one data set is assumed to already include the other data set. For example, if the data sets are "average" and "peak", the "peak" cannot be stacked on top of "average", because the "peak" already contains "average". In the Overlay style, only "peak - average" is stacked on top of "average", and so the total bar length will be "peak".
Percentage	4	The data sets are combined similar to stacked bars, except that the data in a bar are scaled so that they summed to 100. In other words, all stacked bars will be of the same length. A bar segment within a bar represents the percentage of the data item relative to sum of all the data items in the stacked bar.

Parameter	Default	Description
<code>dataCombineMethod</code>	Side	The method to combine the data sets together in the bar layer.
<code>depth</code>	0	The 3D depth of the bar layer.

Return Value

A BarLayer object representing the bar layer created.

To set some more options like transparent border of the bar, please use the methods in CDBarLayerMBS class. e.g. `setBorderColor(CDXYChartMBS.kTransparent)` will make them transparent.

See also:

- 3.67.11 `addBarLayer(data() as Double, colors() as color, depth as Integer = 0) as CDBarLayerMBS`
755
- 3.67.12 `addBarLayer(data() as Double, colors() as color, names() as string, depth as Integer = 0) as CDBarLayerMBS`
755
- 3.67.14 `addBarLayer(data() as Double, colors() as Integer, names() as string, depth as Integer = 0) as CDBarLayerMBS`
757
- 3.67.15 `addBarLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDBarLayerMBS`
758
- 3.67.16 `addBarLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDBarLayerMBS`
759
- 3.67.17 `addBarLayer(dataCombineMethod as Integer = 3, depth as Integer = 0) as CDBarLayerMBS`
759

3.67.14 `addBarLayer(data() as Double, colors() as Integer, names() as string, depth as Integer = 0) as CDBarLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-color bar layer to the XYChart, and specify the data set to use for drawing the bars.

Notes: A multi-color bar layer is a bar layer in which each bar has a different color. In a normal bar layer, each data set has a different color, but the bars in the same data set have the same color.

Parameter	Default	Description
<code>data</code>	(Mandatory)	An array of numbers representing the data set.
<code>colors</code>	(Mandatory)	An array of colors to draw the bars. An empty array means the colors are automatically selected from the palette.
<code>names</code>	[Empty_Array]	An array of text strings as the names of the bars. The names will be used in the legend box, if one is available. An empty array means that bars have no name.
<code>depth</code>	0	The 3D depth of the bar layer.

Return Value

A BarLayer object representing the bar layer created.

To set some more options like transparent border of the bar, please use the methods in `CDBarLayerMBS` class. e.g. `setBorderColor(CDXYChartMBS.kTransparent)` will make them transparent.

See also:

- 3.67.11 `addBarLayer(data() as Double, colors() as color, depth as Integer = 0)` as `CDBarLayerMBS`
755
- 3.67.12 `addBarLayer(data() as Double, colors() as color, names() as string, depth as Integer = 0)` as
`CDBarLayerMBS` 755
- 3.67.13 `addBarLayer(data() as Double, colors() as Integer, depth as Integer = 0)` as `CDBarLayerMBS`
756
- 3.67.15 `addBarLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0)`
as `CDBarLayerMBS` 758
- 3.67.16 `addBarLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as
Integer = 0)` as `CDBarLayerMBS` 759
- 3.67.17 `addBarLayer(dataCombineMethod as Integer = 3, depth as Integer = 0)` as `CDBarLayerMBS`
759

3.67.15 `addBarLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0)` as `CDBarLayerMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addBarLayer` method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.11 `addBarLayer(data() as Double, colors() as color, depth as Integer = 0)` as `CDBarLayerMBS`
755
- 3.67.12 `addBarLayer(data() as Double, colors() as color, names() as string, depth as Integer = 0)` as
`CDBarLayerMBS` 755
- 3.67.13 `addBarLayer(data() as Double, colors() as Integer, depth as Integer = 0)` as `CDBarLayerMBS`
756
- 3.67.14 `addBarLayer(data() as Double, colors() as Integer, names() as string, depth as Integer = 0)` as
`CDBarLayerMBS` 757
- 3.67.16 `addBarLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as
Integer = 0)` as `CDBarLayerMBS` 759
- 3.67.17 `addBarLayer(dataCombineMethod as Integer = 3, depth as Integer = 0)` as `CDBarLayerMBS`
759

3.67.16 addBarLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDBarLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a bar layer to the XYChart, and specify the data set to use for drawing the bars.

Notes:

Parameter	Default	Description
data	(Mandatory)	An array of numbers representing the data set.
color	-1	The color to draw the bars. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
depth	0	The 3D depth of the bar layer.

Return Value

A BarLayer object representing the bar layer created.

To set some more options like transparent border of the bar, please use the methods in CDBarLayerMBS class. e.g. setBorderColor(CDXYChartMBS.kTransparent) will make them transparent.

See also:

- 3.67.11 addBarLayer(data() as Double, colors() as color, depth as Integer = 0) as CDBarLayerMBS
755
- 3.67.12 addBarLayer(data() as Double, colors() as color, names() as string, depth as Integer = 0) as CDBarLayerMBS
755
- 3.67.13 addBarLayer(data() as Double, colors() as Integer, depth as Integer = 0) as CDBarLayerMBS
756
- 3.67.14 addBarLayer(data() as Double, colors() as Integer, names() as string, depth as Integer = 0) as CDBarLayerMBS
757
- 3.67.15 addBarLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDBarLayerMBS
758
- 3.67.17 addBarLayer(dataCombineMethod as Integer = 3, depth as Integer = 0) as CDBarLayerMBS
759

3.67.17 addBarLayer(dataCombineMethod as Integer = 3, depth as Integer = 0) as CDBarLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an empty bar layer to the XYChart.

Notes: This method is typically used to add multiple data sets to a single bar layer. First an empty bar

layer is created, then the data sets can be added using `Layer.addDataSet`.

The `dataCombineMethod` parameter specifies how to combine the data sets together in the bar layer. The followings methods are supported:

Constant	Value	Description
Side	3	The data sets are combined by plotting the bars side by side.
Stack	1	The data sets are combined by stacking up the bar segments.
Overlay	0	The data sets are combined similar to stacked bars. However, in Overlay, one data set is assumed to already include the other data set. For example, if the data sets are "average" and "peak", the "peak" cannot be stacked on top of "average", because the "peak" already contains "average". In the Overlay style, only "peak - average" is stacked on top of "average", and so the total bar length will be "peak".
Percentage	4	The data sets are combined similar to stacked bars, except that the data in a bar are scaled so that they summed to 100. In other words, all stacked bars will be of the same length. A bar segment within a bar represents the percentage of the data item relative to sum of all the data items in the stacked bar.

Parameter	Default	Description
<code>dataCombineMethod</code>	Side	The method to combine the data sets together in the bar layer.
<code>depth</code>	0	The 3D depth of the bar layer.

Return Value

A `BarLayer` object representing the bar layer created.

To set some more options like transparent border of the bar, please use the methods in `CDBarLayerMBS` class. e.g. `setBorderColor(CDXYChartMBS.kTransparent)` will make them transparent.

See also:

- 3.67.11 `addBarLayer(data() as Double, colors() as color, depth as Integer = 0) as CDBarLayerMBS`
755
- 3.67.12 `addBarLayer(data() as Double, colors() as color, names() as string, depth as Integer = 0) as CDBarLayerMBS`
755
- 3.67.13 `addBarLayer(data() as Double, colors() as Integer, depth as Integer = 0) as CDBarLayerMBS`
756
- 3.67.14 `addBarLayer(data() as Double, colors() as Integer, names() as string, depth as Integer = 0) as CDBarLayerMBS`
757
- 3.67.15 `addBarLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDBarLayerMBS`
758

- 3.67.16 addBarLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDBarLayerMBS 759

3.67.18 addBoxLayer(boxTop() as Double, boxBottom() as Double, colorvalue as color, name as string = "") as CDBoxWhiskerLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBoxLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.19 addBoxLayer(boxTop() as Double, boxBottom() as Double, colorvalue as Integer = -1, name as string = "") as CDBoxWhiskerLayerMBS 761

3.67.19 addBoxLayer(boxTop() as Double, boxBottom() as Double, colorvalue as Integer = -1, name as string = "") as CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a floating box layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: This method is a simplification of XYChart.addBoxWhiskerLayer. Instead of adding a full box-whisker layer, only the box part is used, resulting in a chart layer displaying floating boxes.

Parameter	Default	Description
boxTop	(Mandatory)	An array of numbers representing the top edge of the box.
boxBottom	(Mandatory)	An array of numbers representing the bottom edge of the box.
color	-1	The color to draw the area. -1 means that the color is automatically selected from the color palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.

Return Value

A BoxWhiskerLayer object representing the box-whisker layer created.

See also:

- 3.67.18 addBoxLayer(boxTop() as Double, boxBottom() as Double, colorvalue as color, name as string = "") as CDBoxWhiskerLayerMBS 761

3.67.20 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: Traditionally, each box-whisker symbol represents 5 values, which are the as maximum, 3rd quartile, median, 1st quartile and minimum of some data samples.

A vertical box-whisker symbol (on a non-swapped XY chart) consists of a vertical line joining the maximum and minimum points, a box extending from the 1st quartile point to the 3rd quartile point, and 3 horizontal mark lines at the maximum, median and minimum points.

In practice, the box-whisker symbol is not limited to representing the maximum, 3rd quartile, median, 1st quartile and minimum values. It can be used to represent any ordered values.

The boxTop and boxBottom data sets specifies the top and bottom edges of the box. The maxData, minData and midData specifies the top, bottom and middle mark lines.

You can use empty arrays to disable showing some parts of the box-whisker symbol.

For example, if you just want to show a floating box, you can use only boxTop and boxBottom and set the maxData, minData and midData to empty arrays.

Similarly, if the boxTop, boxBottom and midData are set to empty arrays, only the top and bottom mark lines and the joining center line are visible. This style is most often used as "error bands" together with line charts.

In addition to maxData, minData and midData, you can add additional mark lines to the box-whisker element by adding more data sets using Layer.addDataSet.

By default, the box-whisker symbol will be drawn using the colors specified in the fillColor, whiskerColor and edgeColor argument. The fillColor and edgeColor are used as the fill and border colors of the box, while the whiskerColor is used as the color of the center line and the mark lines.

Internally, ChartDirector maps the colors of different parts of the box-whisker symbol to data set colors as shown in the following table. You may control the colors of the box-whisker symbol in more details by setting the data set colors directly. The data set objects can be obtained using Layer.getDataSet, and the colors can be changed using DataSet.setDataColor.

Return Value

A BoxWhiskerLayer object representing the box-whisker layer created.

See also:

- 3.67.21 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS 763

Box-Whisker Symbol Color	Data Set Color
Fill color of the box	Data Color for the first data set (index = 0)
Border color of the box	Edge Color for the first data set (index = 0)
Center line color	Data Color for the second data set (index = 1)
Maximum value mark line color	Data Color for the third data set (index = 2)
Minimum value mark line color	Data Color for the fourth data set (index = 3)
Middle value mark line color	Data Color for the fifth data set (index = 4)
Mark line color for additional mark lines	Data Color for the data set representing the additional mark line

Parameter	Default	Description
boxTop	(Mandatory)	An array of numbers representing the top edge of the box.
boxBottom	(Mandatory)	An array of numbers representing the bottom edge of the box.
maxData	[Empty_Array]	An array of numbers representing the maximum value mark lines.
minData	[Empty_Array]	An array of numbers representing the minimum value mark lines.
midData	[Empty_Array]	An array of numbers representing the middle value mark lines.
fillColor	-1	The color used to fill the box. -1 means that the color is automatically selected from the palette.
whiskerColor	LineColor	The color used to draw the central line and mark lines.
edgeColor	LineColor	The color used to draw the border of the box.

- 3.67.22 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS` 765
- 3.67.23 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as color, whiskerColor as color, edgeColor as color) as CDBoxWhiskerLayerMBS` 767
- 3.67.24 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as Integer = -1, whiskerColor as Integer = &hfff0001, edgeColor as Integer = -1) as CDBoxWhiskerLayerMBS` 767

3.67.21 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: Traditionally, each box-whisker symbol represents 5 values, which are the as maximum, 3rd quartile, median, 1st quartile and minimum of some data samples.

A vertical box-whisker symbol (on a non-swapped XY chart) consists of a vertical line joining the maximum and minimum points, a box extending from the 1st quartile point to the 3rd quartile point, and 3 horizontal mark lines at the maximum, median and minimum points.

In practice, the box-whisker symbol is not limited to representing the maximum, 3rd quartile, median, 1st quartile and minimum values. It can be used to represent any ordered values.

The `boxTop` and `boxBottom` data sets specifies the top and bottom edges of the box. The `maxData`, `minData` and `midData` specifies the top, bottom and middle mark lines.

You can use empty arrays to disable showing some parts of the box-whisker symbol.

For example, if you just want to show a floating box, you can use only `boxTop` and `boxBottom` and set the `maxData`, `minData` and `midData` to empty arrays.

Similarly, if the `boxTop`, `boxBottom` and `midData` are set to empty arrays, only the top and bottom mark lines and the joining center line are visible. This style is most often used as "error bands" together with line charts.

In addition to `maxData`, `minData` and `midData`, you can add additional mark lines to the box-whisker element by adding more data sets using `Layer.addDataSet`.

By default, the box-whisker symbol will be drawn using the colors specified in the `fillColor`, `whiskerColor` and `edgeColor` argument. The `fillColor` and `edgeColor` are used as the fill and border colors of the box, while the `whiskerColor` is used as the color of the center line and the mark lines.

Internally, `ChartDirector` maps the colors of different parts of the box-whisker symbol to data set colors as shown in the following table. You may control the colors of the box-whisker symbol in more details by setting the data set colors directly. The data set objects can be obtained using `Layer.getDataSet`, and the colors can be changed using `DataSet.setDataColor`.

Box-Whisker Symbol Color	Data Set Color
Fill color of the box	Data Color for the first data set (index = 0)
Border color of the box	Edge Color for the first data set (index = 0)
Center line color	Data Color for the second data set (index = 1)
Maximum value mark line color	Data Color for the third data set (index = 2)
Minimum value mark line color	Data Color for the fourth data set (index = 3)
Middle value mark line color	Data Color for the fifth data set (index = 4)
Mark line color for additional mark lines	Data Color for the data set representing the additional mark line

Return Value

A `BoxWhiskerLayer` object representing the box-whisker layer created.

See also:

- 3.67.20 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS` 761
- 3.67.22 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS` 765

Parameter	Default	Description
boxTop	(Mandatory)	An array of numbers representing the top edge of the box.
boxBottom	(Mandatory)	An array of numbers representing the bottom edge of the box.
maxData	[Empty_Array]	An array of numbers representing the maximum value mark lines.
minData	[Empty_Array]	An array of numbers representing the minimum value mark lines.
midData	[Empty_Array]	An array of numbers representing the middle value mark lines.
fillColor	-1	The color used to fill the box. -1 means that the color is automatically selected from the palette.
whiskerColor	LineColor	The color used to draw the central line and mark lines.
edgeColor	LineColor	The color used to draw the border of the box.

- 3.67.23 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as color, whiskerColor as color, edgeColor as color) as CDBoxWhiskerLayerMBS 767
- 3.67.24 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as Integer = -1, whiskerColor as Integer = &hfff0001, edgeColor as Integer = -1) as CDBoxWhiskerLayerMBS 767

3.67.22 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.
Notes: Traditionally, each box-whisker symbol represents 5 values, which are the as maximum, 3rd quartile, median, 1st quartile and minimum of some data samples.

A vertical box-whisker symbol (on a non-swapped XY chart) consists of a vertical line joining the maximum and minimum points, a box extending from the 1st quartile point to the 3rd quartile point, and 3 horizontal mark lines at the maximum, median and minimum points.

In practice, the box-whisker symbol is not limited to representing the maximum, 3rd quartile, median, 1st quartile and minimum values. It can be used to represent any ordered values.

The boxTop and boxBottom data sets specifies the top and bottom edges of the box. The maxData, minData and midData specifies the top, bottom and middle mark lines.

You can use empty arrays to disable showing some parts of the box-whisker symbol.

For example, if you just want to show a floating box, you can use only boxTop and boxBottom and set the maxData, minData and midData to empty arrays.

Similarly, if the `boxTop`, `boxBottom` and `midData` are set to empty arrays, only the top and bottom mark lines and the joining center line are visible. This style is most often used as "error bands" together with line charts.

In addition to `maxData`, `minData` and `midData`, you can add additional mark lines to the box-whisker element by adding more data sets using `Layer.addDataSet`.

By default, the box-whisker symbol will be drawn using the colors specified in the `fillColor`, `whiskerColor` and `edgeColor` argument. The `fillColor` and `edgeColor` are used as the fill and border colors of the box, while the `whiskerColor` is used as the color of the center line and the mark lines.

Internally, `ChartDirector` maps the colors of different parts of the box-whisker symbol to data set colors as shown in the following table. You may control the colors of the box-whisker symbol in more details by setting the data set colors directly. The data set objects can be obtained using `Layer.getDataSet`, and the colors can be changed using `DataSet.setDataColor`.

Box-Whisker Symbol Color	Data Set Color
Fill color of the box	Data Color for the first data set (index = 0)
Border color of the box	Edge Color for the first data set (index = 0)
Center line color	Data Color for the second data set (index = 1)
Maximum value mark line color	Data Color for the third data set (index = 2)
Minimum value mark line color	Data Color for the fourth data set (index = 3)
Middle value mark line color	Data Color for the fifth data set (index = 4)
Mark line color for additional mark lines	Data Color for the data set representing the additional mark line

Parameter	Default	Description
<code>boxTop</code>	(Mandatory)	An array of numbers representing the top edge of the box.
<code>boxBottom</code>	(Mandatory)	An array of numbers representing the bottom edge of the box.
<code>maxData</code>	[Empty_Array]	An array of numbers representing the maximum value mark lines.
<code>minData</code>	[Empty_Array]	An array of numbers representing the minimum value mark lines.
<code>midData</code>	[Empty_Array]	An array of numbers representing the middle value mark lines.
<code>fillColor</code>	-1	The color used to fill the box. -1 means that the color is automatically selected from the palette.
<code>whiskerColor</code>	LineColor	The color used to draw the central line and mark lines.
<code>edgeColor</code>	LineColor	The color used to draw the border of the box.

Return Value

A `BoxWhiskerLayer` object representing the box-whisker layer created.

See also:

- 3.67.20 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS` 761
- 3.67.21 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS` 763

- 3.67.23 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as color, whiskerColor as color, edgeColor as color) as CDBoxWhiskerLayerMBS 767
- 3.67.24 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as Integer = -1, whiskerColor as Integer = &hfff0001, edgeColor as Integer = -1) as CDBoxWhiskerLayerMBS 767

3.67.23 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as color, whiskerColor as color, edgeColor as color) as CDBoxWhiskerLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBoxWhiskerLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.20 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS 761
- 3.67.21 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS 763
- 3.67.22 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 765
- 3.67.24 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as Integer = -1, whiskerColor as Integer = &hfff0001, edgeColor as Integer = -1) as CDBoxWhiskerLayerMBS 767

3.67.24 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as Integer = -1, whiskerColor as Integer = &hfff0001, edgeColor as Integer = -1) as CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: Traditionally, each box-whisker symbol represents 5 values, which are the as maximum, 3rd quartile, median, 1st quartile and minimum of some data samples.

A vertical box-whisker symbol (on a non-swapped XY chart) consists of a vertical line joining the maximum and minimum points, a box extending from the 1st quartile point to the 3rd quartile point, and 3 horizontal mark lines at the maximum, median and minimum points.

In practice, the box-whisker symbol is not limited to representing the maximum, 3rd quartile, median, 1st quartile and minimum values. It can be used to represent any ordered values.

The `boxTop` and `boxBottom` data sets specifies the top and bottom edges of the box. The `maxData`, `minData` and `midData` specifies the top, bottom and middle mark lines.

You can use empty arrays to disable showing some parts of the box-whisker symbol.

For example, if you just want to show a floating box, you can use only `boxTop` and `boxBottom` and set the `maxData`, `minData` and `midData` to empty arrays.

Similarly, if the `boxTop`, `boxBottom` and `midData` are set to empty arrays, only the top and bottom mark lines and the joining center line are visible. This style is most often used as "error bands" together with line charts.

In addition to `maxData`, `minData` and `midData`, you can add additional mark lines to the box-whisker element by adding more data sets using `Layer.addDataSet`.

By default, the box-whisker symbol will be drawn using the colors specified in the `fillColor`, `whiskerColor` and `edgeColor` argument. The `fillColor` and `edgeColor` are used as the fill and border colors of the box, while the `whiskerColor` is used as the color of the center line and the mark lines.

Internally, `ChartDirector` maps the colors of different parts of the box-whisker symbol to data set colors as shown in the following table. You may control the colors of the box-whisker symbol in more details by setting the data set colors directly. The data set objects can be obtained using `Layer.getDataSet`, and the colors can be changed using `DataSet.setDataColor`.

Box-Whisker Symbol Color	Data Set Color
Fill color of the box	Data Color for the first data set (index = 0)
Border color of the box	Edge Color for the first data set (index = 0)
Center line color	Data Color for the second data set (index = 1)
Maximum value mark line color	Data Color for the third data set (index = 2)
Minimum value mark line color	Data Color for the fourth data set (index = 3)
Middle value mark line color	Data Color for the fifth data set (index = 4)
Mark line color for additional mark lines	Data Color for the data set representing the additional mark line

Return Value

A `BoxWhiskerLayer` object representing the box-whisker layer created.

See also:

- 3.67.20 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS` 761
- 3.67.21 `addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double) as`

Parameter	Default	Description
boxTop	(Mandatory)	An array of numbers representing the top edge of the box.
boxBottom	(Mandatory)	An array of numbers representing the bottom edge of the box.
maxData	[Empty_Array]	An array of numbers representing the maximum value mark lines.
minData	[Empty_Array]	An array of numbers representing the minimum value mark lines.
midData	[Empty_Array]	An array of numbers representing the middle value mark lines.
fillColor	-1	The color used to fill the box. -1 means that the color is automatically selected from the palette.
whiskerColor	LineColor	The color used to draw the central line and mark lines.
edgeColor	LineColor	The color used to draw the border of the box.

CDBoxWhiskerLayerMBS

763

- 3.67.22 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 765
- 3.67.23 addBoxWhiskerLayer(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColor as color, whiskerColor as color, edgeColor as color) as CDBoxWhiskerLayerMBS 767

3.67.25 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-color box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: This method is similar to XYChart.addBoxWhiskerLayer, except that the layer will be added in multi-color mode. Please refer to XYChart.addBoxWhiskerLayer on basic information of what is a box-whisker layer.

In multi-color mode, the boxes of a BoxWhiskerLayer can have different fill colors. For the whisker, instead of specifying a single whisker color for all boxes, ChartDirector computes the whisker colors by darkening or brightening the corresponding fill colors using a configurable brightness factor.

Return Value

A BoxWhiskerLayer object representing the box-whisker layer created.

See also:

- 3.67.26 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS 770
- 3.67.27 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 771
- 3.67.28 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double) as CDBoxWhiskerLayerMBS 773

Parameter	Default	Description
boxTop	(Mandatory)	An array of numbers representing the top edge of the box.
boxBottom	(Mandatory)	An array of numbers representing the bottom edge of the box.
maxData	[Empty_Array]	An array of numbers representing the maximum value mark lines.
minData	[Empty_Array]	An array of numbers representing the minimum value mark lines.
midData	[Empty_Array]	An array of numbers representing the middle value mark lines.
fillColors	[Empty_Array]	An array of colors to be used as the fill color of the boxes. If there are insufficient colors in the array for the boxes, the remaining boxes will have their colors automatically selected from the palette.
whiskerBrightness	0.5	The brightness factor for whisker color. A value less than 1 means darkening. A value larger than 1 means brightening. A zero value means black.
names	[Empty_Array]	An array of names for the boxes to be used in the legend box, if one is configured for the chart.

- 3.67.29 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double = 0.5)` as `CDBoxWhiskerLayerMBS` 774
- 3.67.30 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double, names() as string)` as `CDBoxWhiskerLayerMBS` 775
- 3.67.31 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double = 0.5)` as `CDBoxWhiskerLayerMBS` 775
- 3.67.32 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double, names() as string)` as `CDBoxWhiskerLayerMBS` 777

3.67.26 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double)` as `CDBoxWhiskerLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-color box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: This method is similar to `XYChart.addBoxWhiskerLayer`, except that the layer will be added in multi-color mode. Please refer to `XYChart.addBoxWhiskerLayer` on basic information of what is a box-whisker layer.

In multi-color mode, the boxes of a `BoxWhiskerLayer` can have different fill colors. For the whisker, instead of specifying a single whisker color for all boxes, `ChartDirector` computes the whisker colors by darkening or brightening the corresponding fill colors using a configurable brightness factor.

Return Value

A `BoxWhiskerLayer` object representing the box-whisker layer created.

Parameter	Default	Description
boxTop	(Mandatory)	An array of numbers representing the top edge of the box.
boxBottom	(Mandatory)	An array of numbers representing the bottom edge of the box.
maxData	[Empty_Array]	An array of numbers representing the maximum value mark lines.
minData	[Empty_Array]	An array of numbers representing the minimum value mark lines.
midData	[Empty_Array]	An array of numbers representing the middle value mark lines.
fillColors	[Empty_Array]	An array of colors to be used as the fill color of the boxes. If there are insufficient colors in the array for the boxes, the remaining boxes will have their colors automatically selected from the palette.
whiskerBrightness	0.5	The brightness factor for whisker color. A value less than 1 means darkening. A value larger than 1 means brightening. A zero value means black.
names	[Empty_Array]	An array of names for the boxes to be used in the legend box, if one is configured for the chart.

See also:

- 3.67.25 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS 769
- 3.67.27 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 771
- 3.67.28 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double) as CDBoxWhiskerLayerMBS 773
- 3.67.29 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS 774
- 3.67.30 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS 775
- 3.67.31 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS 775
- 3.67.32 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS 777

3.67.27 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-color box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: This method is similar to `XYChart.addBoxWhiskerLayer`, except that the layer will be added in multi-color mode. Please refer to `XYChart.addBoxWhiskerLayer` on basic information of what is a box-whisker layer.

In multi-color mode, the boxes of a `BoxWhiskerLayer` can have different fill colors. For the whisker, instead of specifying a single whisker color for all boxes, `ChartDirector` computes the whisker colors by darkening or brightening the corresponding fill colors using a configurable brightness factor.

Parameter	Default	Description
<code>boxTop</code>	(Mandatory)	An array of numbers representing the top edge of the box.
<code>boxBottom</code>	(Mandatory)	An array of numbers representing the bottom edge of the box.
<code>maxData</code>	[Empty_Array]	An array of numbers representing the maximum value mark lines.
<code>minData</code>	[Empty_Array]	An array of numbers representing the minimum value mark lines.
<code>midData</code>	[Empty_Array]	An array of numbers representing the middle value mark lines.
<code>fillColors</code>	[Empty_Array]	An array of colors to be used as the fill color of the boxes. If there are insufficient colors in the array for the boxes, the remaining boxes will have their colors automatically selected from the palette.
<code>whiskerBrightness</code>	0.5	The brightness factor for whisker color. A value less than 1 means darkening. A value larger than 1 means brightening. A zero value means black.
<code>names</code>	[Empty_Array]	An array of names for the boxes to be used in the legend box, if one is configured for the chart.

Return Value

A `BoxWhiskerLayer` object representing the box-whisker layer created.

See also:

- 3.67.25 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS` 769
- 3.67.26 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS` 770
- 3.67.28 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double) as CDBoxWhiskerLayerMBS` 773
- 3.67.29 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS` 774
- 3.67.30 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS` 775
- 3.67.31 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS` 775
- 3.67.32 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS` 777

3.67.28 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double) as CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-color box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: This method is similar to XYChart.addBoxWhiskerLayer, except that the layer will be added in multi-color mode. Please refer to XYChart.addBoxWhiskerLayer on basic information of what is a box-whisker layer.

In multi-color mode, the boxes of a BoxWhiskerLayer can have different fill colors. For the whisker, instead of specifying a single whisker color for all boxes, ChartDirector computes the whisker colors by darkening or brightening the corresponding fill colors using a configurable brightness factor.

Parameter	Default	Description
boxTop	(Mandatory)	An array of numbers representing the top edge of the box.
boxBottom	(Mandatory)	An array of numbers representing the bottom edge of the box.
maxData	[Empty_Array]	An array of numbers representing the maximum value mark lines.
minData	[Empty_Array]	An array of numbers representing the minimum value mark lines.
midData	[Empty_Array]	An array of numbers representing the middle value mark lines.
fillColors	[Empty_Array]	An array of colors to be used as the fill color of the boxes. If there are insufficient colors in the array for the boxes, the remaining boxes will have their colors automatically selected from the palette.
whiskerBrightness	0.5	The brightness factor for whisker color. A value less than 1 means darkening. A value larger than 1 means brightening. A zero value means black.
names	[Empty_Array]	An array of names for the boxes to be used in the legend box, if one is configured for the chart.

Return Value

A BoxWhiskerLayer object representing the box-whisker layer created.

See also:

- 3.67.25 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS 769
- 3.67.26 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS 770
- 3.67.27 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 771
- 3.67.29 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS 774
- 3.67.30 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double,

minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS 775

- 3.67.31 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS 775
- 3.67.32 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS 777

3.67.29 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBoxWhiskerLayer2 method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.25 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS 769
- 3.67.26 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS 770
- 3.67.27 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 771
- 3.67.28 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double) as CDBoxWhiskerLayerMBS 773
- 3.67.30 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS 775
- 3.67.31 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS 775
- 3.67.32 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS 777

3.67.30 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addBoxWhiskerLayer2 method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.25 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS 769
- 3.67.26 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS 770
- 3.67.27 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 771
- 3.67.28 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double) as CDBoxWhiskerLayerMBS 773
- 3.67.29 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS 774
- 3.67.31 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS 775
- 3.67.32 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS 777

3.67.31 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-color box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: This method is similar to XYChart.addBoxWhiskerLayer, except that the layer will be added in multi-color mode. Please refer to XYChart.addBoxWhiskerLayer on basic information of what is a box-whisker layer.

In multi-color mode, the boxes of a `BoxWhiskerLayer` can have different fill colors. For the whisker, instead of specifying a single whisker color for all boxes, `ChartDirector` computes the whisker colors by darkening or brightening the corresponding fill colors using a configurable brightness factor.

Parameter	Default	Description
<code>boxTop</code>	(Mandatory)	An array of numbers representing the top edge of the box.
<code>boxBottom</code>	(Mandatory)	An array of numbers representing the bottom edge of the box.
<code>maxData</code>	[Empty_Array]	An array of numbers representing the maximum value mark lines.
<code>minData</code>	[Empty_Array]	An array of numbers representing the minimum value mark lines.
<code>midData</code>	[Empty_Array]	An array of numbers representing the middle value mark lines.
<code>fillColors</code>	[Empty_Array]	An array of colors to be used as the fill color of the boxes. If there are insufficient colors in the array for the boxes, the remaining boxes will have their colors automatically selected from the palette.
<code>whiskerBrightness</code>	0.5	The brightness factor for whisker color. A value less than 1 means darkening. A value larger than 1 means brightening. A zero value means black.
<code>names</code>	[Empty_Array]	An array of names for the boxes to be used in the legend box, if one is configured for the chart.

Return Value

A `BoxWhiskerLayer` object representing the box-whisker layer created.

See also:

- 3.67.25 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS` 769
- 3.67.26 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS` 770
- 3.67.27 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS` 771
- 3.67.28 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double) as CDBoxWhiskerLayerMBS` 773
- 3.67.29 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS` 774
- 3.67.30 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS` 775
- 3.67.32 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS` 777

3.67.32 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-color box-whisker layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: This method is similar to XYChart.addBoxWhiskerLayer, except that the layer will be added in multi-color mode. Please refer to XYChart.addBoxWhiskerLayer on basic information of what is a box-whisker layer.

In multi-color mode, the boxes of a BoxWhiskerLayer can have different fill colors. For the whisker, instead of specifying a single whisker color for all boxes, ChartDirector computes the whisker colors by darkening or brightening the corresponding fill colors using a configurable brightness factor.

Parameter	Default	Description
boxTop	(Mandatory)	An array of numbers representing the top edge of the box.
boxBottom	(Mandatory)	An array of numbers representing the bottom edge of the box.
maxData	[Empty_Array]	An array of numbers representing the maximum value mark lines.
minData	[Empty_Array]	An array of numbers representing the minimum value mark lines.
midData	[Empty_Array]	An array of numbers representing the middle value mark lines.
fillColors	[Empty_Array]	An array of colors to be used as the fill color of the boxes. If there are insufficient colors in the array for the boxes, the remaining boxes will have their colors automatically selected from the palette.
whiskerBrightness	0.5	The brightness factor for whisker color. A value less than 1 means darkening. A value larger than 1 means brightening. A zero value means black.
names	[Empty_Array]	An array of names for the boxes to be used in the legend box, if one is configured for the chart.

Return Value

A BoxWhiskerLayer object representing the box-whisker layer created.

See also:

- 3.67.25 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double) as CDBoxWhiskerLayerMBS 769
- 3.67.26 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double) as CDBoxWhiskerLayerMBS 770
- 3.67.27 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double) as CDBoxWhiskerLayerMBS 771
- 3.67.28 addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double) as CDBoxWhiskerLayerMBS 773

- 3.67.29 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS` 774
- 3.67.30 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as color, whiskerBrightness as Double, names() as string) as CDBoxWhiskerLayerMBS` 775
- 3.67.31 `addBoxWhiskerLayer2(boxTop() as Double, boxBottom() as Double, maxData() as Double, minData() as Double, midData() as Double, fillColors() as Integer, whiskerBrightness as Double = 0.5) as CDBoxWhiskerLayerMBS` 775

3.67.33 `addCandleStickLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, riseColor as color, fallColor as color, edgeColor as color) as CDCandleStickLayerMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addCandleStickLayer` method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.34 `addCandleStickLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, riseColor as Integer = -1, fallColor as Integer = 0, edgeColor as Integer = &hfff0001) as CDCandleStickLayerMBS` 778

3.67.34 `addCandleStickLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, riseColor as Integer = -1, fallColor as Integer = 0, edgeColor as Integer = &hfff0001) as CDCandleStickLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a candlestick layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: By default, the candlestick symbol will be drawn using the colors specified in the `riseColor`, `fallColor` and `edgeColor` argument. The `riseColor` and `fallColor` are used to fill the candle depending on whether the opening value or closing value is larger. The `edgeColor` is used to the center line and the border of the candle.

Internally, ChartDirector maps the colors of different parts of the candlestick symbol to data set colors as shown in the following table. You may control the colors of the candlestick symbol in more details by setting the data set colors directly. The data set objects can be obtained using `Layer.getDataSet`, and the colors can be changed using `DataSet.setDataColor`.

Return Value

A `CandleStickLayer` object representing the candlestick layer created.

See also:

Candlestick Symbol Color	Data Set Color
Fill color for "up" candlesticks	Data Color for the first data set (index = 0).
Fill color for "down" candlesticks	Data Color for the second data set (index = 1).
Border color	Edge Color for the first data set (index = 0).
Center line color	Edge Color for the second data set (index = 1).

Parameter	Default	Description
highData	(Mandatory)	An array of numbers representing the high values.
lowData	(Mandatory)	An array of numbers representing the low values.
openData	(Mandatory)	An array of numbers representing the opening values.
closeData	(Mandatory)	An array of numbers representing the closing values.
riseColor	FFFFFF	The color used to fill the candle if the closing value is higher than the opening value.
fallColor	000000	The color used to fill the candle if the opening value is higher than the closing value.
edgeColor	LineColor	The color used to draw the center line and the border of the candle.

- 3.67.33 addCandleStickLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, riseColor as color, fallColor as color, edgeColor as color) as CDCandleStickLayerMBS
778

3.67.35 addContourLayer(xData() as Double, yData() as Double, zData() as Double) as CDContourLayerMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a contour layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: ChartDirector supports both gridded and scattered data. If the data points are on a rectangular grid will no missing points, they will be handled as gridded data. Otherwise, they will be handled as scattered data.

For gridded data, you may provide the x and y values of the grid, and the z values of the data points. For a 10 x 15 grid, that means the x data series should have 10 values, the y data series should have 15 values, and the z data series should have 150 values. The x and y data series should be strictly monotonic (either strictly increasing or strictly decreasing).

For both gridded and scattered data, you may also provide the (x, y, z) values of the data points. For example, for 150 data points, the x, y and z data series should each have 150 values. ChartDirector will automatically detect if the data points are gridded or scattered.

Arguments:

Return Value:

A ContourLayer object representing the contour layer created.

Argument	Default	Description
xData	(Mandatory)	An array of numbers representing the x data series.
yData	(Mandatory)	An array of numbers representing the y data series.
zData	(Mandatory)	An array of numbers representing the z data series.

3.67.36 addDiscreteHeatMapLayer(xGrid() as double, yGrid() as Double, zData() as Double) as CDDiscreteHeatMapLayerMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a discrete heat map layer to the XYChart by specifying the cell values and positions.

Notes: The cells are assumed to be arranged as a regular grid. The xGrid and yGrid arrays can be the center coordinates of the cells or the boundary coordinates of the cells.

Suppose there are 9 cells (the zData array contains 9 values). If the xGrid and yGrid arrays each contains 3 values, the cells will be assumed to be arranged as a 3 x 3 grid, and the xGrid and yGrid will be assumed to be the center coordinates of the cells.

If the xGrid array and yGrid array each contains 4 values instead, they will be assumed to be the boundaries of the cells. The first column of cells will occupy the space between xGrid [0] and xGrid [1], the second column between xGrid [1] and xGrid [2], and the third column between xGrid [2] and xGrid [3]. The same applies to the yGrid.

In general, if xGrid and yGrid contains enough values to be used as the boundary coordinates, they will be assumed to be the boundary coordinates. Otherwise they will be assumed to be the center coordinates.

Argument	Default	Description
xGrid	(Mandatory)	An array containing the x coordinates of the centers or boundaries of the cells.
yGrid	(Mandatory)	An array containing the y coordinates of the centers or boundaries of the cells.
zData	(Mandatory)	An array containing the data values of the cells.

Returns a DiscreteHeatMapLayer object representing the discrete heat map layer created.

See also:

- 3.67.37 addDiscreteHeatMapLayer(zData() as Double, xCount as Integer) as CDDiscreteHeatMapLayerMBS 780

3.67.37 addDiscreteHeatMapLayer(zData() as Double, xCount as Integer) as CDDiscreteHeatMapLayerMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a discrete heat map layer to the XYChart by specifying the cell values and the cell count in the x direction.

Notes: The zData array contains the cell values. xCount is the cell count in the x direction. The cell count in the y direction will be computed as the total cell count (the size of the zData array) divided by xCount.

The boundary of the cells will be at integer axis coordinates 0, 1, 2, In the x direction, the first column of cells will occupy the space between $x = 0$ and $x = 1$, the second column between $x = 1$ and $x = 2$, and so on. The same applies to the y direction. The first row of cells will be between $y = 0$ and $y = 1$, the second row between $y = 1$ and $y = 2$, and so on.

This method is often used with Axis.setLabels and Axis.setLabelOffset for axis labelling. The Axis.setLabels put labels at coordinates 0, 1, 2, ... along the axis. The Axis.setLabelOffset can offset the labels by 0.5 units, so the labels will be at 0.5, 1.5, 2.5 ... which corresponds to the center of the cells.

A similar method XYChart.addDiscreteHeatMapLayer2 allows you to specify the exact x and y coordinates for the cells.

Argument	Default	Description
zData	(Mandatory)	An array containing the data values of the cells.
xCount	(Mandatory)	The number of cells in the x direction. The number of cells in the y direction will be computed as the total number of cells (the size of the zData array) divided by xCount.

See also:

- 3.67.36 addDiscreteHeatMapLayer(xGrid() as double, yGrid() as Double, zData() as Double) as CD-DiscreteHeatMapLayerMBS 780

3.67.38 addHLOCLayer as CDHLOCLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an empty high-low-open-close (HLOC) layer to the XYChart.

Notes: Return Value

A HLOCLayer object representing the HLOC layer created.

See also:

- 3.67.39 addHLOCLayer(highData() as Double, lowData() as Double) as CDHLOCLayerMBS 782
- 3.67.40 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double) as CDHLOCLayerMBS 783
- 3.67.41 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as color) as CDHLOCLayerMBS 785

- 3.67.42 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS 785
- 3.67.43 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer = -1) as CDHLOCLayerMBS 787
- 3.67.44 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 787
- 3.67.45 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer = -1) as CDHLOCLayerMBS 788
- 3.67.46 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 789

3.67.39 addHLOCLayer(highData() as Double, lowData() as Double) as CDHLOCLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a high-low-open-close (HLOC) layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: In this method, all HLOC symbols will be drawn using the color specified in the color argument. Use XYChart.addHLOCLayer for multi-color HLOC layers.

Internally, ChartDirector maps the colors of different parts of the HLOC symbol to data set colors as shown in the following table. You may control the colors of the HLOC symbol in more details by setting the data set colors directly. The data set objects can be obtained using Layer.getDataSet, and the colors can be changed using DataSet.setDataColor.

HLOC Symbol Color	Data Set Color
Central Line	Data Color for the first data set (index = 0)
Open Mark Line	Data Color for the third data set (index = 2)
Close Mark Line	Data Color for the fourth data set (index = 3)

Return Value

A HLOCLayer object representing the HLOC layer created.

See also:

- 3.67.38 addHLOCLayer as CDHLOCLayerMBS 781
- 3.67.40 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double) as CDHLOCLayerMBS 783

Parameter	Default	Description
highData	(Mandatory)	An array of numbers representing the high values. An empty array means there is no high value data available.
lowData	(Mandatory)	An array of numbers representing the low values. An empty array means there is no low value data available.
openData	[Empty_Array]	An array of numbers representing the opening values. An empty array means there is no opening value data available.
closeData	[Empty_Array]	An array of numbers representing the closing values. An empty array means there is no closing value data available.
color	-1	The color to draw the HLOC symbol. -1 means that the color is automatically selected from the palette.

- 3.67.41 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as color) as CDHLOCLayerMBS 785
- 3.67.42 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS 785
- 3.67.43 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer = -1) as CDHLOCLayerMBS 787
- 3.67.44 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 787
- 3.67.45 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer = -1) as CDHLOCLayerMBS 788
- 3.67.46 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 789

3.67.40 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double) as CDHLOCLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a high-low-open-close (HLOC) layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: In this method, all HLOC symbols will be drawn using the color specified in the color argument. Use XYChart.addHLOCLayer for multi-color HLOC layers.

Internally, ChartDirector maps the colors of different parts of the HLOC symbol to data set colors as shown in the following table. You may control the colors of the HLOC symbol in more details by setting the data set colors directly. The data set objects can be obtained using Layer.getDataSet, and the colors can be

changed using `DataSet.setDataColor`.

HLOC Symbol Color	Data Set Color
Central Line	Data Color for the first data set (index = 0)
Open Mark Line	Data Color for the third data set (index = 2)
Close Mark Line	Data Color for the fourth data set (index = 3)

Parameter	Default	Description
<code>highData</code>	(Mandatory)	An array of numbers representing the high values. An empty array means there is no high value data available.
<code>lowData</code>	(Mandatory)	An array of numbers representing the low values. An empty array means there is no low value data available.
<code>openData</code>	[Empty_Array]	An array of numbers representing the opening values. An empty array means there is no opening value data available.
<code>closeData</code>	[Empty_Array]	An array of numbers representing the closing values. An empty array means there is no closing value data available.
<code>color</code>	-1	The color to draw the HLOC symbol. -1 means that the color is automatically selected from the palette.

Return Value

A `HLOCLayer` object representing the HLOC layer created.

See also:

- 3.67.38 `addHLOCLayer` as `CDHLOCLayerMBS` 781
- 3.67.39 `addHLOCLayer(highData() as Double, lowData() as Double) as CDHLOCLayerMBS` 782
- 3.67.41 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as color) as CDHLOCLayerMBS` 785
- 3.67.42 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS` 785
- 3.67.43 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer = -1) as CDHLOCLayerMBS` 787
- 3.67.44 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS` 787
- 3.67.45 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer = -1) as CDHLOCLayerMBS` 788
- 3.67.46 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS` 789

3.67.41 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as color) as CDHLOCLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addHLOCLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.38 addHLOCLayer as CDHLOCLayerMBS 781
- 3.67.39 addHLOCLayer(highData() as Double, lowData() as Double) as CDHLOCLayerMBS 782
- 3.67.40 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double) as CDHLOCLayerMBS 783
- 3.67.42 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS 785
- 3.67.43 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer = -1) as CDHLOCLayerMBS 787
- 3.67.44 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 787
- 3.67.45 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer = -1) as CDHLOCLayerMBS 788
- 3.67.46 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 789

3.67.42 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a high-low-open-close (HLOC) layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: In this method, all HLOC symbols will be drawn using the color specified in the color argument. Use XYChart.addHLOCLayer for multi-color HLOC layers.

Internally, ChartDirector maps the colors of different parts of the HLOC symbol to data set colors as shown in the following table. You may control the colors of the HLOC symbol in more details by setting the data

set colors directly. The data set objects can be obtained using `Layer.getDataSet`, and the colors can be changed using `DataSet.setDataColor`.

HLOC Symbol Color	Data Set Color
Central Line	Data Color for the first data set (index = 0)
Open Mark Line	Data Color for the third data set (index = 2)
Close Mark Line	Data Color for the fourth data set (index = 3)

Parameter	Default	Description
<code>highData</code>	(Mandatory)	An array of numbers representing the high values. An empty array means there is no high value data available.
<code>lowData</code>	(Mandatory)	An array of numbers representing the low values. An empty array means there is no low value data available.
<code>openData</code>	[Empty_Array]	An array of numbers representing the opening values. An empty array means there is no opening value data available.
<code>closeData</code>	[Empty_Array]	An array of numbers representing the closing values. An empty array means there is no closing value data available.
<code>color</code>	-1	The color to draw the HLOC symbol. -1 means that the color is automatically selected from the palette.

Return Value

A `HLOCLayer` object representing the HLOC layer created.

See also:

- 3.67.38 `addHLOCLayer` as `CDHLOCLayerMBS` 781
- 3.67.39 `addHLOCLayer`(`highData()` as `Double`, `lowData()` as `Double`) as `CDHLOCLayerMBS` 782
- 3.67.40 `addHLOCLayer`(`highData()` as `Double`, `lowData()` as `Double`, `openData()` as `Double`) as `CDHLOCLayerMBS` 783
- 3.67.41 `addHLOCLayer`(`highData()` as `Double`, `lowData()` as `Double`, `openData()` as `Double`, `closeData()` as `Double`, `colorValue` as `color`) as `CDHLOCLayerMBS` 785
- 3.67.43 `addHLOCLayer`(`highData()` as `Double`, `lowData()` as `Double`, `openData()` as `Double`, `closeData()` as `Double`, `upColor` as `color`, `downColor` as `color`, `colorMode` as `Integer = -1`) as `CDHLOCLayerMBS` 787
- 3.67.44 `addHLOCLayer`(`highData()` as `Double`, `lowData()` as `Double`, `openData()` as `Double`, `closeData()` as `Double`, `upColor` as `color`, `downColor` as `color`, `colorMode` as `Integer`, `leadValue` as `Double`) as `CDHLOCLayerMBS` 787
- 3.67.45 `addHLOCLayer`(`highData()` as `Double`, `lowData()` as `Double`, `openData()` as `Double`, `closeData()` as `Double`, `upColor` as `Integer`, `downColor` as `Integer`, `colorMode` as `Integer = -1`) as `CDHLOCLayerMBS` 788
- 3.67.46 `addHLOCLayer`(`highData()` as `Double`, `lowData()` as `Double`, `openData()` as `Double`, `closeData()` as `Double`, `upColor` as `Integer`, `downColor` as `Integer`, `colorMode` as `Integer`, `leadValue` as `Double`) as `CDHLOCLayerMBS` 789

3.67.43 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer = -1) as CDHLOCLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addHLOCLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.38 addHLOCLayer as CDHLOCLayerMBS 781
- 3.67.39 addHLOCLayer(highData() as Double, lowData() as Double) as CDHLOCLayerMBS 782
- 3.67.40 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double) as CDHLOCLayerMBS 783
- 3.67.41 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as color) as CDHLOCLayerMBS 785
- 3.67.42 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS 785
- 3.67.44 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 787
- 3.67.45 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer = -1) as CDHLOCLayerMBS 788
- 3.67.46 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 789

3.67.44 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addHLOCLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.38 addHLOCLayer as CDHLOCLayerMBS 781
- 3.67.39 addHLOCLayer(highData() as Double, lowData() as Double) as CDHLOCLayerMBS 782

- 3.67.40 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double) as CDHLOCLayerMBS` 783
- 3.67.41 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as color) as CDHLOCLayerMBS` 785
- 3.67.42 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS` 785
- 3.67.43 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer = -1) as CDHLOCLayerMBS` 787
- 3.67.45 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer = -1) as CDHLOCLayerMBS` 788
- 3.67.46 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS` 789

3.67.45 `addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer = -1) as CDHLOCLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-color high-low-open-close (HLOC) layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: ChartDirector supports drawing the HLOC symbol using different colors depending on whether the data represents an "up" day or "down" day. ChartDirector supports different definitions of what is an "up" day and "down" day, denoted using the following constants.

Constant	Value	Description
<code>HLOCDefault</code>	0	Do not distinguish between "up" and "down" days and use the same color for all HLOC symbols.
<code>HLOCOpenClose</code>	1	An up day is a day of which the closing value is on or above the opening value. This is the same definition commonly used in candlestick charts.
<code>HLOCUpDown</code>	2	An up day is a day of which the closing value is on or above the closing value of the previous day.

Return Value

A `HLOCLayer` object representing the HLOC layer created.

See also:

- 3.67.38 `addHLOCLayer as CDHLOCLayerMBS` 781

Parameter	Default	Description
highData	(Mandatory)	An array of numbers representing the high values. An empty array means there is no high value data available.
lowData	(Mandatory)	An array of numbers representing the low values. An empty array means there is no low value data available.
openData	(Mandatory)	An array of numbers representing the opening values. An empty array means there is no opening value data available.
closeData	(Mandatory)	An array of numbers representing the closing values. An empty array means there is no closing value data available.
upColor	(Mandatory)	The color to be used on an "up" day.
downColor	(Mandatory)	The color to be used on a "down" day.
colorMode	-1	The method used to determine if a day is an "up" or "down". Must be one of the constants in the above table. The default value of -1 means the colorMode is automatically determine to be either HLOCDefault or HLOCUpDown, depending on whether upColor and downColor are the same or different.
leadValue	[-Infinity]	The lead value to act as the closing pricing before the first day, so as to determine if the first day is an "up" or "down" day.

- 3.67.39 addHLOCLayer(highData() as Double, lowData() as Double) as CDHLOCLayerMBS 782
- 3.67.40 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double) as CDHLOCLayerMBS 783
- 3.67.41 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as color) as CDHLOCLayerMBS 785
- 3.67.42 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS 785
- 3.67.43 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer = -1) as CDHLOCLayerMBS 787
- 3.67.44 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 787
- 3.67.46 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 789

3.67.46 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a multi-color high-low-open-close (HLOC) layer to the XYChart, and specify the data sets to use for drawing the layer.

Notes: ChartDirector supports drawing the HLOC symbol using different colors depending on whether the data represents an "up" day or "down" day. ChartDirector supports different definitions of what is an "up" day and "down" day, denoted using the following constants.

Constant	Value	Description
HLOCDefault	0	Do not distinguish between "up" and "down" days and use the same color for all HLOC symbols.
HLOCOpenClose	1	An up day is a day of which the closing value is on or above the opening value. This is the same definition commonly used in candlestick charts.
HLOCUpDown	2	An up day is a day of which the closing value is on or above the closing value of the previous day.

Parameter	Default	Description
highData	(Mandatory)	An array of numbers representing the high values. An empty array means there is no high value data available.
lowData	(Mandatory)	An array of numbers representing the low values. An empty array means there is no low value data available.
openData	(Mandatory)	An array of numbers representing the opening values. An empty array means there is no opening value data available.
closeData	(Mandatory)	An array of numbers representing the closing values. An empty array means there is no closing value data available.
upColor	(Mandatory)	The color to be used on an "up" day.
downColor	(Mandatory)	The color to be used on a "down" day.
colorMode	-1	The method used to determine if a day is an "up" or "down". Must be one of the constants in the above table. The default value of -1 means the colorMode is automatically determine to be either HLOCDefault or HLOCUpDown, depending on whether upColor and downColor are the same or different.
leadValue	[-Infinity]	The lead value to act as the closing pricing before the first day, so as to determine if the first day is an "up" or "down" day.

Return Value

A HLOCLayer object representing the HLOC layer created.

See also:

- 3.67.38 addHLOCLayer as CDHLOCLayerMBS 781
- 3.67.39 addHLOCLayer(highData() as Double, lowData() as Double) as CDHLOCLayerMBS 782
- 3.67.40 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double) as CDHLOCLayerMBS 783
- 3.67.41 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as color) as CDHLOCLayerMBS 785

- 3.67.42 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, colorValue as Integer = -1) as CDHLOCLayerMBS 785
- 3.67.43 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer = -1) as CDHLOCLayerMBS 787
- 3.67.44 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as color, downColor as color, colorMode as Integer, leadValue as Double) as CDHLOCLayerMBS 787
- 3.67.45 addHLOCLayer(highData() as Double, lowData() as Double, openData() as Double, closeData() as Double, upColor as Integer, downColor as Integer, colorMode as Integer = -1) as CDHLOCLayerMBS 788

3.67.47 addInterLineLayer(line1 as CDLineObjMBS, line2 as CDLineObjMBS, color12 as color, color21 as color) as CDInterLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addInterLineLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.48 addInterLineLayer(line1 as CDLineObjMBS, line2 as CDLineObjMBS, color12 as Integer, color21 as Integer = -1) as CDInterLineLayerMBS 791

3.67.48 addInterLineLayer(line1 as CDLineObjMBS, line2 as CDLineObjMBS, color12 as Integer, color21 as Integer = -1) as CDInterLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an inter-line layer the XYChart. An inter-line layer is used to fill the region between any two lines (straight lines, spline curves, step lines, etc).

Notes: This method expects two "line" objects representing the two lines. The line objects can be obtained using LineLayer.getLine (inherited by SplineLayer and StepLineLayer), TrendLayer.getLine, and Mark.getLine.

Parameter	Default	Description
line1	(Mandatory)	An opaque line object representing the first line.
line2	(Mandatory)	An opaque line object representing the second line.
color12	(Mandatory)	The color used to fill the region of the lines when line1 > line2.
color21	-1	The color used to fill the region of the lines when line2 > line1. -1 means this color is the same as color12.

Return Value

An InterLineLayer object representing the inter-line layer created.

See also:

- 3.67.47 addInterLineLayer(line1 as CDLineObjMBS, line2 as CDLineObjMBS, color12 as color, color21 as color) as CDInterLineLayerMBS 791

3.67.49 addLineLayer(data as CDArrayMBS, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addLineLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.50 addLineLayer(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- 3.67.51 addLineLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793
- 3.67.52 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793
- 3.67.53 addLineLayer(dataCombineMethod as Integer = 0, depth as Integer = 0) as CDLineLayerMBS 794
- 3.67.54 addLineLayer(fastDB as CDDataAcceleratorMBS, seriesId as string, color as integer = -1, name as string = "") as CDLineLayerMBS 795

3.67.50 addLineLayer(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a line layer to the XYChart, and specify the data set to use for drawing the line.

Notes:

Parameter	Default	Description
data	(Mandatory)	An array of numbers representing the data set.
color	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
depth	0	The 3D depth of the line layer.

Return Value

A LineLayer object representing the line layer created.

See also:

- 3.67.49 addLineLayer(data as CDArrayMBS, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- 3.67.51 addLineLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793
- 3.67.52 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793
- 3.67.53 addLineLayer(dataCombineMethod as Integer = 0, depth as Integer = 0) as CDLineLayerMBS 794
- 3.67.54 addLineLayer(fastDB as CDDataAcceleratorMBS, seriesId as string, color as integer = -1, name as string = "") as CDLineLayerMBS 795

3.67.51 addLineLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addLineLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.49 addLineLayer(data as CDArrayMBS, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- 3.67.50 addLineLayer(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- 3.67.52 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793
- 3.67.53 addLineLayer(dataCombineMethod as Integer = 0, depth as Integer = 0) as CDLineLayerMBS 794
- 3.67.54 addLineLayer(fastDB as CDDataAcceleratorMBS, seriesId as string, color as integer = -1, name as string = "") as CDLineLayerMBS 795

3.67.52 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Parameter	Default	Description
data	(Mandatory)	An array of numbers representing the data set.
color	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
depth	0	The 3D depth of the line layer.

Function: Adds a line layer to the XYChart, and specify the data set to use for drawing the line.

Notes:

Return Value

A LineLayer object representing the line layer created.

See also:

- 3.67.49 `addLineLayer(data as CDArrayMBS, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS` 792
- 3.67.50 `addLineLayer(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS` 792
- 3.67.51 `addLineLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS` 793
- 3.67.53 `addLineLayer(dataCombineMethod as Integer = 0, depth as Integer = 0) as CDLineLayerMBS` 794
- 3.67.54 `addLineLayer(fastDB as CDDataAcceleratorMBS, seriesId as string, color as integer = -1, name as string = "") as CDLineLayerMBS` 795

3.67.53 `addLineLayer(dataCombineMethod as Integer = 0, depth as Integer = 0) as CDLineLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an empty line layer to the XYChart.

Notes: This method is typically used to add multiple data sets to a single line layer. First an empty line layer is created, then the data sets can be added using `Layer.addDataSet`.

Parameter	Default	Description
dataCombineMethod	Overlay	In this version of ChartDirector, this parameter is ignored.
depth	0	The 3D depth of the line layer.

Return Value

A LineLayer object representing the line layer created.

See also:

3.67. CLASS CDXYCHARTMBS 795

- 3.67.49 addLineLayer(data as CDArrayMBS, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- 3.67.50 addLineLayer(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- 3.67.51 addLineLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793
- 3.67.52 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793
- 3.67.54 addLineLayer(fastDB as CDDataAcceleratorMBS, seriesId as string, color as integer = -1, name as string = "") as CDLineLayerMBS 795

3.67.54 addLineLayer(fastDB as CDDataAcceleratorMBS, seriesId as string, color as integer = -1, name as string = "") as CDLineLayerMBS

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a line layer using data from a DataAccelerator.

Notes: The LineLayer created using this method will use only the data series provided by the DataAccelerator. It does not support adding more data sets, that is, Layer.addDataSet or Layer.setXData should not be used. Additional LineLayer objects should be used to plot other data series.

Argument	Default	Description
fastDB	(Mandatory)	The CDDataAcceleratorMBS object that provides the x and y data values.
seriesId	(Mandatory)	The id of the data series. The DataAccelerator will provide the x data series and the y data series with the specified id to the LineLayer.
color	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data series. The name will be used in the legend box, if one is available. An empty string means the data set has no name.

Returns a CDLineLayerMBS object representing the line layer created.

See also:

- 3.67.49 addLineLayer(data as CDArrayMBS, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- 3.67.50 addLineLayer(data as CDArrayMBS, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 792
- 3.67.51 addLineLayer(data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793
- 3.67.52 addLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDLineLayerMBS 793

- 3.67.53 `addLineLayer(dataCombineMethod as Integer = 0, depth as Integer = 0)` as `CDLineLayerMBS`
794

3.67.55 `addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1)` as `CDSscatterLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a scatter chart layer to the `XYChart`.

Notes: A scatter chart can be considered as a special configuration of a line chart, in the data symbols are enabled and the line width is set to zero. Therefore only the data symbols are visible and the chart appears as scattered.

Parameter	Default	Description
<code>xData</code>	(Mandatory)	An array of numbers representing the x values of the data points. If no explicit x coordinates are used in the chart (eg. using an enumerated x-axis), an empty array may be used for this argument.
<code>yData</code>	(Mandatory)	An array of numbers representing the y values of the data points.
<code>name</code>	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
<code>symbol</code>	<code>SquareShape</code>	One of the predefined symbol constants to specify the symbol to use. (See Shape Specification for available built-in shapes.)
<code>symbolSize</code>	5	The width and height of the symbol in pixels.
<code>fillColor</code>	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette.
<code>edgeColor</code>	-1	The edge color used to draw the edge of the symbol. The default value of -1 means using <code>LineColor</code> as the edge color.

Return Value

A `ScatterLayer` object representing the scatter layer created.

See also:

- 3.67.56 `addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color)` as `CDSscatterLayerMBS` 797
- 3.67.57 `addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1)` as `CDSscatterLayerMBS`
798
- 3.67.60 `addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color)` as `CDSscatterLayerMBS` 801
- 3.67.61 `addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1)` as `CDSscatterLayerMBS` 803

- 3.67.62 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 804
- 3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 804
- 3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 805
- 3.67.66 addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 808
- 3.67.67 addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 809
- 3.67.68 addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 810

3.67.56 addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addScatterLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796
- 3.67.57 addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 798
- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800
- 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- 3.67.61 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 803

- 3.67.62 `addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 804
- 3.67.63 `addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS` 804
- 3.67.65 `addScatterLayer(xDate() as Double, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 806
- 3.67.68 `addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 810

3.67.57 `addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS`

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a scatter chart layer to the XYChart.

Notes: A scatter chart can be considered as a special configuration of a line chart, in the data symbols are enabled and the line width is set to zero. Therefore only the data symbols are visible and the chart appears as scattered.

Parameter	Default	Description
<code>xData</code>	(Mandatory)	An array of numbers representing the x values of the data points. If no explicit x coordinates are used in the chart (eg. using an enumerated x-axis), an empty array may be used for this argument.
<code>yData</code>	(Mandatory)	An array of numbers representing the y values of the data points.
<code>name</code>	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
<code>symbol</code>	<code>SquareShape</code>	One of the predefined symbol constants to specify the symbol to use. (See Shape Specification for available built-in shapes.)
<code>symbolSize</code>	5	The width and height of the symbol in pixels.
<code>fillColor</code>	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette.
<code>edgeColor</code>	-1	The edge color used to draw the edge of the symbol. The default value of -1 means using <code>LineColor</code> as the edge color.

Return Value

A `ScatterLayer` object representing the scatter layer created.

See also:

- 3.67.55 `addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 796

- 3.67.56 addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 797
- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800
- 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- 3.67.62 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 804
- 3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 804
- 3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 805
- 3.67.66 addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 808
- 3.67.67 addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 809

3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Same as the other addScatterLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796
- 3.67.57 addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 798
- 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800

- 3.67.60 `addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 801
- 3.67.62 `addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 804
- 3.67.63 `addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS` 804
- 3.67.65 `addScatterLayer(xDate() as Double, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 806
- 3.67.66 `addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 808
- 3.67.67 `addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS` 809
- 3.67.68 `addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 810

3.67.59 `addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS`

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a scatter chart layer to the XYChart.

Notes: A scatter chart can be considered as a special configuration of a line chart, in the data symbols are enabled and the line width is set to zero. Therefore only the data symbols are visible and the chart appears as scattered.

Return Value

A ScatterLayer object representing the scatter layer created.

See also:

- 3.67.55 `addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 796
- 3.67.56 `addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 797
- 3.67.57 `addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 798

Parameter	Default	Description
xData	(Mandatory)	An array of numbers representing the x values of the data points. If no explicit x coordinates are used in the chart (eg. using an enumerated x-axis), an empty array may be used for this argument.
yData	(Mandatory)	An array of numbers representing the y values of the data points.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
symbol	SquareShape	One of the predefined symbol constants to specify the symbol to use. (See Shape Specification for available built-in shapes.)
symbolSize	5	The width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette.
edgeColor	-1	The edge color used to draw the edge of the symbol. The default value of -1 means using LineColor as the edge color.

- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 899
- 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- 3.67.61 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 803
- 3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 805
- 3.67.65 addScatterLayer(xDate() as Double, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 806
- 3.67.66 addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 808
- 3.67.68 addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 810

3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a scatter chart layer to the XYChart.

Notes: A scatter chart can be considered as a special configuration of a line chart, in the data symbols are enabled and the line width is set to zero. Therefore only the data symbols are visible and the chart appears

Parameter	Default	Description
xData	(Mandatory)	An array of numbers representing the x values of the data points. If no explicit x coordinates are used in the chart (eg. using an enumerated x-axis), an empty array may be used for this argument.
yData	(Mandatory)	An array of numbers representing the y values of the data points.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
symbol	SquareShape	One of the predefined symbol constants to specify the symbol to use. (See Shape Specification for available built-in shapes.)
symbolSize	5	The width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette.
edgeColor	-1	The edge color used to draw the edge of the symbol. The default value of -1 means using LineColor as the edge color.

as scattered.

Return Value

A ScatterLayer object representing the scatter layer created.

See also:

- 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796
- 3.67.56 addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 797
- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800
- 3.67.61 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 803
- 3.67.62 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 804
- 3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 804
- 3.67.66 addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 808

3.67. CLASS CDXYCHARTMBS 803

- 3.67.67 addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 809
- 3.67.68 addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 810

3.67.61 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds an area chart layer to the XYChart, and specify the data set to use for drawing the area.
See also:

- 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796
- 3.67.57 addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 798
- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800
- 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- 3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 804
- 3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 805
- 3.67.65 addScatterLayer(xDate() as Double, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 806
- 3.67.66 addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 808
- 3.67.67 addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 809

3.67.62 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a scatter chart layer to the XYChart.

See also:

- 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796
- 3.67.56 addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 797
- 3.67.57 addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 798
- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800
- 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- 3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 804
- 3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 805
- 3.67.67 addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 809
- 3.67.68 addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 810

3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a scatter chart layer to the XYChart.

See also:

- 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796
- 3.67.57 addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 798
- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800
- 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- 3.67.61 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 803
- 3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 805
- 3.67.65 addScatterLayer(xDate() as Double, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 806
- 3.67.66 addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 808
- 3.67.68 addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 810

3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a scatter chart layer to the XYChart.

See also:

- 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796

- 3.67.56 `addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 797
- 3.67.57 `addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 798
- 3.67.58 `addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 799
- 3.67.60 `addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 801
- 3.67.61 `addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS` 803
- 3.67.65 `addScatterLayer(xDate() as Double, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 806
- 3.67.66 `addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 808
- 3.67.67 `addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS` 809
- 3.67.68 `addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 810

3.67.65 `addScatterLayer(xDate() as Double, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS`

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a scatter chart layer to the XYChart.

Notes: A scatter chart can be considered as a special configuration of a line chart, in the data symbols are enabled and the line width is set to zero. Therefore only the data symbols are visible and the chart appears as scattered.

Return Value

A ScatterLayer object representing the scatter layer created.

See also:

- 3.67.55 `addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 796

Parameter	Default	Description
xData	(Mandatory)	An array of numbers representing the x values of the data points. If no explicit x coordinates are used in the chart (eg. using an enumerated x-axis), an empty array may be used for this argument.
yData	(Mandatory)	An array of numbers representing the y values of the data points.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
symbol	SquareShape	One of the predefined symbol constants to specify the symbol to use. (See Shape Specification for available built-in shapes.)
symbolSize	5	The width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette.
edgeColor	-1	The edge color used to draw the edge of the symbol. The default value of -1 means using LineColor as the edge color.

- 3.67.56 addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 797
- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- 3.67.61 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 803
- 3.67.62 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 804
- 3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 804
- 3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 805
- 3.67.67 addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 809
- 3.67.68 addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 810

3.67.66 addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a scatter chart layer to the XYChart.

Notes: A scatter chart can be considered as a special configuration of a line chart, in the data symbols are enabled and the line width is set to zero. Therefore only the data symbols are visible and the chart appears as scattered.

Parameter	Default	Description
xData	(Mandatory)	An array of numbers representing the x values of the data points. If no explicit x coordinates are used in the chart (eg. using an enumerated x-axis), an empty array may be used for this argument.
yData	(Mandatory)	An array of numbers representing the y values of the data points.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
symbol	SquareShape	One of the predefined symbol constants to specify the symbol to use. (See Shape Specification for available built-in shapes.)
symbolSize	5	The width and height of the symbol in pixels.
fillColor	-1	The color used to fill the symbol. -1 means that the color is automatically selected from the palette.
edgeColor	-1	The edge color used to draw the edge of the symbol. The default value of -1 means using LineColor as the edge color.

Return Value

A ScatterLayer object representing the scatter layer created.

See also:

- 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796
- 3.67.56 addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 797
- 3.67.57 addScatterLayer(xDate() as date, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 798
- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800

3.67. CLASS CDXYCHARTMBS 809

- 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- 3.67.62 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 804
- 3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 804
- 3.67.64 addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 805
- 3.67.67 addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 809

3.67.67 addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a scatter chart layer to the XYChart.

See also:

- 3.67.55 addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 796
- 3.67.58 addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 799
- 3.67.59 addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS 800
- 3.67.60 addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 801
- 3.67.61 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 803
- 3.67.62 addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS 804
- 3.67.63 addScatterLayer(xDate() as DateTime, yDate() as double, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS 804

- 3.67.64 `addScatterLayer(xDate() as DateTime, yDate() as double, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 805
- 3.67.66 `addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 808
- 3.67.68 `addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 810

3.67.68 `addScatterLayer(xDate() as double, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS`

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a scatter chart layer to the XYChart.

See also:

- 3.67.55 `addScatterLayer(xData() as Double, yData() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 796
- 3.67.56 `addScatterLayer(xData() as Double, yData() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 797
- 3.67.58 `addScatterLayer(xDate() as date, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 799
- 3.67.59 `addScatterLayer(xDate() as date, yDate() as Double, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 800
- 3.67.60 `addScatterLayer(xDate() as date, yDate() as Double, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 801
- 3.67.61 `addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS` 803
- 3.67.62 `addScatterLayer(xDate() as DateTime, yDate() as DateTime, name as string, symbol as integer, symbolSize as integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 804
- 3.67.65 `addScatterLayer(xDate() as Double, yDate() as date, name as string = "", symbol as Integer = 1, symbolSize as Integer = 5, fillColor as Integer = -1, edgeColor as Integer = -1) as CDScatterLayerMBS` 806
- 3.67.66 `addScatterLayer(xDate() as Double, yDate() as date, name as string, symbol as Integer, symbolSize as Integer, fillColor as color, edgeColor as color) as CDScatterLayerMBS` 808
- 3.67.67 `addScatterLayer(xDate() as double, yDate() as DateTime, name as string = "", symbol as integer = 1, symbolSize as integer = 5, fillColor as integer = -1, edgeColor as integer = -1) as CDScatterLayerMBS` 809

3.67.69 addSplineLayer as CDSplineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a spline layer to the XYChart.

Notes: In a spline layer, the data points are connected together using cardinal spline curves (as opposed to straight lines). The "tension" of the curve can be configured using SplineLayer.setTension.

Parameter	Default	Description
data	[Empty_Array]	An array of numbers representing the data set.
color	-1	The color to draw the spline line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.

Return Value

A SplineLayer object representing the spline layer created.

See also:

- 3.67.70 addSplineLayer(data() as Double, colorvalue as color, name as string = "") as CDSplineLayerMBS 811
- 3.67.71 addSplineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDSplineLayerMBS 811

3.67.70 addSplineLayer(data() as Double, colorvalue as color, name as string = "") as CDSplineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addSplineLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.69 addSplineLayer as CDSplineLayerMBS 811
- 3.67.71 addSplineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDSplineLayerMBS 811

3.67.71 addSplineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDSplineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a spline layer to the XYChart.

Notes: In a spline layer, the data points are connected together using cardinal spline curves (as opposed to

straight lines). The "tension" of the curve can be configured using `SplineLayer.setTension`.

Parameter	Default	Description
data	[Empty_Array]	An array of numbers representing the data set.
color	-1	The color to draw the spline line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.

Return Value

A `SplineLayer` object representing the spline layer created.

See also:

- 3.67.69 `addSplineLayer` as `CDSplineLayerMBS` 811
- 3.67.70 `addSplineLayer(data() as Double, colorvalue as color, name as string = "")` as `CDSplineLayerMBS` 811

3.67.72 `addStepLineLayer` as `CDStepLineLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a step line chart layer to the `XYChart`.

Notes: In a step line layer, the data points are connected together using steps. The alignment of the steps relative to the data points can be configured using `StepLineLayer.setAlignment`.

Parameter	Default	Description
data	[Empty_Array]	An array of numbers representing the data set.
color	-1	The color to draw the step line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.

Return Value

A `StepLineLayer` object representing the step line layer created.

See also:

- 3.67.73 `addStepLineLayer(data() as Double, colorvalue as color, name as string = "")` as `CDStepLineLayerMBS` 813
- 3.67.74 `addStepLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "")` as `CDStepLineLayerMBS` 813

3.67.73 addStepLineLayer(data() as Double, colorvalue as color, name as string = "") as CDStepLineLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addStepLineLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.72 addStepLineLayer as CDStepLineLayerMBS 812
- 3.67.74 addStepLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDStepLineLayerMBS 813

3.67.74 addStepLineLayer(data() as Double, colorvalue as Integer = -1, name as string = "") as CDStepLineLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a step line chart layer to the XYChart.

Notes: In a step line layer, the data points are connected together using steps. The alignment of the steps relative to the data points can be configured using StepLineLayer.setAlignment.

Parameter	Default	Description
data	[Empty_Array]	An array of numbers representing the data set.
color	-1	The color to draw the step line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.

Return Value

A StepLineLayer object representing the step line layer created.

See also:

- 3.67.72 addStepLineLayer as CDStepLineLayerMBS 812
- 3.67.73 addStepLineLayer(data() as Double, colorvalue as color, name as string = "") as CDStepLineLayerMBS 813

3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addTrendLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 814
- 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815
- 3.67.78 addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 816
- 3.67.79 addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 817
- 3.67.80 addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 818
- 3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 820
- 3.67.83 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 822
- 3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 823

3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a trend chart layer to the XYChart, and specify the data set to use for drawing the trend line.

Notes:

Return Value

A TrendLayer object representing the trend layer created.

See also:

- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813

Parameter	Default	Description
data	(Mandatory)	An array of numbers representing the data set.
color	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
depth	0	The 3D depth of the line layer.

- 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815
- 3.67.78 addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 816
- 3.67.80 addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 818
- 3.67.81 addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 819
- 3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 820
- 3.67.83 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 822
- 3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 823

3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a trend chart layer to the XYChart, and specify the data set to use for drawing the trend line.

Notes:

Return Value

A TrendLayer object representing the trend layer created.

See also:

Parameter	Default	Description
data	(Mandatory)	An array of numbers representing the data set.
color	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
depth	0	The 3D depth of the line layer.

- 3.67.76 `addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 814
- 3.67.78 `addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 816
- 3.67.79 `addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 817
- 3.67.80 `addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 818
- 3.67.81 `addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 819
- 3.67.82 `addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 820
- 3.67.83 `addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 821
- 3.67.84 `addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 821
- 3.67.85 `addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 822
- 3.67.86 `addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 823

3.67.78 `addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a trend chart layer to the XYChart, and specify the data set to use for drawing the trend line.

Notes:

Return Value

A TrendLayer object representing the trend layer created.

See also:

Parameter	Default	Description
data	(Mandatory)	An array of numbers representing the data set.
color	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
depth	0	The 3D depth of the line layer.

- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813
- 3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 814
- 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815
- 3.67.79 addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 817
- 3.67.81 addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 819
- 3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 820
- 3.67.83 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 822
- 3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 823

3.67.79 addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Same as the other addTrendLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813

- 3.67.76 `addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 814
- 3.67.77 `addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 815
- 3.67.78 `addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 816
- 3.67.80 `addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 818
- 3.67.81 `addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 819
- 3.67.82 `addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 820
- 3.67.83 `addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 821
- 3.67.85 `addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 822
- 3.67.86 `addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 823

3.67.80 `addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS`

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a trend chart layer to the XYChart, and specify the x and y values for drawing the trend line.

Notes:

Parameter	Default	Description
<code>xData</code>	(Mandatory)	An array of numbers representing the x values of the data points.
<code>yData</code>	(Mandatory)	An array of numbers representing the y values of the data points.
<code>color</code>	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
<code>name</code>	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
<code>depth</code>	0	The 3D depth of the line layer.

Return Value

A `TrendLayer` object representing the trend layer created.

See also:

- 3.67. CLASS CDXYCHARTMBS 819
- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813
 - 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815
 - 3.67.78 addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 816
 - 3.67.79 addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 817
 - 3.67.81 addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 819
 - 3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 820
 - 3.67.83 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
 - 3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
 - 3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 822
 - 3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 823

3.67.81 addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a trend chart layer to the XYChart, and specify the data set to use for drawing the trend line.

See also:

- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813
- 3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 814
- 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815
- 3.67.78 addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 816

- 3.67.79 addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 817
- 3.67.80 addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 818
- 3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 820
- 3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 822
- 3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 823

3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a trend chart layer to the XYChart, and specify the data set to use for drawing the trend line.

See also:

- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813
- 3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 814
- 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815
- 3.67.79 addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 817
- 3.67.80 addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 818
- 3.67.81 addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 819
- 3.67.83 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821

- 3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 822
- 3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 823

3.67.83 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a trend chart layer to the XYChart, and specify the data set to use for drawing the trend line.

See also:

- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813
- 3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 814
- 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815
- 3.67.78 addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 816
- 3.67.79 addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 817
- 3.67.81 addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 819
- 3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 820
- 3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 822
- 3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 823

3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a trend chart layer to the XYChart, and specify the data set to use for drawing the trend line.

See also:

- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813
- 3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 814
- 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815
- 3.67.78 addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 816
- 3.67.80 addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 818
- 3.67.81 addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 819
- 3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 820
- 3.67.83 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 822
- 3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 823

3.67.85 addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other addTrendLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813
- 3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 814
- 3.67.77 addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 815

3.67. CLASS CDXYCHARTMBS 823

- 3.67.78 addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 816
- 3.67.79 addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 817
- 3.67.80 addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 818
- 3.67.81 addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 819
- 3.67.82 addTrendLayer(dates() as DateTime, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 820
- 3.67.83 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821
- 3.67.84 addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS 821

3.67.86 addTrendLayer(xData() as Double, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a trend chart layer to the XYChart, and specify the x and y values for drawing the trend line.

Notes:

Parameter	Default	Description
xData	(Mandatory)	An array of numbers representing the x values of the data points.
yData	(Mandatory)	An array of numbers representing the y values of the data points.
color	-1	The color to draw the line. -1 means that the color is automatically selected from the palette.
name	""	The name of the data set. The name will be used in the legend box, if one is available. An empty string means the data set has no name.
depth	0	The 3D depth of the line layer.

Return Value

A TrendLayer object representing the trend layer created.

See also:

- 3.67.75 addTrendLayer(Data() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 813
- 3.67.76 addTrendLayer(Data() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS 814

- 3.67.77 `addTrendLayer(dates() as date, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 815
- 3.67.78 `addTrendLayer(dates() as date, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 816
- 3.67.79 `addTrendLayer(dates() as date, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 817
- 3.67.80 `addTrendLayer(dates() as date, yData() as Double, colorvalue as Integer = -1, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 818
- 3.67.81 `addTrendLayer(dates() as DateTime, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 819
- 3.67.83 `addTrendLayer(dates() as DateTime, yData() as double, colorvalue as color, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 821
- 3.67.84 `addTrendLayer(dates() as DateTime, yData() as double, colorvalue as integer = -1, name as string = "", depth as integer = 0) as CDTrendLayerMBS` 821
- 3.67.85 `addTrendLayer(xData() as Double, yData() as Double, colorvalue as color, name as string = "", depth as Integer = 0) as CDTrendLayerMBS` 822

3.67.87 `addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "") as CDVectorLayerMBS`

Plugin Version: 9.6, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Adds a vector layer to the chart.

Notes: The vectors are specified as 4 data series, representing the x and y coordinates of the reference points to put the vectors, and the lengths and directions of the vectors.

By default, the vector starts from the reference point and points away from it. You may use `VectorLayer.setArrowAlignment` to specify other options, such as for the vectors to point into the reference point, or to have the reference as a pivot at the mid-point of the vector.

ChartDirector supports specifying vectors lengths as pixels or in axis scale. The unit is specified by using the following predefined constants.

Constant	Value	Description
<code>PixelScale</code>	0	The unit is measured in pixels.
<code>XAxisScale</code>	1	The unit is measured in x-axis scale.
<code>YAxisScale</code>	2	The unit is measured in y-axis scale.

Parameter	Default	Description
xData	(Mandatory)	An array of numbers representing the x coordinates for the reference points of the vectors.
yData	(Mandatory)	An array of numbers representing the y coordinates for the reference points of the vectors.
directions	(Mandatory)	An array of numbers representing the direction of the vectors as a clockwise angle in degrees, where 0 is upward pointing direction.
lengths	(Mandatory)	An array of numbers representing the lengths of the vectors, in unit as specified in the lengthScale argument.
lengthScale	PixelScale	The unit for the lengths, which must be one of the predefined constants in the table above.
color	-1	The color to draw the data points. -1 means that the color is automatically selected from the palette.
name	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

Return Value

A VectorLayer object representing the vector layer created.

See also:

- 3.67.88 addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "") as CDVectorLayerMBS 825
- 3.67.89 addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer = 0, colorvalue as integer = -1, name as string = "") as CDVectorLayerMBS 826
- 3.67.90 addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer, colorvalue as color, name as string = "") as CDVectorLayerMBS 826
- 3.67.91 addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "") as CDVectorLayerMBS 827
- 3.67.92 addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "") as CDVectorLayerMBS 828

3.67.88 addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "") as CDVectorLayerMBS

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Same as the other addVectorLayer method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.87 `addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDVectorLayerMBS` 824
- 3.67.89 `addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer = 0, colorvalue as integer = -1, name as string = "")` as `CDVectorLayerMBS` 826
- 3.67.90 `addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 826
- 3.67.91 `addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDVectorLayerMBS` 827
- 3.67.92 `addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 828

3.67.89 `addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer = 0, colorvalue as integer = -1, name as string = "")` as `CDVectorLayerMBS`

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a vector layer to the chart.

See also:

- 3.67.87 `addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDVectorLayerMBS` 824
- 3.67.88 `addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 825
- 3.67.90 `addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 826
- 3.67.91 `addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDVectorLayerMBS` 827
- 3.67.92 `addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 828

3.67.90 `addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS`

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a vector layer to the chart.

See also:

- 3.67.87 `addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDVectorLayerMBS` 824
- 3.67.88 `addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 825
- 3.67.89 `addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer = 0, colorvalue as integer = -1, name as string = "")` as `CDVectorLayerMBS` 826
- 3.67.91 `addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDVectorLayerMBS` 827
- 3.67.92 `addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 828

3.67.91 `addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDVectorLayerMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adds a vector layer to the chart.

Notes: The vectors are specified as 4 data series, representing the x and y coordinates of the reference points to put the vectors, and the lengths and directions of the vectors.

By default, the vector starts from the reference point and points away from it. You may use `VectorLayer.setArrowAlignment` to specify other options, such as for the vectors to point into the reference point, or to have the reference as a pivot at the mid-point of the vector.

`ChartDirector` supports specifying vectors lengths as pixels or in axis scale. The unit is specified by using the following predefined constants.

Constant	Value	Description
<code>PixelScale</code>	0	The unit is measured in pixels.
<code>XAxisScale</code>	1	The unit is measured in x-axis scale.
<code>YAxisScale</code>	2	The unit is measured in y-axis scale.

Return Value

A `VectorLayer` object representing the vector layer created.

See also:

Parameter	Default	Description
xData	(Mandatory)	An array of numbers representing the x coordinates for the reference points of the vectors.
yData	(Mandatory)	An array of numbers representing the y coordinates for the reference points of the vectors.
directions	(Mandatory)	An array of numbers representing the direction of the vectors as a clockwise angle in degrees, where 0 is upward pointing direction.
lengths	(Mandatory)	An array of numbers representing the lengths of the vectors, in unit as specified in the lengthScale argument.
lengthScale	PixelScale	The unit for the lengths, which must be one of the predefined constants in the table above.
color	-1	The color to draw the data points. -1 means that the color is automatically selected from the palette.
name	""	The name of the layer. The name will be used in the legend box, if one is available. An empty string means the layer has no name.

- 3.67.87 `addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDVectorLayerMBS` 824
- 3.67.88 `addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 825
- 3.67.89 `addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer = 0, colorvalue as integer = -1, name as string = "")` as `CDVectorLayerMBS` 826
- 3.67.90 `addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 826
- 3.67.92 `addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS` 828

3.67.92 `addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "")` as `CDVectorLayerMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `addVectorLayer` method, but uses `color` instead of `integer` data type for passing color values.

See also:

- 3.67.87 `addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "")` as `CDVectorLayerMBS` 824

- 3.67.88 addVectorLayer(dates() as date, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer, colorvalue as color, name as string = "") as CDVectorLayerMBS 825
- 3.67.89 addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer = 0, colorvalue as integer = -1, name as string = "") as CDVectorLayerMBS 826
- 3.67.90 addVectorLayer(dates() as DateTime, yData() as double, directions() as double, lengths() as double, lengthScale as integer, colorvalue as color, name as string = "") as CDVectorLayerMBS 826
- 3.67.91 addVectorLayer(xData() as Double, yData() as Double, directions() as Double, lengths() as Double, lengthScale as Integer = 0, colorvalue as Integer = -1, name as string = "") as CDVectorLayerMBS 827

3.67.93 Constructor(width as Integer = 640, height as Integer = 480, bgcolor as Integer = &hFFFF0000, edgeColor as Integer = &hFF000000, raisedEffect as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new XYChart object.

Notes:

Parameter	Default	Description
width	(Mandatory)	The width of the chart in pixels.
height	(Mandatory)	The height of the chart in pixels.
bgColor	BackgroundColor	The background color of the chart.
edgeColor	Transparent	The edge color of the chart.
raisedEffect	0	The 3D border width. For positive values, the border will appear raised. For negative values, the border will appear depressed. A zero value means the border will appear flat.

See also:

- 3.67.94 Constructor(width as Integer, height as Integer, bgcolor as color, edgeColor as color, raisedEffect as Integer = 0) 829

3.67.94 Constructor(width as Integer, height as Integer, bgcolor as color, edgeColor as color, raisedEffect as Integer = 0)

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other Constructor method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.93 Constructor(width as Integer = 640, height as Integer = 480, bgcolor as Integer = &hFFFF0000, edgeColor as Integer = &hFF000000, raisedEffect as Integer = 0) 829

3.67.95 `getLayer(i as Integer) as CDLayerMBS`

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets a Layer object based on the order in which it is being added to the chart.

Notes:

Argument	Default	Description
i	(Mandatory)	The index of the data layer. The index of the first layer added to the chart is 0. The index of the Nth layer added to the chart is N - 1.

3.67.96 `getLayerByZ(i as Integer) as CDLayerMBS`

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets a Layer object based on the order in which it is being drawn.

Notes: In ChartDirector, by default, the first layer added will stay on the top of the chart. The second layer will be added under the first layer, and so on. It means The layers will be drawn in the reverse order that they are added to the chart. The ordering can be configured with `Layer.moveFront` or `Layer.moveBack`.

Argument	Default	Description
zIndex	(Mandatory)	The z-index of the required layer. The z-index of the first layer drawn is 0. The z-index for the Nth layer drawn is N - 1.

3.67.97 `getLayerCount as Integer`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the number of layers in the chart.

3.67.98 `getNearestXValue(xCoor as Double) as Double`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x data value that is nearest to the specified x pixel coordinate.

Notes: This method will search all x data values in the XYChart to look for the x data value that is nearest to the given x coordinate. If there are two x data values equally near to the specified x pixel coordinate, this method will arbitrarily return one of the values.

Returns the x data value that is nearest to the specified x coordinate.

Argument	Default	Description
xCoor	(Mandatory)	The x pixel coordinate to search for.

3.67.99 getPlotArea as CDPlotAreaMBS

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the PlotArea object representing the plot area.

3.67.100 getXCoor(value as Double) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x pixel coordinate of a point given the x data value.

Notes: Note: You must call BaseChart.layout first before calling this method. It is because ChartDirector needs to perform auto-scaling and determine the axis scale first before it can compute the coordinates.

For a 3D chart, this method will get the pixel coordinate of the data value on the top surface of the chart. Use Layer.getXCoor to obtain the pixel coordinate of the data value on a particular chart layer.

Parameter	Default	Description
v	(Mandatory)	The x data value.

Return Value

The x coordinate of the x data value.

3.67.101 getXValue(xCoor as Integer) as Double

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the x data value given the x pixel coordinate.

Notes: Note: This method should be used only after ChartDirector has output the chart image, or after XYChart.layoutAxes, BaseChart.layout or XYChart.packPlotArea has been called. ChartDirector needs to perform auto-scaling and layout the axis before it can convert between pixel coordinates and data values.

Argument	Default	Description
xCoor	(Mandatory)	The x pixel coordinate.

Returns the x data value at the x pixel coordinate.

3.67.102 `getYCoor(value as Double, yAxis as CDAxisMBS=nil)` as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y pixel coordinate of a point given the y data value.

Notes: Note: You must call `BaseChart.layout` first before calling this method. It is because `ChartDirector` needs to perform auto-scaling and determine the axis scale first before it can compute the coordinates.

For a 3D chart, this method will get the pixel coordinate of the data value on the top surface of the chart. Use `Layer.getYCoor` to obtain the pixel coordinate of the data value on a particular chart layer.

Parameter	Default	Description
<code>v</code>	(Mandatory)	The y data value.
<code>yAxis</code>	<code>nil</code>	The y-axis to use to determine the pixel coordinates of data values. The y-axis may be obtained using <code>XYChart.yAxis</code> , <code>XYChart.yAxis2</code> or <code>XYChart.addAxis</code> . The default is to use the primary y-axis.

Return Value

The y coordinate of the y data value.

3.67.103 `getYValue(yCoor as Integer, axis as CDAxisMBS = nil) as Double`

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the y data value given the y pixel coordinate.

Notes: Note: This method should be used only after `ChartDirector` has output the chart image, or after `XYChart.layoutAxes`, `BaseChart.layout` or `XYChart.packPlotArea` has been called. `ChartDirector` needs to perform auto-scaling and layout the axis before it can convert between pixel coordinates and data values.

Argument	Default	Description
<code>yCoor</code>	(Mandatory)	The y pixel coordinate.
<code>yAxis</code>	nil	The y-axis to use to determine the pixel coordinates of data values. The y-axis may be obtained using <code>XYChart.yAxis</code> , <code>XYChart.yAxis2</code> or <code>XYChart.addAxis</code> . The default is to use the primary y-axis.

Returns the y data value at the y pixel coordinate.

3.67.104 `layoutAxes`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Performs axis auto-scaling and sets the axis positions and lengths.

Notes: `ChartDirector` automatically calls this method when creating the chart output (eg. using `BaseChart.makeChart`). There is usually no need to call this method explicitly.

However, if you would like to draw custom objects whose positions or contents depend on the axis scales, you may call this method to auto-scaling the axes before creating the chart output.

The axes will auto-scale based on the data at the time of calling this method. New data added afterwards will not affect the axis scale.

You should not modify the axis scale (eg. using `Axis.setLinearScale`, `Axis.setLabels`, etc) after calling this method.

3.67.105 `packPlotArea(leftX as Integer, topY as Integer, rightX as Integer, bottomY as Integer, minWidth as Integer = 0, minHeight as Integer = 0)`

Plugin Version: 9.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Adjusts the plot area size and position to fit the plot area and the axis labels in a bounding box.

Notes: In some charts, the axis labels may vary widely and can be very short or very long. It may be difficult to determine to plot area position and size so as to reserve reasonable and sufficient space for the axis labels.

This method adjusts the size and position of the plot area, so that the plot area, together with the axis labels can fit within a given bounding box.

Note that this method only adjusts for the thickness of the primary and secondary axes, and assumes they are at the border of the plot area. For other axes, you may use `Axis.getThickness` to get their thickness and adjust the plot area size and position accordingly.

For this method to determine the thickness of the axes, it needs to auto-scale the axes to determine the axis labels. That means all the data should be entered to the chart before calling this method.

Furthermore, the labels generated by auto-scaling depend on the number of labels that can fit on the axis, which in turn depends on the plot area size. So even the final plot area size is determined by this method, the `XYChart.setPlotArea` must still be used to set a reasonable initial plot area size.

To adjust only the plot area width and horizontal position, and leave the height and vertical position unchanged, simply use a bounding box with zero height (eg. set both `topY` and `bottomY` to 0). Similarly, to adjust only the height and vertical position, use a bounding box with zero width.

In some extreme cases, the axis labels may be so long that the plot area needs to adjust to an unreasonably small size, or even zero in size. For example, the axis labels may be names entered by the user, and the user may enter a name thousands of characters long. These extreme cases should be avoided by checking and limiting the length of the labels before passing the labels to `ChartDirector`. This method also has a safeguard for these unreasonable labels by ensuring the plot area will not shrink below a given minimum size.

Arguments:

Argument	Default	Description
<code>leftX</code>	(Mandatory)	The left x coordinate of the bounding box.
<code>topY</code>	(Mandatory)	The top y coordinate of the bounding box.
<code>rightX</code>	(Mandatory)	The right x coordinate of the bounding box.
<code>bottomY</code>	(Mandatory)	The bottom y coordinate of the bounding box.
<code>minWidth</code>	0	The minimum width the plot area can adjust to.
<code>minHeight</code>	0	The minimum height the plot area can adjust to.

3.67.106 `setAxisAtOrigin(originMode as Integer = 3, symmetryMode as Integer = 0)`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the chart into 4 quadrant mode. The axes may intersect at origin inside the plot area (as opposed to at the border of the plot area).

Notes: By default, `ChartDirector` puts the axes at the border of the plot area. This ensures the axes and the axes labels will not block the contents of the plot area.

However, in some cases, it may be desirable for the axes to intersect at the origin. The axes may need to move inside the plot area.

ChartDirector allows you to move the primary x-axis and/or y-axis to so that they intersect with the zero point of the other axis. The following constants represent different possibilities:

Constant	Value	Description
XAxisAtOrigin	1	Move the x-axis so that it will intersect with the zero point of the y-axis, if that point exists.
YAxisAtOrigin	2	Move the y-axis so that it will intersect with the zero point of the x-axis, if that point exists.
XYAxisAtOrigin	3	Move the x-axis so that it will intersect with the zero point of the y-axis, if that point exists. Move the y-axis so that it will intersect with the zero point of the x-axis, if that point exists.

In addition to ensure the axes intersects at the origin, ChartDirector allows you to control if the axes are symmetrical about the origin with the following flags. Multiple flags can be specified by "or" them together.

Constant	Value	Description
XAxisSymmetric	1	Adjust the x-axis so that it becomes symmetrical about the zero point, that is, the positive and negative part of the x-axis will be of the same length.
XAxisSymmetricIfNeeded	2	Adjust the x-axis will be symmetrical about the zero point if the axis scale needs both positive and negative values.
YAxisSymmetric	4	Adjust the y-axis so that it becomes symmetrical about the zero point, that is, the positive and negative part of the y-axis will be of the same length.
YAxisSymmetricIfNeeded	8	Adjust the y-axis will be symmetrical about the zero point if the axis scale needs both positive and negative values.
XYAxisSymmetric	16	Adjust the x-axes and y-axes so that they become symmetrical about the zero point. The positive and negative parts of the x-axes and the y-axes will all be of the same length.
XYAxisSymmetricIfNeeded	32	Adjust the x-axes and y-axes so that they become symmetrical about the zero point. However, if an axis does not need both positive and negative values, it will not be adjusted.

Parameter	Default	Description
originMode	XYAxisAtOrigin	predefined constants to determine which axis needs to move to intersect with the zero point of the other axis.
symmetryMode	0	Flags to determine if the axes need to be symmetrical about the origin.

3.67.107 setClipping(margin as Integer = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables clipping on the plot area.

Notes: In normal usage, ChartDirector will auto-scale the axis to ensure all the data points will be within

the plot area.

However, if manual scaling is used (e.g. using `Axis.setLinearScale`), it is possible to choose an axis scale such that some data points will be outside the plot area. So some of the data representation (bars, lines, etc) will be outside the plot area.

This method can be used to clip the plot area, so that any data representation that is outside the plot area will not be drawn.

Parameter	Default	Description
<code>margin</code>	0	Additional margin, measured in pixels, to be added to the 4 sides of the plot area for the purpose of clipping. A positive value means the clipping region will be larger than the plot area. A negative value means the clipping region will be smaller than the plot area. The default value of 0 means the clipping region is equal to the plot area.

3.67.108 `setPlotArea(x as Integer, y as Integer, width as Integer, height as Integer, bgColor as color, altBgColor as color, edgeColor as color, hGridColor as color, vGridColor as color) as CDPlotAreaMBS`

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other `setPlotArea` method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.109 `setPlotArea(x as Integer, y as Integer, width as Integer, height as Integer, bgColor as Integer = &hff000000, altBgColor as Integer = -1, edgeColor as Integer = -1, hGridColor as Integer = &hc0c0c0, vGridColor as Integer = &hff000000) as CDPlotAreaMBS` 836

3.67.109 `setPlotArea(x as Integer, y as Integer, width as Integer, height as Integer, bgColor as Integer = &hff000000, altBgColor as Integer = -1, edgeColor as Integer = -1, hGridColor as Integer = &hc0c0c0, vGridColor as Integer = &hff000000) as CDPlotAreaMBS`

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the position, size, background colors, edge color and grid colors of the plot area.

Notes:

Return Value

A `PlotArea` object representing the plot area.

See also:

Parameter	Default	Description
x	(Mandatory)	The x coordinate of the left of the plot area.
y	(Mandatory)	The y coordinate of the top of the plot area.
width	(Mandatory)	The width of the plot area in pixels.
height	(Mandatory)	The height of the plot area in pixels.
bgColor	Transparent	The background color of the plot area.
altBgColor	-1	The second background color of the plot area. -1 means there is no second background color. If there is a second background color, the two background colors will be used alternatively as horizontal bands on the background grid.
edgeColor	-1	The border color of the plot area. -1 means to use the default, which is LineColor. However, if the axes are configured in 4 quadrant mode (see XYChart.setAxisAtOrigin), the default will change to Transparent.
hGridColor	C0C0C0	The horizontal grid color.
vGridColor	Transparent	The vertical grid color.

- 3.67.108 setPlotArea(x as Integer, y as Integer, width as Integer, height as Integer, bgColor as color, altBgColor as color, edgeColor as color, hGridColor as color, vGridColor as color) as CDPlotAreaMBS
836

3.67.110 setTrimData(startPos as Integer, len as Integer = &h7ffffff)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Uses only a subset of the data series to draw charts.

Notes: In finance charts, "lead data" are commonly needed to plot technical indicators. For example, to show a 20-days moving average line for 30 days, one needs 50 days of data. It is because computing 20-days moving average requires 20 days of "lead data". The setTrimData method is a convenience method for trimming off these "lead data" so they will not appear on the chart.

Parameter	Default	Description
startPos	(Mandatory)	The index for the first data position to use for plotting charts.
len	7FFFFFFF	The length of the data points used for plotting charts.

3.67.111 setXAxisOnTop(value as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Interchange the position of the primary x-axis and the secondary x-axis.

Notes: By default, the primary x-axis is the x-axis on the bottom side of the plot area (left side if XYChart.swapXY is in effect), and the secondary x-axis is on the top side of the plot area (right side if XYChart.swapXY is in effect). This method can be used to interchange their positions.

Parameter	Default	Description
b	true	A true value means to interchange the positions of the primary and secondary x-axes. A false value means using the default positions for the primary and secondary x-axes.

3.67.112 setYAxisOnRight(value as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Interchange the position of the primary y-axis and the secondary y-axis.

Notes: By default, the primary y-axis is the y-axis on the left side of the plot area (bottom side if XY-Chart.swapXY is in effect), and the secondary y-axis is on the right side of the plot area (top side if XYChart.swapXY is in effect). This method can be used to interchange their positions.

Parameter	Default	Description
b	true	A true value means to interchange the positions of the primary and secondary y-axes. A false value means using the default positions for the primary and secondary y-axes.

3.67.113 swapXY(value as boolean=true)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Swap the position of the x and y axes, so the x-axes will become vertical, and the y-axes will become horizontal.

Notes: After swapping the x and y axes, the charts will appear rotated. For example, the bars in a bar chart will become horizontal instead of vertical, and the area in a stacked area chart will grow from left to right (instead of bottom to top).

Parameter	Default	Description
b	true	A true value means the x and y axes will be swapped. A false value means the x and y axes not be swapped.

3.67.114 syncYAxis(slope as Double = 1, intercept as Double = 0)

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets a linear formula to synchronize the secondary y-axis scale to the primary y-axis scale.

Notes: The formula is in the format:

$$y2 = y1 * \text{slope} + \text{intercept}$$

This method is usually used if the two y-axes represent the same measurement using different units. Examples including temperature in Celsius and in Fahrenheit, and length in meters and feet.

Parameter	Default	Description
slope	1	The slope parameter for the formula linking the secondary y-axis to the primary y-axis.
intercept	0	The intercept parameter for the formula linking the secondary y-axis to the primary y-axis.

3.67.115 xScaleColor(scale() as double) as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an x-scale color. An x-scale is a color that will vary depending on the x value of the pixel.

Notes:

Argument	Default	Description
colorStops	(Mandatory)	An array that defines the mapping from data values to colors. Please refer to ColorAxis.setColorScale for the format of this array.

Returns a 32-bit integer representing the x scale color.

3.67.116 xZoneColor(threshold as Double, belowColor as color, aboveColor as color) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other xZoneColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.117 xZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer) as Integer 839

3.67.117 xZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a x-zone color. A x-zone will change from one color to another depending on a threshold value on the x-axis.

Notes: For example, if a x-zone color is used as the line color in a line layer, the line will switch from one color to another when it passes through a certain value on the x-axis. Similarly, if a x-zone color is used as the fill color in an area layer, the area will switch from one color to another when it passes through a certain value on the x-axis.

The two colors used in a x-zone color can be other dynamic colors. For example, one color could be a solid color, while the other color could be a dash line color (see !BaseChart.dashLineColor). When this x-zone color is as the line color, the line will change from a solid style to a dash line style when the line passes through a certain value on the x-axis

You may create x-zone colors with more than 2 zones by cascading multiple x-zone colors.

For a 3D chart, this method will get the zone color at the top surface of the chart. Use Layer.xZoneColor to obtain the zone color for a particular chart layer.

threshold	(Mandatory)	The x value serving as the threshold for switching between two colors.
belowColor	(Mandatory)	The color to use when the x-axis value of the pixel is smaller than the threshold.
aboveColor	(Mandatory)	The color to use when the x-axis value of the pixel is greater than the threshold.

Return Value

A 32-bit integer representing the x-zone color.

See also:

- 3.67.116 xZoneColor(threshold as Double, belowColor as color, aboveColor as color) as Integer 839

3.67.118 yScaleColor(scale() as double, yAxis as CDAxisMBS = nil) as Integer

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a y-scale color. An y-scale is a color that will vary depending on the y value of the pixel.

Notes:

Argument	Default	Description
colorStops	(Mandatory)	An array that defines the mapping from data values to colors. Please refer to ColorAxis.setColorScale for the format of this array.
yAxis	nil	The y-axis to use to determine the pixel coordinates of data values. The y-axis may be obtained using XYChart.yAxis, XYChart.yAxis2 or XYChart.addAxis. The default is to use the primary y-axis.

Returns a 32-bit integer representing the y scale color.

3.67.119 yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as CDAxisMBS=nil) as Integer

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Same as the other yZoneColor method, but uses color instead of integer data type for passing color values.

See also:

- 3.67.120 yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as CDAxisMBS=nil) as Integer 841

3.67.120 yZoneColor(threshold as Double, belowColor as Integer, aboveColor as Integer, yAxis as CDAxisMBS=nil) as Integer

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a x-zone color. A x-zone will change from one color to another depending on a threshold value on the x-axis.

Notes: For example, if a x-zone color is used as the line color in a line layer, the line will switch from one color to another when it passes through a certain value on the x-axis. Similarly, if a x-zone color is used as the fill color in an area layer, the area will switch from one color to another when it passes through a certain value on the x-axis.

The two colors used in a x-zone color can be other dynamic colors. For example, one color could be a solid color, while the other color could be a dash line color (see !BaseChart.dashLineColor). When this x-zone color is as the line color, the line will change from a solid style to a dash line style when the line passes through a certain value on the x-axis

You may create x-zone colors with more than 2 zones by cascading multiple x-zone colors.

For a 3D chart, this method will get the zone color at the top surface of the chart. Use Layer.xZoneColor to obtain the zone color for a particular chart layer.

Parameter	Default	Description
threshold	(Mandatory)	The x value serving as the threshold for switching between two colors.
belowColor	(Mandatory)	The color to use when the x-axis value of the pixel is smaller than the threshold.
aboveColor	(Mandatory)	The color to use when the x-axis value of the pixel is greater than the threshold.

Return Value

A 32-bit integer representing the x-zone color.

See also:

- 3.67.119 yZoneColor(threshold as Double, belowColor as color, aboveColor as color, yAxis as CDx-

isMBS=nil) as Integer

841

3.67.121 Properties

3.67.122 xAxis as CDAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the Axis object representing primary x-axis of the XYChart.

Notes: By default, the primary x-axis is the x-axis on the bottom side of the plot area (left side if XYChart.swapXY is in effect), and the secondary x-axis is on the top side of the plot area (right side if XYChart.swapXY is in effect). You may interchange their positions using XYChart.setXAxisOnTop.

Return Value

The Axis object representing the primary x-axis of the XYChart.

(Read only property)

3.67.123 xAxis2 as CDAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the Axis object representing secondary x-axis of the XYChart.

Notes: By default, the primary x-axis is the x-axis on the bottom side of the plot area (left side if XYChart.swapXY is in effect), and the secondary x-axis is on the top side of the plot area (right side if XYChart.swapXY is in effect). You may interchange their positions using XYChart.setXAxisOnTop.

Return Value

The Axis object representing the secondary x-axis of the XYChart.

(Read only property)

3.67.124 yAxis as CDAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the Axis object representing primary y-axis of the XYChart.

Notes: Return Value

The Axis object representing the primary y-axis of the XYChart.

(Read only property)

3.67.125 yAxis2 as CDAxisMBS

Plugin Version: 8.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves the Axis object representing secondary y-axis of the XYChart.

Notes: By default, the primary y-axis is the y-axis on the left side of the plot area (bottom side if XYChart.swapXY is in effect), and the secondary y-axis is on the right side of the plot area (top side if XYChart.swapXY is in effect). You may interchange their positions using XYChart.setYAxisOnRight.

Return Value

The Axis object representing the secondary y-axis of the XYChart.

(Read only property)

Chapter 4

List of Questions in the FAQ

- 5.0.1 Can anyone help me convert seconds to time in this format hh:mm:ss? 855
- 5.0.2 Do you have plugins for Android? 856
- 5.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 856
- 5.0.4 How to catch delete key? 857
- 5.0.5 How to convert cmyk to rgb? 858
- 5.0.6 How to delete a folder? 859
- 5.0.7 How to detect if CPU is 64bit processor? 860
- 5.0.8 How to query variant type string for a variant? 861
- 5.0.9 How to refresh a htmlviewer on Windows? 862
- 5.0.10 Is there an example for vector graphics in Xojo? 863
- 5.0.11 Picture functions do not preserve resolution values? 864
- 5.0.12 A toolbox call needs a rect - how do I give it one? 864
- 5.0.13 API client not supported? 864
- 5.0.14 Can I access Access Database with Java classes? 865
- 5.0.15 Can I create PDF from Xojo Report using DynaPDF? 866
- 5.0.16 Can I use AppleScripts in a web application? 866
- 5.0.17 Can I use graphics class with DynaPDF? 866
- 5.0.18 Can I use sockets on a web application? 867
- 5.0.19 Can I use your ChartDirector plugin on a web application? 867

- 5.0.20 Can I use your DynaPDF plugin on a web application? 868
- 5.0.21 Can I use your plugin controls on a web application? 869
- 5.0.22 Can you get an unique machine ID? 869
- 5.0.23 ChartDirector: Alignment Specification 869
- 5.0.24 ChartDirector: Color Specification 870
- 5.0.25 ChartDirector: Font Specification 873
- 5.0.26 ChartDirector: Mark Up Language 877
- 5.0.27 ChartDirector: Parameter Substitution and Formatting 881
- 5.0.28 ChartDirector: Shape Specification 885
- 5.0.29 Copy styled text? 886
- 5.0.30 Do you have code to validate a credit card number? 887
- 5.0.31 Do you have plugins for X-Rite EyeOne, eXact or i1Pro? 888
- 5.0.32 Does SQL Plugin handle stored procedures with multiple result sets? 888
- 5.0.33 Does the plugin home home? 888
- 5.0.34 folderitem.absolutepath is limited to 255 chars. How can I get longer ones? 889
- 5.0.35 Has anyone played round with using CoreImage to do things like add dissolve transitions say when changing from one tab to another within a window? 889
- 5.0.36 How about Plugin support for older OS X? 890
- 5.0.37 How can I detect whether an Intel CPU is a 64bit CPU? 891
- 5.0.38 How can I disable the close box of a window on Windows? 892
- 5.0.39 How can I get all the environment variables from Windows? 892
- 5.0.40 How can i get similar behavior to Roxio Toast or iTunes where clicking a 'burn' button allows the next inserted blank CD-R to bypass the Finder and be accepted by my application? 893
- 5.0.41 How can I get text from a PDF? 893
- 5.0.42 How can I get text from a Word Document? 893
- 5.0.43 How can I get the item string for a given file creator? 894
- 5.0.44 How can I launch an app using it's creator code? 895
- 5.0.45 How can I learn what shared libraries are required by a plugin on Linux? 895
- 5.0.46 How can I validate an email address? 897
- 5.0.47 How do I decode correctly an email subject? 897

	847
• 5.0.48 How do I enable/disable a single tab in a tabpanel?	898
• 5.0.49 How do I find the root volume for a file?	899
• 5.0.50 How do I get the current languages list?	899
• 5.0.51 How do I get the Mac OS Version?	900
• 5.0.52 How do I get the printer name?	901
• 5.0.53 How do I make a metal window if RB does not allow me this?	902
• 5.0.54 How do I make a smooth color transition?	902
• 5.0.55 How do I read the applications in the dock app?	903
• 5.0.56 How do I truncate a file?	904
• 5.0.57 How do update a Finder's windows after changing some files?	904
• 5.0.58 How to access a USB device directly?	905
• 5.0.59 How to add icon to file on Mac?	905
• 5.0.60 How to ask the Mac for the Name of the Machine?	905
• 5.0.61 How to automatically enable retina in my apps?	906
• 5.0.62 How to avoid leaks with Cocoa functions?	906
• 5.0.63 How to avoid trouble connecting to oracle database with SQL Plugin?	907
• 5.0.64 How to avoid ___NSAutoreleaseNoPool console messages in threads?	907
• 5.0.65 How to bring app to front?	908
• 5.0.66 How to bring my application to front?	908
• 5.0.67 How to catch Control-C on Mac or Linux in a console app?	909
• 5.0.68 How to change name of application menu?	909
• 5.0.69 How to change the name in the menubar of my app on Mac OS X?	910
• 5.0.70 How to check if a folder/directory has subfolders?	910
• 5.0.71 How to check if Macbook runs on battery or AC power?	911
• 5.0.72 How to check if Microsoft Outlook is installed?	912
• 5.0.73 How to check on Mac OS which country or language is currently selected?	912
• 5.0.74 How to code sign my app with plugins?	913
• 5.0.75 How to collapse a window?	913
• 5.0.76 How to compare two pictures?	914

- 5.0.77 How to compile PHP library? 916
- 5.0.78 How to convert a `BrowserType` to a `String` with `WebSession.Browser`? 917
- 5.0.79 How to convert a `EngineType` to a `String` with `WebSession.Engine`? 918
- 5.0.80 How to convert a `PlatformType` to a `String` with `WebSession.Platform`? 918
- 5.0.81 How to convert a text to iso-8859-1 using the `TextEncoder`? 919
- 5.0.82 How to convert `ChartTime` back to Xojo date? 920
- 5.0.83 How to convert line endings in text files? 920
- 5.0.84 How to convert picture to string and back? 921
- 5.0.85 How to copy an array? 922
- 5.0.86 How to copy an dictionary? 922
- 5.0.87 How to copy parts of a movie to another one? 922
- 5.0.88 How to create a birthday like calendar event? 923
- 5.0.89 How to create a GUID? 924
- 5.0.90 How to create a Mac picture clip file? 924
- 5.0.91 How to create a PDF file in Xojo? 925
- 5.0.92 How to create `EmailAttachment` for PDF Data in memory? 925
- 5.0.93 How to create PDF for image files? 926
- 5.0.94 How to CURL Options translate to Plugin Calls? 927
- 5.0.95 How to delete file with ftp and curl plugin? 928
- 5.0.96 How to detect display resolution changed? 928
- 5.0.97 How to detect retina? 929
- 5.0.98 How to disable force quit? 929
- 5.0.99 How to disable the error dialogs from Internet Explorer on javascript errors? 929
- 5.0.100 How to display a PDF file in Xojo? 929
- 5.0.101 How to do a lottery in RB? 930
- 5.0.102 How to do an asycron DNS lookup? 931
- 5.0.103 How to draw a dashed pattern line? 931
- 5.0.104 How to draw a nice antialiased line? 932
- 5.0.105 How to dump java class interface? 933

	849
• 5.0.106 How to duplicate a picture with mask or alpha channel?	934
• 5.0.107 How to enable assistive devices?	935
• 5.0.108 How to encrypt a file with Blowfish?	935
• 5.0.109 How to extract text from HTML?	936
• 5.0.110 How to find empty folders in a folder?	936
• 5.0.111 How to find iTunes on a Mac OS X machine fast?	936
• 5.0.112 How to find network interface for a socket by it's name?	937
• 5.0.113 How to find version of Microsoft Word?	938
• 5.0.114 How to fix CURL error 60/53 on connecting to server?	939
• 5.0.115 How to format double with n digits?	939
• 5.0.116 How to get a time converted to user time zone in a web app?	940
• 5.0.117 How to get an handle to the frontmost window on Windows?	940
• 5.0.118 How to get CFAbsoluteTime from date?	941
• 5.0.119 How to get client IP address on web app?	941
• 5.0.120 How to get fonts to load in charts on Linux?	941
• 5.0.121 How to get fonts to load in DynaPDF on Linux?	942
• 5.0.122 How to get GMT time and back?	943
• 5.0.123 How to get good crash reports?	943
• 5.0.124 How to get list of all threads?	944
• 5.0.125 How to get parameters from webpage URL in Xojo Web Edition?	944
• 5.0.126 How to get the color for disabled textcolor?	944
• 5.0.127 How to get the current free stack space?	945
• 5.0.128 How to get the current timezone?	946
• 5.0.129 How to get the current window title?	947
• 5.0.130 How to get the cursor blink interval time?	948
• 5.0.131 How to get the list of the current selected files in the Finder?	949
• 5.0.132 How to get the Mac OS system version?	950
• 5.0.133 How to get the Mac OS Version using System.Gestalt?	950
• 5.0.134 How to get the screensize excluding the task bar?	951

- 5.0.135 How to get the size of the frontmost window on Windows? 951
- 5.0.136 How to get the source code of a HTMLViewer? 952
- 5.0.137 How to get Xojo apps running Linux? 952
- 5.0.138 How to handle really huge images with GraphicsMagick or ImageMagick? 952
- 5.0.139 How to handle tab key for editable cells in listbox? 953
- 5.0.140 How to hard link MapKit framework? 954
- 5.0.141 How to have a PDF downloaded to the user in a web application? 955
- 5.0.142 How to hide all applications except mine? 955
- 5.0.143 How to hide script errors in HTMLViewer on Windows? 956
- 5.0.144 How to hide the grid/background/border in ChartDirector? 956
- 5.0.145 How to hide the mouse cursor on Mac? 956
- 5.0.146 How to insert image to NSTextView or TextArea? 956
- 5.0.147 How to jump to an anchor in a htmlviewer? 957
- 5.0.148 How to keep a movieplayer unclickable? 957
- 5.0.149 How to keep my web app from using 100% CPU time? 958
- 5.0.150 How to kill a process by name? 958
- 5.0.151 How to know how many CPUs are present? 959
- 5.0.152 How to know the calling function? 959
- 5.0.153 How to launch an app using it's creator code? 960
- 5.0.154 How to launch disc utility? 960
- 5.0.155 How to make a lot of changes to a REAL SQL Database faster? 961
- 5.0.156 How to make a NSImage object for my retina enabled app? 961
- 5.0.157 How to make a window borderless on Windows? 961
- 5.0.158 How to make an alias using AppleEvents? 962
- 5.0.159 How to make AppleScripts much faster? 963
- 5.0.160 How to make double clicks on a canvas? 963
- 5.0.161 How to make my Mac not sleeping? 965
- 5.0.162 How to make my own registration code scheme? 966
- 5.0.163 How to make small controls on Mac OS X? 966

	851
• 5.0.164 How to mark my Mac app as background only?	967
• 5.0.165 How to move a file or folder to trash?	967
• 5.0.166 How to move an application to the front using the creator code?	968
• 5.0.167 How to move file with ftp and curl plugin?	969
• 5.0.168 How to normalize string on Mac?	969
• 5.0.169 How to obscure the mouse cursor on Mac?	970
• 5.0.170 How to open icon file on Mac?	970
• 5.0.171 How to open PDF in acrobat reader?	970
• 5.0.172 How to open printer preferences on Mac?	971
• 5.0.173 How to open special characters panel on Mac?	972
• 5.0.174 How to optimize picture loading in Web Edition?	972
• 5.0.175 How to parse XML?	972
• 5.0.176 How to play audio in a web app?	973
• 5.0.177 How to pretty print xml?	974
• 5.0.178 How to print to PDF?	974
• 5.0.179 How to query Spotlight's Last Open Date for a file?	975
• 5.0.180 How to quit windows?	976
• 5.0.181 How to read a CSV file correctly?	976
• 5.0.182 How to read the command line on windows?	977
• 5.0.183 How to render PDF pages with PDF Kit?	977
• 5.0.184 How to restart a Mac?	978
• 5.0.185 How to resume ftp upload with curl plugin?	978
• 5.0.186 How to rotate a PDF page with CoreGraphics?	979
• 5.0.187 How to rotate image with CoreImage?	980
• 5.0.188 How to run a 32 bit application on a 64 bit Linux?	981
• 5.0.189 How to save HTMLViewer to PDF with landscape orientation?	981
• 5.0.190 How to save RTFD?	981
• 5.0.191 How to save RTFD?	982
• 5.0.192 How to scale a picture proportionally with mask?	982

- 5.0.193 How to scale a picture proportionally? 983
- 5.0.194 How to scale/resize a CImageMBS? 984
- 5.0.195 How to scale/resize a picture? 985
- 5.0.196 How to search with regex and use unicode codepoints? 985
- 5.0.197 How to see if a file is invisible for Mac OS X? 986
- 5.0.198 How to set cache size for SQLite or REALSQLDatabase? 987
- 5.0.199 How to set the modified dot in the window? 987
- 5.0.200 How to show a PDF file to the user in a Web Application? 987
- 5.0.201 How to show Keyboard Viewer programmatically? 988
- 5.0.202 How to show the mouse cursor on Mac? 989
- 5.0.203 How to shutdown a Mac? 989
- 5.0.204 How to sleep a Mac? 990
- 5.0.205 How to speed up rasterizer for displaying PDFs with DynaPDF? 990
- 5.0.206 How to use PDFLib in my RB application? 990
- 5.0.207 How to use quotes in a string? 991
- 5.0.208 How to use Sybase in Web App? 991
- 5.0.209 How to use the Application Support folder? 991
- 5.0.210 How to use the IOPMCopyScheduledPowerEvents function in Xojo? 992
- 5.0.211 How to validate a GUID? 995
- 5.0.212 How to walk a folder hierarchie non recursively? 995
- 5.0.213 I got this error: PropVal, QDPictMBS.Name (property value), Type mismatch error. Expected CGDataProviderMBS, but got Variant, Name:QDPictMBS 996
- 5.0.214 I registered the MBS Plugins in my application, but later the registration dialog is shown. 996
- 5.0.215 I want to accept Drag & Drop from iTunes 997
- 5.0.216 I'm drawing into a listbox but don't see something. 999
- 5.0.217 I'm searching for a method or so to move a window from position x.y to somewhere else on the screen. 999
- 5.0.218 If I use one of your plug-ins under windows, would this then impose the use of dll after compilation or my would my compiled soft still be a stand-alone single file software? 999
- 5.0.219 Is the fn key on a powerbook keyboard down? 1000
- 5.0.220 Is there a case sensitive Dictionary? 1000

- 5.0.221 Is there a way to use the MBS plugin to get only the visible item and folder count on a volume?
1001
- 5.0.222 Is there an easy way I can launch the Displays preferences panel? 1001
- 5.0.223 List of Windows Error codes? 1002
- 5.0.224 Midi latency on Windows problem? 1002
- 5.0.225 My Xojo Web App does not launch. Why? 1002
- 5.0.226 SQLiteDatabase not initialized error? 1003
- 5.0.227 Textconverter returns only the first x characters. Why? 1003
- 5.0.228 The type translation between CoreFoundation/Foundation and Xojo data types. 1004
- 5.0.229 Uploaded my web app with FTP, but it does not run on the server! 1006
- 5.0.230 What classes to use for hotkeys? 1006
- 5.0.231 What do I need for Linux to get picture functions working? 1006
- 5.0.232 What does the NAN code mean? 1007
- 5.0.233 What font is used as a 'small font' in typical Mac OS X apps? 1007
- 5.0.234 What is last plugin version to run on Mac OS X 10.4? 1008
- 5.0.235 What is last plugin version to run on PPC? 1008
- 5.0.236 What is last version of the plugins for macOS 32-bit? 1009
- 5.0.237 What is the difference between Timer and WebTimer? 1009
- 5.0.238 What is the list of Excel functions? 1009
- 5.0.239 What is the replacement for PluginMBS? 1010
- 5.0.240 What to do on Xojo reporting a conflict? 1010
- 5.0.241 What to do with a NSImageCacheException? 1011
- 5.0.242 What to do with MySQL Error 2014? 1011
- 5.0.243 What to do with SQL Plugin reporting Malformed string as error? 1011
- 5.0.244 Where is CGGetActiveDisplayListMBS? 1011
- 5.0.245 Where is CGGetDisplaysWithPointMBS? 1012
- 5.0.246 Where is CGGetDisplaysWithRectMBS? 1012
- 5.0.247 Where is CGGetOnlineDisplayListMBS? 1012
- 5.0.248 Where is GetObjectClassNameMBS? 1012
- 5.0.249 Where is NetworkAvailableMBS? 1012

- 5.0.250 Where is StringHeight function in DynaPDF? 1013
- 5.0.251 Where is XLSDocumentMBS class? 1013
- 5.0.252 Where to get information about file formats? 1013
- 5.0.253 Where to register creator code for my application? 1014
- 5.0.254 Which Mac OS X frameworks are 64bit only? 1014
- 5.0.255 Which plugins are 64bit only? 1015
- 5.0.256 Why application doesn't launch because of a missing ddraw.dll!? 1015
- 5.0.257 Why application doesn't launch because of a missing shlwapi.dll!? 1015
- 5.0.258 Why do I hear a beep on keydown? 1015
- 5.0.259 Why does folderitem.item return nil? 1015
- 5.0.260 Why doesn't showurl work? 1015
- 5.0.261 Why don't the picture functions not work on Linux? 1016
- 5.0.262 Why have I no values in my chart? 1016
- 5.0.263 Will application size increase with using plugins? 1016
- 5.0.264 XLS: Custom format string guidelines 1016
- 5.0.265 Xojo doesn't work with your plugins on Windows 98. 1017
- 5.0.266 Xojo or my RB application itself crashes on launch on Mac OS Classic. Why? 1018

Chapter 5

The FAQ

5.0.1 Can anyone help me convert seconds to time in this format hh:mm:ss?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Sure, here's a routine I use (which has an advantage over the previously-posted Date-based solution in that you don't have to rely on the creation of an object – all that happens is some division and string concatenation):

Example:

```
Function SecsToTimeString(timeInSecs as Integer, padHours as boolean, padMinutes as boolean) as string
// Given an amount time (in seconds), generates a string representing that amount
// of time. The padHours and padMinutes parameters determine whether to display
// hours and minutes if their values are zero.
```

```
// Examples:
// timeInSecs = 90, padHours = true; returns "00:01:30"
// timeInSecs = 1, padHours = false, padMinutes = true; returns "00:01"
// timeInSecs = 3601, padMinutes = false; returns "01:00:01"
```

```
dim hours, minutes, seconds as Integer
dim hoursString, minutesString as string
```

```
hours = timeInSecs / 3600
minutes = (timeInSecs mod 3600) / 60
seconds = timeInSecs mod 60
```

```
if hours = 0 then
if padHours then
hoursString = "00:"
else
hoursString = ""
end if
```

```

else
hoursString = Format(hours, "##\:")
end if
if minutes = 0 then
if hours <>0 or padMinutes then
minutesString = "00:"
else
minutesString = ""
end if
else
minutesString = Format(minutes, "00\:")
end if

return hoursString + minutesString + Format(seconds, "00")
End Function

```

Notes: (from the rb mailinglist)

5.0.2 Do you have plugins for Android?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Since there is no plugin SDK for Android, we have no way to make a plugin for Android.

Notes: We support macOS, Windows, Linux and iOS.

5.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use functions from NSColor to get proper highlight color in RGB:

Example:

```

Function ProperHighlightColor(active as Boolean) As Color
#if TargetCocoa
Dim theColor As NSColorMBS
If active Then
theColor = NSColorMBS.alternateSelectedControlColor
Else
theColor = NSColorMBS.secondarySelectedControlColor
End If

```

```

Dim rgbColor As NSColorMBS = theColor.colorUsingColorSpaceName(NSColorSpaceMBS.NSCalibrate-

```

```

dRGBColorSpace)
If rgbColor <>Nil Then
Dim red as Integer = rgbColor.redComponent * 255.0
Dim green as Integer = rgbColor.greenComponent * 255.0
Dim blue as Integer = rgbColor.blueComponent * 255.0
Return RGB(red, green, blue)
Else
Return HighlightColor
End If
#else
return HighlightColor
#endif
End Function

```

Notes: As you see we convert color to Calibrated RGB for best results.
See also:

- 5.0.4 How to catch delete key? 857
- 5.0.5 How to convert cmyk to rgb? 858
- 5.0.6 How to delete a folder? 859
- 5.0.7 How to detect if CPU is 64bit processor? 860
- 5.0.8 How to query variant type string for a variant? 861
- 5.0.9 How to refresh a htmlviewer on Windows? 862

5.0.4 How to catch delete key?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The following is the code in keydown event catches delete or backspace keys.

Example:

```

Function KeyDown(Key As String) As Boolean
if asc(key) = 8 or asc(key) = 127 then
MsgBox "Delete"
Return true
end if
End Function

```

See also:

- 5.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 856

- 5.0.5 How to convert cmyk to rgb? 858
- 5.0.6 How to delete a folder? 859
- 5.0.7 How to detect if CPU is 64bit processor? 860
- 5.0.8 How to query variant type string for a variant? 861
- 5.0.9 How to refresh a htmlviewer on Windows? 862

5.0.5 How to convert cmyk to rgb?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

The following is the code to convert cmyk values to an RGB color datatype.

It's just a basic estimate of the color values. If you are looking for completely color accurate solution, this is not it. It should work for most people. :)

Example:

Function CMYKToRGB(c as Integer, m as Integer, y as Integer, k as Integer) As color

// converts c,m,y,k values (0-100) to color data type RGB

// place this in a method. Supply C,M,Y,K values-

// it returns color datatype

```
dim color_RGB as color
```

```
dim r, g, b as Integer
```

```
r=255-round(2.55*(c+k))
```

```
if r<0 then
```

```
  r=0
```

```
end if
```

```
g=255-round(2.55*(m+k))
```

```
if g<0 then
```

```
  g=0
```

```
end if
```

```
b=255-round(2.55*(y+k))
```

```
if b<0 then
```

```
  b=0
```

```
end if
```

```
color_RGB=RGB(r,g,b)
```

```
return color_RGB
```

```
End Function
```

Notes:

(from the rb mailinglist)
See also:

- 5.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 856
- 5.0.4 How to catch delete key? 857
- 5.0.6 How to delete a folder? 859
- 5.0.7 How to detect if CPU is 64bit processor? 860
- 5.0.8 How to query variant type string for a variant? 861
- 5.0.9 How to refresh a htmlviewer on Windows? 862

5.0.6 How to delete a folder?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The following is the code that deletes a folder recursively.

Example:

```
Sub deletefolder(f as folderitem)
dim files(-1) as FolderItem

if f=nil then Return

// delete single file
if f.Directory=false then
f.Delete
Return
end if

// get a list of all items in that folder
dim i,c as Integer
c=F.Count
for i=1 to c
files.Append f.TrueItem(i)
next

// delete each item
for each fo as FolderItem in files
if fo=nil then
' ignore
elseif fo.Directory then
deletefolder fo
fo.delete
else ' file
```

```
fo.Delete
end if
next
```

```
f.Delete
End Sub
```

See also:

- 5.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 856
- 5.0.4 How to catch delete key? 857
- 5.0.5 How to convert cmyk to rgb? 858
- 5.0.7 How to detect if CPU is 64bit processor? 860
- 5.0.8 How to query variant type string for a variant? 861
- 5.0.9 How to refresh a htmlviewer on Windows? 862

5.0.7 How to detect if CPU is 64bit processor?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Via CPUID you can ask CPU:

Example:

```
dim c as new CPUIDMBS

if c.Flags(CPUIDMBS.kFeatureLM) then
MsgBox "64-bit CPU"
else
MsgBox "32-bit CPU"
end if
```

Notes: Should work on all intel compatible CPUs.

See also:

- 5.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 856
- 5.0.4 How to catch delete key? 857
- 5.0.5 How to convert cmyk to rgb? 858
- 5.0.6 How to delete a folder? 859
- 5.0.8 How to query variant type string for a variant? 861
- 5.0.9 How to refresh a htmlviewer on Windows? 862

5.0.8 How to query variant type string for a variant?

Plugin Version: 20.5, Platforms: macOS, Linux, Windows.

Answer: The following example function returns type string for variant.

Example:

```
Public Function VariantTypeString(v as Variant) as string
// Xojo's VarType doesn't know Unsigned integers
'Dim type As Integer = VarType(v)

// MBS VarType can detect unsigned integer
Dim type As Integer = GetVariantTypeMBS(v)

Dim IsArray As Boolean = BitwiseAnd(type, Variant.TypeArray) = Variant.TypeArray

// type without array
type = BitwiseAnd(type, Bitwise.OnesComplement(Variant.TypeArray))

// build a dictionary to map types on first call
Static TypeMap As Dictionary
If TypeMap = Nil Then
TypeMap = New Dictionary
TypeMap.Value(Variant.TypeBoolean) = "Boolean"
TypeMap.Value(Variant.TypeCFStringRef) = "CFStringRef"
TypeMap.Value(Variant.TypeColor) = "Color"
TypeMap.Value(Variant.TypeCString) = "CString"
TypeMap.Value(Variant.TypeCurrency) = "Currency"
TypeMap.Value(Variant.TypeDate) = "Date"
TypeMap.Value(Variant.TypeDateTime) = "DateTime"
TypeMap.Value(Variant.TypeDouble) = "Double"
TypeMap.Value(Variant.TypeInt32) = "Int32"
TypeMap.Value(Variant.TypeInt64) = "Int64"
TypeMap.Value(Variant.TypeInteger) = "Integer"
TypeMap.Value(Variant.TypeNil) = "Nil"
TypeMap.Value(Variant.TypeObject) = "Object"
TypeMap.Value(Variant.TypeOSType) = "OSType"
TypeMap.Value(Variant.TypePString) = "PString"
TypeMap.Value(Variant.TypePtr) = "Ptr"
TypeMap.Value(Variant.TypeSingle) = "Single"
TypeMap.Value(Variant.TypeString) = "String"
TypeMap.Value(Variant.TypeStructure) = "Structure"
TypeMap.Value(Variant.TypeText) = "Text"
TypeMap.Value(Variant.TypeWindowPtr) = "WindowPtr"
TypeMap.Value(Variant.TypeWString) = "WString"

// MBS extra types
TypeMap.Value(Variant.TypeInt32+100) = "UInt32"
TypeMap.Value(Variant.TypeInt64+100) = "UInt64"
```

End If

```
// lookup type

#if DebugBuild then
If Not TypeMap.HasKey(type) Then
Break // missing type
End If
#endif

If IsArray Then
Return "Array of " + TypeMap.Lookup(type,"?")
Else
Return TypeMap.Lookup(type,"?")
End If
End Function
```

See also:

- 5.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 856
- 5.0.4 How to catch delete key? 857
- 5.0.5 How to convert cmyk to rgb? 858
- 5.0.6 How to delete a folder? 859
- 5.0.7 How to detect if CPU is 64bit processor? 860
- 5.0.9 How to refresh a htmlviewer on Windows? 862

5.0.9 How to refresh a htmlviewer on Windows?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can ask the browser to reload the website with this code line:

Example:

```
call htmlViewer1.IERunJavaScriptMBS("javascript:document.location.reload()")
```

See also:

- 5.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 856
- 5.0.4 How to catch delete key? 857
- 5.0.5 How to convert cmyk to rgb? 858

- 5.0.6 How to delete a folder? 859
- 5.0.7 How to detect if CPU is 64bit processor? 860
- 5.0.8 How to query variant type string for a variant? 861

5.0.10 Is there an example for vector graphics in Xojo?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this example inside the paint event of a window:

Example:

```
dim v as Group2D
dim r as RectShape
dim s as StringShape
```

```
const pi=3.14
```

```
s=new StringShape
s.Text="Hello World!"
s.TextFont="Geneva"
s.TextSize=24
s.FillColor=rgb(0,0,255)
s.Italic=true
s.y=5
s.x=0
```

```
r=new RectShape
```

```
r.X=0
r.y=0
r.Height=100
r.Width=180
r.BorderColor=rgb(255,0,0)
r.FillColor=rgb(0,255,0)
r.BorderWidth=5
r.Border=50
```

```
v=new Group2d
v.Append r
v.Append s
v.Rotation=pi*-20.0/180.0
v.x=150
v.y=150
```

```
g.DrawObject v
```

5.0.11 Picture functions do not preserve resolution values?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, the picture functions return pictures with no/default resolution values.

Example:

```
dim l as Picture = LogoMBS(500)
```

```
l.HorizontalResolution = 300
```

```
l.VerticalResolution = 300
```

```
dim r as Picture = l.Rotate90MBS
```

```
MsgBox str(r.HorizontalResolution)+" x "+str(r.VerticalResolution)
```

```
r.HorizontalResolution = l.HorizontalResolution
```

```
r.VerticalResolution = l.VerticalResolution
```

```
MsgBox str(r.HorizontalResolution)+" x "+str(r.VerticalResolution)
```

Notes: So please fix them yourself after calling a function.

Maybe in the future this changes, but currently you can't really set this easily from plugin code.

5.0.12 A toolbox call needs a rect - how do I give it one?

Plugin Version: all, Platforms: macOS, Windows.

Answer: Fill a memoryblock like this:

Example:

```
Dim MB As Memoryblock
```

```
MB = NewMemoryBlock(8)
```

```
MB.Short(0) = window1.Top
```

```
MB.Short(2) = window1.Left
```

```
MB.Short(4) = window1.Height+window1.Top // bottom
```

```
MB.Short(6) = window1.Width+window1.Left // right
```

5.0.13 API client not supported?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you get this exception message on `SQLConnectionMBS.Connect`, we may have a problem.

Notes: First case is that the given thing is not supported (e.g. MS SQL directly on Mac).

Second case is that the plugin compilation went wrong and the support for the database was not linked into the plugin. Like MySQL missing or MS SQL on Windows missing. In that case please contact us to fix the plugin.

5.0.14 Can I access Access Database with Java classes?

Plugin Version: all, Platform: Windows.

Answer: You can use `ucanaccess` to access databases created with Microsoft

Example:

```

dim options(-1) as string

// load all the jar files we have in a folder called java:

dim appFolder as FolderItem = GetFolderItem("")

Dim count as Integer = appFolder.Parent.Child("java").Count
dim libjs() as string
For i as Integer = 1 to count
Dim f As FolderItem = appFolder.Parent.Child("java").item(i)
If f <> Nil and f.Exists Then
libjs.append f.NativePath+";"
End If
Next

// now init virtual machine
dim library as string = Join(libjs, "")
dim vm as new JavaVMMBS(library)

if vm.Handle = 0 then
MsgBox "Failed to initialize virtual machine"
else
// now make a new database connection with ucanaccess
dim d as new JavaDatabaseMBS(vm,"net.ucanaccess.jdbc.UcanaccessDriver")
Dim DbFile as FolderItem = appFolder.Parent.Child("Database11.accdb")
dim j as JavaConnectionMBS = d.getConnection("jdbc:ucanaccess://" + DbFile.NativePath)

// select and show values
dim r as JavaResultSetMBS = j.MySelectSQL("Select * From test")
while r.NextRecord
MsgBox r.getString("FirstName") + " " + r.getString("LastName")
wend

end if

```

Exception e as JavaExceptionMBS
 MsgBox e.message+" **errorcode:** "+str(e.ErrorNumber)

Notes: see website:
<http://ucanaccess.sourceforge.net/site.html>

5.0.15 Can I create PDF from Xojo Report using DynaPDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, we have a graphics class integration for DynaPDF.

Notes: Since MBS Plugin in version 19.2, we can integrate reports with Xojo.

5.0.16 Can I use AppleScripts in a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, but they run on the server, not on the client.

Example:

```
dim a as new AppleScriptMBS

// query my application name
a.Compile "tell application ""System Events"" to return name of current application"

// run
a.Execute

// show result
label1.text = a.Result

// shows something like "My Application.fcgi.debug"
```

Notes: This can be useful to control the server from remote, if and only if the your sever is running Mac OS X.

5.0.17 Can I use graphics class with DynaPDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Sorry, no. We can't provide a graphics subclass from plugin.

Notes: This is a feature request to allow graphics subclasses:
Feedback case 11391: [feedback://showreport?report_id=11391](https://feedback.apple.com/feedback/showreport?report_id=11391)

5.0.18 Can I use sockets on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, but they run on the server, not on the client.

Notes: You can use `HTTPSocket`, `SMTPSocket`, `POP3Socket`, `SMTPSecureSocket`, `SecurePOP3Socket`, `EasyTCPSocket`, `EasyUDPSocket`, `AutoDiscovery`, our Bonjour classes or our `CURL*` classes. But all of them work on the server, not on the client.

This means if you search for a printer with Bonjour, you can find the printers in the local network on your server hosting site. Using `SMTPSocket` may be a good idea for sending emails from the server like notifications.

5.0.19 Can I use your ChartDirector plugin on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, our ChartDirector plugin works just fine on the Xojo Web Edition.

Example:

```
// The data for the pie chart
dim data(-1) as Double=array(55.0, 18.0, 25.0, 22.0, 18.0, 30.0, 35.0)

// The labels for the pie chart, Words are chosen random to check font!
dim labels(-1) as string=array("Germany", "Italy", "France", "Spain", "UK", "Poland", "Russia")

// The colors to use for the sectors
dim colors(-1) as Integer

colors.Append &h66aaee
colors.Append &heebb22
colors.Append &hbbsbbb
colors.Append &h8844ff

if TargetLinux then
  CDBaseChartMBS.SetFontSearchPath "/usr/share/fonts/truetype/msttcorefonts"
end if

// Create a PieChart object of size 360 x 300 pixels
dim c as new CDPieChartMBS(700, 600)
```

```

c.setBackground(c.linearGradientColor(0, 0, 0, c.getHeight(), &h0000cc, &h000044))
c.setRoundedFrame(&hffffff, 16)
dim tt as CDTextBoxMBS = c.addTitle("ChartDirector Demonstration", "timesbi.ttf", 18)
tt.setMargin(0, 0, 16, 0)
tt.setFontColor(&hFFFFFF)

// Set the center of the pie at (180, 140) and the radius to 100 pixels
c.setPieSize 350,300,150
// Set the sector colors
c.setColors(c.kDataColor, colors)

// Draw the pie in 3D with a pie thickness of 20 pixels
c.set3D(20)

dim t as CDTextBoxMBS = c.setLabelStyle("arialbd.ttf", 10, &h000000)
t.setBackground(CDPieChartMBS.kSameAsMainColor, CDPieChartMBS.kTransparent, CDPieChartMBS.soft-
Lighting(CDPieChartMBS.kRight, 0))
t.setRoundedCorners(8)

// Use local gradient shading for the sectors, with 5 pixels wide
// semi-transparent white (bbffffff) borders
c.setSectorStyle(CDPieChartMBS.kLocalGradientShading, &hbbffffff, 0)

// Set the pie data and the pie labels
c.setData data,labels
call c.setLabelStyle "arialbd.ttf",18

dim pic as picture = c.makeChartPicture
dim wp as new WebPicture(pic, Picture.FormatJPEG) // JPEG makes it smaller and faster

ImageView1.Picture=wp

```

Notes: Be aware that our plugin produces pictures for you, which you assign to ImageViews. Transferring those pictures takes time, so you can optimize that with using WebPicture class. There you can decide between different compressions to improve speed (use JPEG instead of PNG).

e.g. if you use ubuntu, you can install the ttf-mscorefonts-installer package and call this method with `"/usr/share/fonts/truetype/msttcorefonts"` as the path. No backslash on the end of a path, please.

5.0.20 Can I use your DynaPDF plugin on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, our DynaPDF plugin works just fine on the Xojo Web Edition.

Notes: PDF files are created on the server. You may want to offer a preview to the user which uses reduced resolution images to reduce the time to download the PDF.

See our Create PDF example for the Xojo Web Edition.

5.0.21 Can I use your plugin controls on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: No.

5.0.22 Can you get an unique machine ID?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: There is nothing like an unique machine ID.

Notes: 1:

You can use the MAC IDs of the network interfaces.

This can be changed by the user with software tools.

And the list of network interfaces changes if user reorder the interfaces.

2:

You can use the system folder creation date/time.

This may stay equal after cloning machines or after migration to new PC.

3:

You can use the Mac Serialnumber.

Mac only and it can happen that a Mac does not have a serial number.

4:

You can use the x86 CPU ID.

This is x86 CPU only and does not avoid running on the same CPU in different PCs.

5.0.23 ChartDirector: Alignment Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Alignment Specification

Notes: In many ChartDirector objects, you may specify the alignment of the object's content relative to its boundary. For example, for a TextBox object, you may specify the text's alignment relative to the box boundary by using TextBox.setAlignment.

The ChartDirector API defines several constants for the alignment options.

ConstantValueDescription

BottomLeft	1	The leftmost point on the bottom line.
BottomCenter	2	The center point on the bottom line.
BottomRight	3	The rightmost point on the bottom line.
Left	4	The leftmost point on the middle horizontal line.
Center	5	The center point on the middle horizontal line.
Right	6	The rightmost point on the middle horizontal line.
TopLeft	7	The leftmost point on the top line.
TopCenter	8	The center point on the top line.
TopRight	9	The rightmost point on the top line.
Bottom	2	The center point on the bottom line. Same as BottomCenter.
Top	8	The center point on the top line. Same as TopCenter.
TopLeft2	10	An alternative top-left position used in Axis.setTitlePos for axis title positioning only. For a vertical axis, TopLeft2 refers to refers to the left of the top side, while TopLeft refers to the top of the left side. The reverse applies for a horizontal axis.
TopRight2	11	An alternative top-right position used in Axis.setTitlePos for axis title positioning only. For a vertical axis, TopRight2 refers to refers to the right of the top side, while TopRight refers to the top of the right side. The reverse applies for a horizontal axis.
BottomLeft2	12	An alternative bottom-left position used in Axis.setTitlePos for axis title positioning only. For a vertical axis, BottomLeft2 refers to refers to the left of the bottom side, while BottomLeft refers to the bottom of the left side. The reverse applies for a horizontal axis.
BottomRight2	13	An alternative bottom-right position used in Axis.setTitlePos for axis title positioning only. For a vertical axis, BottomRight2 refers to refers to the right of the bottom side, while BottomRight refers to the bottom of the right side. The reverse applies for a horizontal axis.

5.0.24 ChartDirector: Color Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Color Specification

Notes: Many functions in the ChartDirector API accept colors as parameters. ChartDirector supports col-

ors specified in web and HTML compatible ARGB format, in which ARGB refers to the Alpha transparency, Red, Green and Blue components of the color.

In addition to ARGB colors, ChartDirector supports "dynamic" colors. A dynamic color is a color that changes depending on the position of the pixels. The "dynamic" colors that ChartDirector supports include "pattern colors", "metal colors", "gradient colors", "zone colors" and "dash line colors".

ChartDirector supports specifying colors indirectly using "palette colors". When a "palette color" is used, the color is specified as an index to a palette. The actual color is looked up from the palette. ARGB Color ARGB color consists of 4 components - alpha transparency, red, green and blue. The four components are encoded as a 32-bit number, with each component occupying 8 bits. In hexadecimal notation, it is AAR-RGGBB, where AA, RR, GG and BB are the alpha transparency, red, green and blue components.

Each component ranges from 00 - FF (0 - 255), representing its intensity. For example, pure red color is 00FF0000, pure green color is 0000FF00, and pure blue color is 000000FF. White color is 00FFFFFF, and black color is 00000000.

Most programming language requires you to put special prefix in front of hexadecimal characters. For C++, the prefix is "0x". For example, the syntax for the hexadecimal number 00FFFFFF is 0x00FFFFFF, or simply 0xFFFFFF.

For the alpha transparency component, a zero value means the color is not transparent at all. This is equivalent to traditional RGB colors. A non-zero alpha transparency means the color is partially transparent. The larger the alpha transparency, the more transparent the color will be. If a partially transparent color is used to draw something, the underlying background can still be seen.

For example, 80FF0000 is a partially transparent red color, while 00FF0000 is a non-transparent red color.

Note that ChartDirector's ARGB color is web and HTML compatible. For example, red is FF0000, the same as in HTML. There are many resources on the web that provide tables in which you can click a color and it will show its HTML color code. These color codes can be used in ChartDirector.

If alpha transparency is FF (255), the color is totally transparent. That means the color is invisible. It does not matter what the RGB components are. So in ChartDirector, only one totally transparent color is used - FF000000. All other colors of the form FFnnnnnn are reserved to represent palette colors and dynamic colors, and should not be interpreted as the normal ARGB colors.

The totally transparent color FF000000 is often used in ChartDirector to disable drawing something. For example, if you want to disable drawing the border of a rectangle, you can set the border color to totally transparent.

For convenience, ChartDirector defines a constant called Transparent, which is equivalent to FF000000. Pattern Color

A pattern color is a dynamic color that changes according to a 2D periodic pattern. When it is used to fill an area, the area will look like being tiled with a wallpaper pattern.

Pattern colors are created using `BaseChart.patternColor`, `BaseChart.patternColor2`, `DrawArea.patternColor` and `DrawArea.patternColor2`. The `patternColor` method creates pattern colors using an array of colors as a bitmap. The `patternColor2` method creates pattern colors by loading the patterns from image files.

These methods return a 32-bit integer acting as a handle to the pattern color. The handle can be used in any `ChartDirector` API that expects a color as its input.

Metal Color
A metal color is a color of which the brightness varies smoothly across the chart surface as to make the surface look shiny and metallic. `ChartDirector` supports using any color as the base color of the metal color. In particular, using yellow and grey as the base colors will result in metal colors that look gold and silver.

Metal colors are most often used as background colors of charts. They are created using `CDBaseChartMBS.metalColor`, `CDBaseChartMBS.goldColor` and `CDBaseChartMBS.silverColor`. The first method allows you to specify an arbitrary base color. The second and third methods use yellow and grey as the base colors, resulting in gold and silver metal colors.

These methods return a 32-bit integer acting as a handle to the gradient color. The handle can be used in any `ChartDirector` API that expects a color as its input.

Gradient Color
A gradient color is a color that changes progressively across a direction.

Gradient colors are created using `BaseChart.gradientColor`, `BaseChart.gradientColor2`, `DrawArea.gradientColor` and `DrawArea.gradientColor2`. The `gradientColor` method creates a 2-point gradient color that changes from color A to color B. The `gradientColor2` method creates a multi-point gradient colors that changes from color A to B to C

These methods return a 32-bit integer acting as a handle to the gradient color. The handle can be used in any `ChartDirector` API that expects a color as its input.

One common use of multi-point gradient colors is to define colors that have metallic look and feel. Please refer to `DrawArea.gradientColor2` for details.

Dash Line Colors
A dash line color is a color that switches on and off periodically. When used to draw a line, the line will appear as a dash line.

Dash line colors are created using `BaseChart.dashLineColor` and `DrawArea.dashLineColor`. They accept a line color and a dash pattern code as arguments, and return a 32-bit integer acting as a handle to the dash line color. The handle can be used in any `ChartDirector` API that expects a color as its input.

Zone Colors
A zone color is for XY charts only. It is a color that automatically changes upon reaching a data threshold value along the x-axis or y-axis. Zone colors are created using `Layer.xZoneColor`, `Layer.yZoneColor`, `XYChart.xZoneColor` or `XYChart.yZoneColor`.

Palette Colors
Palette colors are colors of the format `FFFFnnnn`, where the least significant 16 bits (`nnnn`) are the index to the palette. A palette is simply an array of colors. For a palette color, the actual color is obtained by

looking up the palette using the index. For example, the color FFFF0001 is the second color in the palette (first color is index 0).

The colors in the palette can be ARGB colors or "dynamic" colors (pattern, gradient and dash line colors).

The first eight palette colors have special significance. The first three palette colors are the background color, default line color, and default text color of the chart. The 4th to 7th palette colors are reserved for future use. The 8th color is a special dynamic color that is equal to the data color of the "current data set".

The 9th color (index = 8) onwards are used for automatic data colors. For example, in a pie chart, if the sector colors are not specified, ChartDirector will automatically use the 9th color for the first sector, the 10th color for the second sector, and so on. Similarly, for a multi-line chart, if the line colors are not specified, ChartDirector will use the 9th color for the first line, the 10th color for the second line, and so on.

The ChartDirector API defines several constants to facilitate using palette colors.

ConstantValueDescription

Palette	FFFF0000	The starting point of the palette. The first palette color is (Palette + 0). The nth palette color is (Palette + n - 1).
BackgroundColor	FFFF0000	The background color.
LineColor	FFFF0001	The default line color.
TextColor	FFFF0002	The default text color.
[Reserved]	FFFF0003 - FFFF0006	These palette positions are reserved. Future versions of ChartDirector may use these palette positions for colors that have special significance.
SameAsMainColor	FFFF0007	A dynamic color that is equal to the data color of the current data set. This color is useful for objects that are associated with data sets. For example, in a pie chart, if the sector label background color is SameAsMainColor, its color will be the same as the corresponding sector color.
DataColor	FFFF0008	The starting point for the automatic data color allocation.

When a chart is created, it has a default palette. You may modify the palette using BaseChart.setColor, BaseChart.setColors, or BaseChart.setColors2.

The advantages of using palette colors are that you can change the color schemes of the chart in one place. ChartDirector comes with several built-in palettes represented by the following predefined constants.

ConstantDescription

5.0.25 ChartDirector: Font Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

defaultPalette	An array of colors representing the default palette. This palette is designed for drawing charts on white backgrounds (or lightly colored backgrounds).
whiteOnBlackPalette	An array of colors useful for drawing charts on black backgrounds (or darkly colored backgrounds).
transparentPalette	An array of colors useful drawing charts on white backgrounds (or lightly colored backgrounds). The data colors in this palette are all semi-transparent.

Answer: ChartDirector: Font Specification

Notes: Font Name

In ChartDirector, the font name is simply the file name that contains the font. For example, under the Windows platform, the "Arial" font is "arial.ttf", while the "Arial Bold" font is "arialbd.ttf".

NOTE: Mac OS X Specific Information

In Mac OS X, in addition to ".ttf", ChartDirector also supports Mac OS X font file formats, such as Font Suitcase files and Datafork files (.dfont). These files often contain multiple fonts. For example, the "GillSans.dfont" file contains 6 fonts.

So in addition to the file name, an index is needed to determine the font. The index is specified by appending a " | " character to the font name, followed by the index number. For example, the third font in "GillSans.dfont" is denoted as "GillSans.dfont | 2". (Note: The first font starts at 0.) If no index number is provided, the first font is assumed.

ChartDirector also supports using Mac OS X Font Manager names. For example, one may use "Gill Sans Light Italic" instead of using "GillSans.dfont | 1" as the font name. However, the Mac OS X Font Manager is active only if someone has logged into the Mac GUI console, so this method is only recommended for developing applications that run on the GUI console.

The sample programs that come with ChartDirector are designed to run on all operating systems, so they use generic font file names (eg. "arial.ttf") instead of Mac OS X specific names. To allow them to run on Mac OS X, ChartDirector on Mac OS X has a built-in table to map common font file names to Mac OS X font names:

"arial.ttf", "arialbd.ttf", "ariali.ttf" and "arialbi.ttf" are mapped to "Arial | 0" (Arial), "Arial | 1" (Arial Bold), "Arial | 2" (Arial Italic) and "Arial | 3" (Arial Bold Italic)

"times.ttf", "timesbd.ttf", "timesi.ttf" and "timesbi.ttf" are mapped to "Times New Roman | 0" (Times New Roman), "Times New Roman | 1" (Times New Roman Bold), "Times New Roman | 2" (Times New Roman Italic) and "Times New Roman | 3" (Times New Roman Bold Italic)

"cour.ttf", "courbd.ttf", "couri.ttf" and "courbi.ttf" are mapped to "Courier New | 0" (Courier New), "Courier New | 1" (Courier New Bold), "Courier New | 2" (Courier New Italic) and "Courier New | 3" (Courier New Bold Italic)

Font Location

ChartDirector on Windows does not come with any font files. It relies on the operating system's font files in the " [windows] \Fonts" directory. To see what fonts are installed in your operating system and their file names, use the File Explorer to view that directory.

ChartDirector on Windows will also search for the font files in the "fonts" subdirectory (if it exists) under the directory where the ChartDirector DLL "chartdir.dll" is installed. This is useful for private fonts. Also, for some especially secure web servers, the web anonymous user may not have access to the " [windows] \Fonts" directory. In this case, you may copy the font files to the above subdirectory.

ChartDirector on Mac OS X relies on operating system font files in "/Library/Fonts" and "/System/Library/Fonts".

ChartDirector on Linux, FreeBSD and Solaris assume the fonts files are in the "fonts" subdirectory under the directory where the ChartDirector shared object "libchartdir.so" is installed. ChartDirector on Linux, FreeBSD and Solaris come with a number of font files in the "fonts" subdirectory.

To keep the download size small, ChartDirector on Linux, FreeBSD and Solaris only come with some commonly used fonts. You may download additional fonts from the Internet. In particular, the Microsoft fonts at

http://sourceforge.net/project/showfiles.php?group_id=34153&release_id=105355

is highly recommended. Please refer to

<http://www.microsoft.com/typography/faq/faq8.htm>

on how you could use the fonts legally in your system.

ChartDirector supports True Type fonts (.ttf), Type 1 fonts (.pfa and .pfb) and Windows bitmap fonts (.fon). On Mac OS X, ChartDirector also supports Font Suitcase and Datafork (.dfont) files. On Linux, FreeBSD and Solaris, ChartDirector also supports Portable Compiled Fonts (.pcf fonts).

If you want ChartDirector to search other directories for the font files, you may list the directories in an environment variable called "FONTSPATH".

If you specify an absolute path name for the font file, ChartDirector will use the absolute path name and will not search other directories.

Artificial Boldening and Italicizing

Whereas most popular font comes with different styles for "normal", "bold", "italic" and "bold italic", some fonts only come with one style (the normal style). For example, the Monotype Corsiva font that comes with MS Office only has the normal style (mtcorsva.ttf). For these cases, you may append the "Bold" and/or "Italic" words after the font file name (separated with a space) to ask ChartDirector to artificially bolden and/or italicize the font. For example, you may specify the font name as "mtcorsva.ttf Bold".

Font List

Instead of specifying a single font file as the font name, you may specify a list of font files as the font name, separated by semi-colons. This is useful when using international characters that are only available in some fonts.

For example, if you would like to use the Arial font ("arial.ttf") for western characters, and the MingLiu font "mingliu.ttc" for Chinese characters (since the Arial font does not have Chinese characters), you may specify the font name as "arial.ttf;mingliu.ttc". In this case, ChartDirector will try the Arial font first. If it cannot find a certain character there, it will try the MingLiu font.

ChartDirector supports several special keywords for specifying the font name indirectly. When these keywords are used as font names, ChartDirector will look up the actual font names from a font table. The keywords are as follows:

KeywordsDescription

"normal"	This default normal font, which is the first font in the font table. This is initially mapped to "arial.ttf" (Arial).
"bold"	The default bold font, which is the second font in the font table. This is initially mapped to "arialbd.ttf" (Arial Bold).
"italic"	The default italic font, which is the third font in the font table. This is initially mapped to "ariali.ttf" (Arial Italic).
"boldItalic"	The default bold-italic font, which is the fourth font in the font table. This is initially mapped to "arialbi.ttf" (Arial Bold Italic).
"fontN"	The (N + 1)th font in the font table (the first font is "font0").

The font table can be modified using `BaseChart.setFontTable` or `DrawArea.setFontTable`.

The advantage of using indirect font names is that you can change the fonts in your charts in one place.

Most font files contain one font. However, it is possible a font file contains multiple fonts (that is, a font collection). For example, in True Type fonts, font files with extension ".ttc" may represent a font collection.

If a font file contains multiple font, the font index can be used to specify which font to use. By default, the font index is 0, which means the first font in the font file will be used.

The font size decides how big a font will appear in the image. The font size is expressed in a font unit called points. This is the same unit used in common word processors.

Instead of specifying font size, some ChartDirector API (eg. `TextBox.setFontSize`) allow you to specify font height and font width separately. You may use different point sizes for font height and font width to create special effects.

This is the color to draw the font. (See Color Specification on how colors are represented in ChartDirector.)

This is the angle in degrees by which the font should be rotated anti-clockwise.

By default, text are laid out horizontally, with characters being drawn from left to right.

ChartDirector also supports vertical layout, with characters being drawn from top to bottom. For example, you may use `BaseChart.addText` to add text that are laid out vertically. Vertical layout is common for

oriental languages such as Chinese, Japanese and Korean.

5.0.26 ChartDirector: Mark Up Language

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Mark Up Language

Notes: ChartDirector Mark Up Language (CDML) is a language for including formatting information in text strings by marking up the text with tags.

CDML allows a single text string to be rendered using multiple fonts, with different colors, and even embed images in the text. **Font Styles**

You can change the style of the text by using CDML tags. For example, the line:

```
<*font=timesi.ttf,size=16,color=FF0000>Hello <*font=arial.ttf,size=12,color=8000*>world!
```

will result in the following text rendered:

In general, all tags in CDML are enclosed by <*> and *>. Attributes within the tags determine the styles of the text following the tags within the same block.

If you want to include <*> in text without being interpreted as CDML tags, use «* as the escape sequence.

The following table describes the supported font style attributes in CDML. See [Font Specification](#) for details on various font attributes.

Attribute	Description
super	Set the following text to be in superscript style. This attribute does not need to have a value. (You may use "super" as the attribute instead of "super=1".)

Note that unlike HTML tags, no double or single quotes are used in the tags. It is because CDML tags are often embedded as string literals in source code. The double or single quotes, if used, will conflict with the string literal quotes in the source code. Therefore in CDML, no quotes are necessary and they must not be used.

Also, unlike HTML tags, CDML uses the comma character as the delimiter between attributes. It is because certain attributes may contain embed spaces (such as the font file name). So space is not used as the delimiter and the comma character is used instead.

Note the font attribute above starts a new style section, while other attributes just modify the current style

font	Starts a new style section, and sets the font name. You may use this attribute without a value (that is, use "font" instead of "font=arial.ttf") to create a new style section without modifying the font name.
size	The font size.
width	The font width. This attribute is used to set the font width and height to different values. If the width and height are the same, use the size attribute.
height	The font height. This attribute is used to set the font width and height to different values. If the width and height are the same, use the size attribute.
color	The text color in hex format.
bgColor	The background color of the text in hex format.
underline	The line width of the line used to underline the following characters. Set to 0 to disable underline.
sub	Set the following text to be in subscript style. This attribute does not need to have a value. (You may use "sub" as the attribute instead of "sub=1".)
super	Set the following text to be in superscript style.
xoffset	Draw the following the text by shifting the text horizontally from the original position by the specified offset in pixels.
yoffset	Draw the following the text by shifting the text vertically from the original position by the specified offset in pixels.
advance	Move the cursor forward (to the right) by the number of pixels as specified by the value this attribute.
advanceTo	Move the cursor forward (to the right) to the position as specified by the value this attribute. The position is specified as the number of pixels to the right of the left border of the block. If the cursor has already passed through the specified position, the cursor is not moved.

section. You may use `</font*>` to terminate a style section, which will restore the font styles to the state before the style section.

Blocks and Lines

In CDML, a text string may contain multiple blocks. A block may contain multiple lines of text by separating them with new line characters ("`\n`") or with `<br*>`. The latter is useful for programming languages that cannot represent new line characters easily.

For example, the line:

```
<*size=15*><*block*><*color=FF*>BLOCK<*br*>ONE<*/*>and <*block*><*color=FF00*>BLOCK<*br*>TWO
```

will result in the following text rendered:

The above example contains a line of text. The line contains two blocks with the characters " and " in between. Each block in turn contains two lines. The blocks are defined using `<*block*>` as the start tag and

`<*/*>` as the end tag.

When a block ends, font styles will be restored to the state before entering the block. Embedding Images
CDML supports embedding images in text using the following syntax:

```
<*img=my_image_file.png*>
where my_image_file.png is the path name of the image file.
```

For example, the line:

```
<*size=20*>A <*img=sun.png*>day
will result in the following text rendered:
```

ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to `BaseChart.setSearchPath` or `DrawArea.setSearchPath` on the directory that ChartDirector will search for the file.

The `<*img*>` tag may optionally contain width and height attributes to specify its pixel width and height. In this case, ChartDirector will stretch or compress the image if necessary to the required width and height. Blocks Attributes

CDML supports nesting blocks, that is, a block can contain other sub-blocks. Attributes are supported in the `<*block*>` tag to control the alignment and orientation of the sub-blocks. The `<*img=my_image_file.png*>` is treated as a block for layout purposes.

For example, the line:

```
<*block,valign=absmiddle*><*img=molecule.png*><*block*>Hydrazino\nMolecule<*/*><*/*>
will result in the following text rendered:
```

The the above starts `<*block,valign=absmiddle*>` which specifies its content should align with each others in the vertical direction using the absolute middle alignment. The block contains an image, followed by a space characters, and then another block which has two lines of text.

The following table describes the supported attributes inside `<*block*>` tag:

Attribute	Description
-----------	-------------

The value `baseline` means the baseline of sub-blocks should align with the baseline of the block. The `baseline`

width	The width of the block in pixels. By default, the width is automatically determined to be the width necessary for the contents of the block. If the width attribute is specified, it will be used as the width of the block. If the width is insufficient for the contents, the contents will be wrapped into multiple lines.
height	The height of the block in pixels. By default, the height is automatically determined to be the height necessary for the contents of the block. If the height attribute is specified, it will be used as the height of the block.
maxwidth	The maximum width of the block in pixels. If the content is wider than maximum width, it will be wrapped into multiple lines.
truncate	The maximum number of lines of the block. If the content requires more than the maximum number of lines, it will be truncated. In particular, if truncate is 1, the content will be truncated if it exceeds the maximum width (as specified by maxwidth or width) without wrapping. The last few characters at the truncation point will be replaced with "...".
linespacing	The spacing between lines as a ratio to the default line spacing. For example, a line spacing of 2 means the line spacing is two times the default line spacing. The default line spacing is the line spacing as specified in the font used.
bgColor	The background color of the block in hex format.
valign	The vertical alignment of sub-blocks. This is for blocks that contain sub-blocks. Supported values are baseline, top, bottom, middle and absmiddle.

is the underline position of text. This is normal method of aligning text, and is the default in CDML. For images or blocks that are rotated, the baseline is the same as the bottom.

The value top means the top line of sub-blocks should align with the top line of the block.

The value bottom means the bottom line of sub-blocks should align with the bottom line of the block.

The value middle means the middle line of sub-blocks should align with the the middle line of the block. The middle line is the middle position between the top line and the baseline.

The value absmiddle means the absolute middle line of sub-blocks should align with the absolute middle line of the block. The absolute middle line is the middle position between the top line and the bottom line.

halign The horizontal alignment of lines. This is for blocks that contain multiple lines. Supported values are left, center and right.

The value left means the left border of each line should align with the left border of the block. This is the default.

The value center means the horizontal center of each line should align with the horizontal center of the block.

The value right means the right border of each line should align with the right border of the block.

angle Rotate the content of the block by an angle. The angle is specified in degrees in counter-clockwise direction.

5.0.27 ChartDirector: Parameter Substitution and Formatting

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Parameter Substitution and Formatting

Notes: ChartDirector charts often contain a lot of text strings. For example, sector labels in pie charts, axis labels for x and y axes, data labels for the data points, HTML image maps, etc, are all text strings.

ChartDirector uses parameter substitution to allow you to configure precisely the information contained in the text and their format.

Format Strings

In parameter substitution, format strings are used to specify the entities to be include into labels and how to format numbers and dates.

For example, when drawing a pie chart with side label layout, the default sector label format string is:

```
" { label } ( { percent } %)"
```

When the sector label is actually drawn, ChartDirector will replace " { label } " with the sector name, and " { percent } " with the sector percentage. So the above label format will result is a sector label similar to "ABC (34.56%)".

You may change the sector label format by changing the format string. For example, you may change it to:

```
" { label } : US$ { value | 2 } K ( { percent } %)"
```

The sector label will then become something like "ABC: US\$ 123.00 (34.56%)".

In general, in ChartDirector parameter substitution, parameters enclosed by curly brackets will be substituted with their actual values when creating the texts.

For parameters that are numbers or dates/times, ChartDirector supports a special syntax in parameter substitution to allow formatting for these values. Please refer to the Number Formatting and Date/Time Formatting sections below for details.

Parameter Expressions

ChartDirector supports numeric expressions in format strings. They are denoted by enclosing the expression with curly brackets and using "=" as the first character. For example:

```
"USD { value } (Euro { = { value } *0.9 } )"
```

In the above, " { value } " will be substituted with the actual value of the sector. The expression " { = { value } *0.9 } " will be substituted with the actual value of the sector multiplied by 0.9.

ChartDirector parameter expressions support operators "+", "-", "*", "/", "%" (modulo) and "^" (exponentiation). Operators "*", "/", "%", "^" is computed first, followed by "+" and "-". Operators of the same precedence are computed from left to right). Parenthesis "(" and ")" can be used to change the computation order.

Parameters for Pie Charts

The following table describes the parameters available for pie charts.

Parameter	Description
sector	The sector number. The first sector is 0, while the nth sector is (n-1).
dataSet	Same as { sector } . See above.
label	The text label of the sector.
dataSetName	Same as { label } . See above.
value	The data value of the sector.
percent	The percentage value of the sector.
fieldN	The (N + 1)th extra field. For example, { field0 } means the first extra field. An extra field is an array of custom elements added using BaseChart.addExtraField or BaseChart.addExtraField2.

Parameters for All XY Chart Layers

The followings are parameters that are apply to all XY Chart layers in general. Some layer types may have additional parameters (see below).

Note that certain parameters are inapplicable in some context. For example, when specifying the aggregate label of a stacked bar chart, the { dataSetName } parameter is inapplicable. It is because a stacked bar is composed of multiple data sets. It does not belong to any particular data set and hence does not have a data set name.

{ fieldN } means the extra field is indexed by the data point number. The Pth data point corresponds to the Pth element of the extra field.

Additional Parameters for Line Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for Trend Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for Box-Whisker Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for HLOC and CandleStick Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for Vector Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Parameters for All Polar Layers

The followings are parameters that are apply to all Polar Chart layers in general. Some layer types may have additional parameters (see below).

{ fieldN } means the extra field is indexed by the data point number. The Pth data point corresponds to the Pth element of the extra field.

Additional Parameters for PolarVector Layers

The followings are parameters that are in additional to the parameters for all Polar Chart layers.

Parameters for Axis

The following table describes the parameters available for pie charts.

Number Formatting

For parameters that are numbers, ChartDirector supports a number of formatting options in parameter substitution.

For example, if you want a numeric field { value } to have a precision of two digits to the right of the decimal point, use ',' (comma) as the thousand separator, and use '.' (dot) as the decimal point, and you may use { value | 2, . } . The number 123456.789 will then be displayed as 123,456.79.

For numbers, the formatting options are specified using the following syntax:

```
{ [ param ] | [ a ] [ b ] [ c ] [ d ] }
```

where:

If this field starts with "E" or "e", followed by a number, it means formatting the value using scientific notation with the specified number of decimal places. If the "E" or "e" is not followed by a number, 3 is assumed.

For example, { value | E4 } will format the value 10.3 to 1.0300E+1, and { value | e4 } will format the same value to 1.0300e+1.

If this field starts with "G" or "g", followed by a number, it means formatting the value using the scientific notation only if the value is large and requires more than the specified number of digits, or the value is less than 0.001. If scientific notation is used, the number following "G" or "g" also specifies the number of significant digits to use. If the "G" or "g" is not followed by a number, 4 is assumed.

For example, consider the format string { value | G4 } . The value 10 will be formatted to 10. The value 100000 will be formatted to 1.000E+5. Similarly, for { value | g4 } , the value 10 will be formatted to 10, while the value 100000 will be formatted to 1.000e+5.

If you skip this argument, ChartDirector will display the exact value using at most 6 decimal places.

You may skip [b] [c] [d] . In this case, the default will be used.

Date/Time Formatting

For parameters that are dates/times, the formatting options can be specified using the following syntax:

```
{ [ param ] | [ datetime_format_string ] }
```

where [datetime_format_string] must start with an english character (A-Z or a-z) that is not "G", "g", "E" or "e", and may contain any characters except ' ' . (If it starts with "G", "g", "E" or "e", it will be considered as a number format string.)

Certain characters are substituted according to the following table. Characters that are not substituted will be copied to the output.

For example, a parameter substitution format of { value | mm-dd-yyyy } will display a date as something similar to 09-15-2002. A format of { value | dd/mm/yy hh:nn:ss a } will display a date as something similar to 15/09/02 03:04:05 pm.

If you want to include characters in the format string without substitution, you may enclose the characters in single or double quotes.

For example, the format `{ value | mmm '<*color=dd0000*>'yyyy }` will display a date as something like `Jan <*color=dd0000*>2005` (the `<*color=dd0000*>` is a CDML tag to specify red text color). Note that the `<*color=dd0000*>` tag is copied directly without substitution, even it contains "dd" which normally will be substituted with the day of month.

Escaping URL/HTML/CDML characters

Parameter substitution is often used to create HTML image maps. In HTML, some characters has special meanings and cannot be used reliably. For example, the `'>'` is used to represent the end of an HTML tag.

Furthermore, if the field happens to be used as an URL, characters such as `'?'`, `'&'` and `'+'` also have special meanings.

By default, ChartDirector will escape template fields used in URL and query parameters when generating image maps. It will modify URL special characters to the URL escape format `"%XX"` (eg. `"?"` will become `"%3F"`). After that, it will modify HTML special characters to the HTML escape format `"&#nn;"` (eg. `">"` will become `">"`). Similarly, it will escape other attributes in the image map using HTML escape format (but not URL escape format).

In addition to escaping HTML and URL special characters, ChartDirector will also remove CDML fields in creating image maps. It is because CDML is only interpreted in ChartDirector, should not be useful outside of ChartDirector (such as in browser tool tips).

In some cases, you may not want ChartDirector to escape the special characters. For example, if the parameters have already been escaped before passing to ChartDirector, you may want to disable ChartDirector from escaping them again.

ChartDirector supports the following special fields to control the escape methods - `" { escape_url } "`, `" { noescape_url } "`, `" { escape_html } "`, `" { noescape_html } "`, `" { escape_cdml } "` and `" { noescape_cdml } "`. These fields enable/disable the escape methods used in the template fields that follow them.

5.0.28 ChartDirector: Shape Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Shape Specification

Notes: Several ChartDirector API accept shape specification as arguments. For example, `BarLayer.setBarShape` and `BarLayer.setBarShape2` can be used to specify shapes of bars in bar charts, while `DataSet.setDataSymbol`, `DataSet.setDataSymbol4`, `PolarLayer.setDataSymbol` and `PolarLayer.setDataSymbol4` can be used to specify shapes for data symbols.

Note that in addition to shapes, in many cases ChartDirector also accepts images or custom draw objects for data representation. For example, see `DataSet.setDataSymbol2`, `DataSet.setDataSymbol3`, `PolarLayer.setDataSymbol2` and `PolarLayer.setDataSymbol3`.

Built-In Shapes

Built-in shapes are specified as integers. The integers can be explicit constants, or can be generated by a `ChartDirector` method for parameterized shapes. For example, a circle is represented by an explicit constant `CircleShape (=7)`. On the other hand, the number representing a polygon depends on the number of sides the polygon has, so it is generated by using the `PolygonShape` method, passing in the number of sides as argument.

The following table illustrates the various `ChartDirector` shapes:

Custom Shapes

In `ChartDirector`, custom shapes are specified as an array of integers `x0, y0, x1, y1, x2, y2 ...` representing the coordinates of the vertices of the custom polygonal shape.

The polygon should be defined with a bounding square of 1000 x 1000 units, in which the x-axis is from -500 to 500 going from left to right, and the y-axis is from 0 to 1000 going from bottom to top.

`ChartDirector` will automatically scale the polygon so that 1000 units will become to the pixel size as requested by the various `ChartDirector` API.

As an example, the shape of the standard diamond shape in `ChartDirector` is represented as an array with 8 numbers:

```
0, 0, 500, 500, 0, 1000, -500, 500
```

5.0.29 Copy styled text?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: How to quickly copy styled text from one textarea to another?

Example:

```
#if TargetWin32 then
TextArea1.WinRTFDataMBS = TextArea2.WinRTFDataMBS
#elseif TargetMacOS then
TextArea1.NSTextViewMBS.textStorage.setAttributedString TextArea2.NSTextViewMBS.textStorage
#else
TextArea1.StyledText = TextArea2.StyledText
#endif
```

Notes: The code above uses special plugin functions on Mac and Windows and falls back to framework for Linux.

5.0.30 Do you have code to validate a credit card number?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can check the checksum to tell if a credit card number is not valid.

Example:

```
Dim strNumber As String
Dim nLength as Integer
Dim nValue as Integer
Dim nChecksum as Integer
Dim nIndex as Integer

strNumber = EditField1.Text
nLength = Len(strNumber)
nChecksum = 0

For nIndex = 0 To nLength - 2
nValue = Val(Mid(strNumber, nLength - (nIndex + 1), 1)) * (2 - (nIndex Mod 2))
If nValue <10 Then
nChecksum = nChecksum + nValue
Else
nChecksum = nChecksum + (nValue - 9)
End If
Next

If Val(Mid(strNumber, Len(strNumber), 1)) = (10 - (nChecksum Mod 10)) Mod 10 Then
MsgBox("The credit card number looks valid")
Else
MsgBox("The credit card number is invalid")
End IF
```

Notes: Here's some code that will validate the checksum for a credit card. It works for Visa, MasterCard, American Express and Discover. Not sure about others, but I imagine they use the same basic algorithm. Of course, this doesn't actually mean that the credit card is valid, it's only useful for helping the user catch typos.

The above code doesn't have any error checking and it expects that the credit card number will be entered without spaces, dashes or any other non-numeric characters. Addressing those issues will be an exercise left to the reader. :)

(From Mike Stefanik)

5.0.31 Do you have plugins for X-Rite EyeOne, eXact or i1Pro?

Plugin Version: all.

Answer: Our EyeOne plugin is available on request for licensees of the X-Rite SDKs.

Notes: Please first go to X-Rite and get a SDK license.

Then we can talk about the plugin.

5.0.32 Does SQL Plugin handle stored procedures with multiple result sets?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, the plugin can work with multiple recordsets.

Notes: You need to use SQLCommandMBS class. When you get back results, you use FetchNext to walk over all records in the first result set. Then you simply start again with FetchNext to get the second record set.

Even the RecordSet functions should work, just use them twice to get all records from both record sets.

5.0.33 Does the plugin home home?

Plugin Version: all, Platform: macOS.

Answer: Yes, we like to know who is using the plugin, so the plugin may contact our server.

Example:

none.

Notes: Please note that this does not affect your users as the plugin will only do this in the IDE and the relevant plugin part is never included in your applications.

The plugin if used for some hours, does contact our server to provide statistical data about Xojo version and OS versions. This way we know what versions are used. We can return the version number of the current plugin which may be visible in future versions somehow. And we transmit partial licenses data so we can track use of illegal license keys.

If you do not like to have this, you can block Xojo IDE from contacting our website via your Firewall.

Blocking the transfer will not disable the plugin or change the features.

Or contact us for a plugin version which explicitly does not contain this feature.

5.0.34 folderitem.absolutePath is limited to 255 chars. How can I get longer ones?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Paths on a Mac are not unique, so use them only to display them to the user.

Example:

```
Function AbsolutePath(f as FolderItem) As String
Dim s as string
Dim nf as FolderItem
nf = f
s = ""
while nf<>nil
s = nf.name + "." + s
nf = nf.parent
wend
Return s
End Function
```

5.0.35 Has anyone played round with using CoreImage to do things like add dissolve transitions say when changing from one tab to another within a window?

Platform: macOS.

Answer: This code implements animations for a tabpanel change:

Example:

// in a tabpanel.change event:

```
dim r as CGSTransitionRequestMBS
dim co as new CGSConnectionMBS
dim cw as CGSWindowMBS
dim ct as CGSTransitionMBS
static OldTab as Integer

cw=co.CGSWindow(window1)
If cw = Nil Then
return // 10.3...
End If
r=new CGSTransitionRequestMBS
r.TransitionType=r.CGSFlip
r.HasBackGround=false
r.HasBackColor=false
r.Win=cw
```

```

// watch the value of the clicked tab versus the last tab
if tabpanel1.Value=0 or tabpanel1.Value <OldTab then
r.TransitionOption=r.CGSLeft
ct=co.NewTransition(r)
if ct<>Nil then
Refresh
ct.Invoke(1)
ct.Wait(1)
ct.Release
else
MsgBox "Error creating the transition."
end if
else
r.TransitionOption=r.CGSRight
ct=co.NewTransition(r)
if ct<>Nil then
Refresh
ct.Invoke(1)
ct.Wait(1)
ct.Release
else
MsgBox "Error creating the transition."
end if
end if
// Keep track of the last tab clicked
OldTab = tabpanel1.Value

```

Notes: See CGS* classes for more details.

5.0.36 How about Plugin support for older OS X?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We support in general Mac OS X 10.5 and newer.

Notes: All the 64-bit plugins on Mac require OS X 10.7.

Intel 32-bit plugins on Mac require OS X 10.5 or newer.

Currently the ChartDirector 6, GraphicsMagick and GameKit plugins requires Mac OS X 10.6. Also for SQL Plugin the built in SQLite library requires 10.6.

5.0.37 How can I detect whether an Intel CPU is a 64bit CPU?

Plugin Version: all.

Answer: Look on the CPU family returned by sysctl:

Example:

Function is64bit() As Boolean

```
#if TargetLittleEndian
```

```
dim m as MemoryBlock = NewMemoryBlock(8)
```

```
dim family as Integer
```

```
dim s as string
```

```
m=SystemControlNameToMIBMBS("hw.cpufamily")
```

```
m=SystemControlMBS(m)
```

```
if m<>nil then
```

```
m.LittleEndian=True
```

```
family=m.Long(0)
```

```
const CPUFAMILY_INTEL_6_14 = &h73d67300 /* "Intel Core Solo" and "Intel Core Duo" (32-bit Pentium-M with SSE3) */
```

```
const CPUFAMILY_INTEL_6_15 = &h426f69ef /* "Intel Core 2 Duo" */
```

```
const CPUFAMILY_INTEL_6_23 = &h78ea4fbc /* Penryn */
```

```
const CPUFAMILY_INTEL_6_26 = &h6b5a4cd2 /* Nehalem */
```

```
Select case family
```

```
case CPUFAMILY_INTEL_6_14
```

```
Return false
```

```
case CPUFAMILY_INTEL_6_15
```

```
Return true
```

```
case CPUFAMILY_INTEL_6_23
```

```
Return true
```

```
case CPUFAMILY_INTEL_6_26
```

```
Return true
```

```
// newer CPUs may be missing here
```

```
end Select
```

```
end if
```

```
#endif
```

```
Return false
```

```
Exception
```

```
Return false
```

```
End Function
```

Notes: This code is written for Mac OS X where you only have a limited number of possible CPUs.

5.0.38 How can I disable the close box of a window on Windows?

Plugin Version: all, Platform: Windows.

Answer: The following code will remove the close item from the system menu of the window.

Example:

```
#if TargetWin32 then
Declare Function GetSystemMenu Lib "user32" (hwnd as Integer, bRevert as Integer) as Integer
Declare Function RemoveMenu Lib "user32" (hMenu as Integer, nPosition as Integer, wFlags as Integer) as Integer
Dim hSysMenu as Integer
hSysMenu = GetSystemMenu(me.WinHWND, 0)
RemoveMenu hSysMenu, &HF060, &H0
#endif
```

Notes: The window may not be updated directly.

5.0.39 How can I get all the environment variables from Windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
#if targetWin32
declare function GetEnvironmentStrings Lib "kernel32" () as ptr
dim m as memoryBlock
dim n as Integer

m=GetEnvironmentStrings()

n=0
do
msgBox m.cstring(n)
while m.byte(n)<>0
n=n+1
wend
n=n+1
```

```
loop until m.byte(n)=0
#endif
```

Notes: The MBS Plugin has an EnvironmentMBS class for this.

5.0.40 How can i get similar behavior to Roxio Toast or iTunes where clicking a 'burn' button allows the next inserted blank CD-R to bypass the Finder and be accepted by my application?

Plugin Version: all, Platform: macOS.

Answer: You need to get a media reservation.

Example:

```
dim d as DRDeviceMBS // get a device
d.AcquireMediaReservation
```

Notes: Use the plugin function AcquireMediaReservation and later release it using ReleaseMediaReservation.

See plugin examples on how to use it and check Apples DiscRecording framework documentation for more details.

5.0.41 How can I get text from a PDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Crossplatform you can use DynaPDF Pro.

Notes: On Mac OS X you can also use PDFKit for the same job.

While DynaPDF Pro gives you each bit of text with rotation, font information and encoding details, PDFKit gives you only the text string for a PDF page.

5.0.42 How can I get text from a Word Document?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: to get the text string from a doc file, use the NSAttributedStringMBS class.

Notes: The NSAttributedStringMBS class is Mac OS X only and we have currently no solution for Windows or Linux.

Use the `NSAttributedStringMBS.initWithDocFormat(data as string)` as boolean method.

5.0.43 How can I get the item string for a given file creator?

Plugin Version: all.

Answer: Try this function:

Example:

```
Sub pullNativeDocs(aCREA As string)
Dim result as Integer
Dim m, k as memoryBlock
Dim f as folderItem
Dim newType as string
Dim anIcon As picture
Dim ofs as Integer
```

```
Declare Function GetFileTypesThatAppCanNativelyOpen Lib "Carbon" (appVRefNumHint as Short, appSignature as OSType, nativeTypes as Ptr) as Short Inline68K("701CABFC")
Declare Function GetDocumentKindString Lib "Carbon" (docVRefNum as Short, docType as OSType, docCreator as OSType, kindString as ptr) as Short Inline68K("7016ABFC")
```

```
listBox1.deleteAllRows
```

```
m = newMemoryBlock(1024)
result = GetFileTypesThatAppCanNativelyOpen(Volume(0).MacVRefNum, aCREA, m)
if result <> 0 then
listBox1.addRow "<Not found.>"
return
end if
```

```
do
if m.byte(ofs*4) = 0 then
exit
else
newType = m.OSTypeMBS(ofs*4)
listBox1.addRow newType
k = newMemoryBlock(64)
result = GetDocumentKindString(Volume(0).MacVRefNum, newType, aCREA, k)
if result = 0 then
listBox1.cell(ofs,1) = k.pString(0)
ofs = ofs + 1
else
listBox1.cell(ofs,1) = "(unknown)"
end if
end if
```

loop

End Sub

Notes: Change "Translation" to "CarbonLib" for Mac OS X.

5.0.44 How can I launch an app using it's creator code?

Plugin Version: all, Platform: macOS.

Answer: Send an AppleEvent "odoc" with the creator code to the Finder ("MACS"):

Example:

```
Function LaunchByCreator(C As String) As Boolean
Dim A As AppleEvent
A = NewAppleEvent("aevt","odoc","MACS")
A.ObjectSpecifierParam("—") = GetUniqueIDObjectDescriptor("appf",nil,C)
return A.Send
End Function
```

5.0.45 How can I learn what shared libraries are required by a plugin on Linux?

Plugin Version: all, Platform: macOS.

Answer: Please use the ldd command in the terminal.

Notes: You build an app on any platform, but for Linux.

For the resulting .so files in the libs folder, you can run the ldd command with the library path as parameter. It shows you references lib files and you can make sure you have those installed.

This is a sample run of our graphicsmagick plugin:

```
cs@Ubuntu32:
textasciitilde /MeinProgramm/MeinProgramm Libs$ ldd libMBSGraphicsMagickPlugin17744.so
linux-gate.so.1 =>(0xb76ee000)
libdl.so.2 =>/lib/i386-linux-gnu/libdl.so.2 (0xb6f0e000)
libgtk-x11-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgtk-x11-2.0.so.0 (0xb6aa6000)
libpthread.so.0 =>/lib/i386-linux-gnu/libpthread.so.0 (0xb6a8a000)
libstdc++.so.6 =>/usr/lib/i386-linux-gnu/libstdc++.so.6 (0xb69a5000)
libm.so.6 =>/lib/i386-linux-gnu/libm.so.6 (0xb6979000)
libgcc_s.so.1 =>/lib/i386-linux-gnu/libgcc_s.so.1 (0xb695b000)
libc.so.6 =>/lib/i386-linux-gnu/libc.so.6 (0xb67b1000)
```

```

/lib/ld-linux.so.2 (0xb76ef000)
libgdk-x11-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgdk-x11-2.0.so.0 (0xb6701000)
libpangocairo-1.0.so.0 =>/usr/lib/i386-linux-gnu/libpangocairo-1.0.so.0 (0xb66f4000)
libX11.so.6 =>/usr/lib/i386-linux-gnu/libX11.so.6 (0xb65c0000)
libXfixes.so.3 =>/usr/lib/i386-linux-gnu/libXfixes.so.3 (0xb65ba000)
libatk-1.0.so.0 =>/usr/lib/i386-linux-gnu/libatk-1.0.so.0 (0xb659a000)
libcairo.so.2 =>/usr/lib/i386-linux-gnu/libcairo.so.2 (0xb64ce000)
libgdk_pixbuf-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgdk_pixbuf-2.0.so.0 (0xb64ad000)
libgio-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgio-2.0.so.0 (0xb6356000)
libpangoft2-1.0.so.0 =>/usr/lib/i386-linux-gnu/libpangoft2-1.0.so.0 (0xb632a000)
libpango-1.0.so.0 =>/usr/lib/i386-linux-gnu/libpango-1.0.so.0 (0xb62e0000)
libfontconfig.so.1 =>/usr/lib/i386-linux-gnu/libfontconfig.so.1 (0xb62ab000)
libgobject-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgobject-2.0.so.0 (0xb625c000)
libglib-2.0.so.0 =>/lib/i386-linux-gnu/libglib-2.0.so.0 (0xb6163000)
libXext.so.6 =>/usr/lib/i386-linux-gnu/libXext.so.6 (0xb6151000)
libXrender.so.1 =>/usr/lib/i386-linux-gnu/libXrender.so.1 (0xb6147000)
libXinerama.so.1 =>/usr/lib/i386-linux-gnu/libXinerama.so.1 (0xb6142000)
libXi.so.6 =>/usr/lib/i386-linux-gnu/libXi.so.6 (0xb6132000)
libXrandr.so.2 =>/usr/lib/i386-linux-gnu/libXrandr.so.2 (0xb6129000)
libXcursor.so.1 =>/usr/lib/i386-linux-gnu/libXcursor.so.1 (0xb611e000)
libXcomposite.so.1 =>/usr/lib/i386-linux-gnu/libXcomposite.so.1 (0xb611a000)
libXdamage.so.1 =>/usr/lib/i386-linux-gnu/libXdamage.so.1 (0xb6115000)
libfreetype.so.6 =>/usr/lib/i386-linux-gnu/libfreetype.so.6 (0xb607b000)
libxcb.so.1 =>/usr/lib/i386-linux-gnu/libxcb.so.1 (0xb605a000)
libpixman-1.so.0 =>/usr/lib/i386-linux-gnu/libpixman-1.so.0 (0xb5fc2000)
libpng12.so.0 =>/lib/i386-linux-gnu/libpng12.so.0 (0xb5f98000)
libxcb-shm.so.0 =>/usr/lib/i386-linux-gnu/libxcb-shm.so.0 (0xb5f93000)
libxcb-render.so.0 =>/usr/lib/i386-linux-gnu/libxcb-render.so.0 (0xb5f89000)
libz.so.1 =>/lib/i386-linux-gnu/libz.so.1 (0xb5f73000)
libgmodule-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgmodule-2.0.so.0 (0xb5f6e000)
libselinux.so.1 =>/lib/i386-linux-gnu/libselinux.so.1 (0xb5f4f000)
libresolv.so.2 =>/lib/i386-linux-gnu/libresolv.so.2 (0xb5f36000)
libexpat.so.1 =>/lib/i386-linux-gnu/libexpat.so.1 (0xb5f0c000)
libffi.so.6 =>/usr/lib/i386-linux-gnu/libffi.so.6 (0xb5f05000)
libpcre.so.3 =>/lib/i386-linux-gnu/libpcre.so.3 (0xb5ec9000)
librt.so.1 =>/lib/i386-linux-gnu/librt.so.1 (0xb5ec0000)
libXau.so.6 =>/usr/lib/i386-linux-gnu/libXau.so.6 (0xb5ebb000)
libXdmcp.so.6 =>/usr/lib/i386-linux-gnu/libXdmcp.so.6 (0xb5eb4000)
cs@Ubuntu32:
textasciitilde /MeinProgramm/MeinProgramm Libs$

```

As you see all library have been found and their load address is printed behind the name. If a library is missing, you usually see the address missing there or being zero.


```

while theRegexMatch <>nil
theStart = theRegexMatch.subExpressionStartB(0) + len(theRegexMatch.subExpressionString(0))

result = result + theRegexMatch.subExpressionString(1)
infoCharset = theRegexMatch.subExpressionString(2)
encodedPart = theRegexMatch.subExpressionString(4)
if theRegexMatch.subExpressionString(3) = "B" then
encodedPart = DecodeBase64(encodedPart)
elseif theRegexMatch.subExpressionString(3) = "Q" then
encodedPart = DecodeQuotedPrintable(encodedPart)
end if
if right(result, 1) = " " then
result = mid(result, 1, len(result)-1)
end if
encodedPart = encodedPart.DefineEncoding(GetInternetTextEncoding(infoCharset))
result = result + encodedPart

theRegex.SearchStartPosition = theStart
theRegexMatch = theRegex.search()
wend

result = result + mid(src, theStart+1)

else
result = src
end if
// theRegexMatch = theRegex.search

msgbox result

```

Notes: May not look nice depending on the controls used.
This is no longer needed when using MimeEmailMBS class which decodes for you.

5.0.48 How do I enable/disable a single tab in a tabpanel?

Plugin Version: all, Platform: macOS.

Answer: Use the TabpanelEnabledMBS method.

Example:

```
TabpanelEnabledMBS(tabpanel1, 1, false)
```

Notes: Use Carbon for MachO and CarbonLib for Mac Carbon and AppearanceLib for Mac OS Classic as

library.

For Cocoa, please use enabled property of NSTabViewItemMBS class.

5.0.49 How do I find the root volume for a file?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this function:

Example:

```
Function GetRootVolume(f as FolderItem) as FolderItem
dim root, dum as folderItem
if f <> nil then
root = f // f might be the volume
do
dum = root.parent
if dum <> nil then
root = dum
end if
loop until dum = nil
return root
end if
End Function
```

5.0.50 How do I get the current languages list?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
dim p as new CFPREFERENCESMBS
dim a as CFArrayMBS
dim s as CFStringMBS
dim o as CFOBJECTMBS
dim sa(-1) as string

o=p.CopyAppValue("AppleLanguages", ".GlobalPreferences")

if o<>Nil then
a=CFArrayMBS(o)

dim i,c as Integer
```

```
c=a.Count-1
for i=0 to c
o=a.Item(i)

if o isa CFStringMBS then
s=CFStringMBS(o)
sa.Append s.str
end if
next
end if

MsgBox Join(sa,EndOfLine)
```

Notes: On Mac OS X you can get the list of current languages like this list:

```
de
en
ja
fr
es
it
pt
pt-PT
nl
sv
nb
da
fi
ru
pl
zh-Hans
zh-Hant
ko
```

Which has German (de) on the top for a German user.
This code has been tested on Mac OS X 10.5 only.

5.0.51 How do I get the Mac OS Version?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

dim i as Integer
if system.gestalt("sysv", i) then
//do this in an 'If' in case you don't get any value back at all and system.gestalt returns boolean
if i = &h750 then //If OS is 7.5
//do stuff
elseif i = &h761 then //If OS is 7.6.1
//do stuff
end if
end if

```

Notes: The MBS Plugin has a function SystemInformationMBS.OSVersionString for this.

5.0.52 How do I get the printer name?

Plugin Version: all.

Answer: For Mac OS Classic see the code below and for Mac OS X use the Carbon Print Manager Classes from the MBS Plugin.

Example:

```

dim s as String
dim i as Integer

s=app.ResourceFork.GetResource("STR ",-8192)
if s<>"" then
i=ascb(leftb(s,1))
s=mid(s,2,i)

MsgBox s
end if

```

Notes: A note from Craig Hoyt:

After looking at your example I had a little deja-vu experience. Several years ago I played around with this same code if FutureBasic. I discovered that it did not and still doesn't provide the 'Printer Name', it does return the print driver name. If it returns 'LaserWriter 8' as the print driver you can look into this file and get the 'PAPA' resource #-8192 to get the actual Printer Name. Unfortunately this does not hold true for other printers. My Epson and HP Printers (the Epson has an Ethernet Card and the HP is USB) do not provide this info in their drivers. As far as I can tell it only returns the name by polling the printer itself.

5.0.53 How do I make a metal window if RB does not allow me this?

Plugin Version: all, Platform: macOS.

Answer: The following declare turns any window on Mac OS X 10.2 or newer into a metal one.

Example:

```
declare sub ChangeWindowAttributes lib "Carbon" (win as windowptr, a as Integer, b as Integer)
```

```
ChangeWindowAttributes window1,256,0
```

Notes: May not look nice depending on the controls used.

5.0.54 How do I make a smooth color transition?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

I'd like to show in a report some bars, which start with color A and end with color B.

The color change should be very smooth.

My problem: If I would start from 255,0,0 and end by 0,0,0, I would have 255 different colors. If the bars are longer than 255 pixels, would this look nice?

Example:

```
// Window.Paint:
Sub Paint(g As Graphics)
dim w,w1,x,p as Integer
dim c1,c2,c as color
dim p1,p2 as Double

c1=rgb(255,0,0) // start color
c2=rgb(0,255,0) // end color

w=g.Width
w1=w-1

for x=0 to w1
p1=x/w1
p2=1.0-p1
```

```

c=rgb(c1.red*p1+c2.red*p2, c1.green*p1+c2.green*p2, c1.blue*p1+c2.blue*p2)

g.ForeColor=c
g.DrawLine x,0,x,g.Height

next
End Sub

```

Notes:

Try the code above in a window paint event handler.

5.0.55 How do I read the applications in the dock app?

Plugin Version: all, Platform: macOS.

Answer: Use CFPreferencesMBS class like in this example:

Example:

```

// Reads file names from persistent dock applications and puts them into the list

dim pref as new CFPreferencesMBS

dim persistentapps as CFStringMBS = NewCFStringMBS("persistent-apps")
dim ApplicationID as CFStringMBS = NewCFStringMBS("com.apple.dock")
dim tiledata as CFStringMBS = NewCFStringMBS("tile-data")
dim filelabel as CFStringMBS = NewCFStringMBS("file-label")

// get the array of persistent applications from dock preferences
dim o as CObjectMBS = pref.CopyValue(persistentapps, ApplicationID, pref.kCFPreferencesCurrentUser,
pref.kCFPreferencesAnyHost)

if o isa CFArrayMBS then
dim a as CFArrayMBS = CFArrayMBS(o)

// walk over all items in array
dim c as Integer = a.Count-1
for i as Integer = 0 to c

// get dictionary describing item
o = a.Item(i)

if o isa CFDictionaryMBS then
dim d as CFDictionaryMBS = CFDictionaryMBS(o)

```

```

// and pick tile data dictionary
o = d.Value(tiledata)
if o isa CFDictionaryMBS then
d = CFDictionaryMBS(o)

// and pick there the file label
o = d.Value(filelabel)
if o isa CFStringMBS then
// and display it
dim name as string = CFStringMBS(o).str
List.AddRow name
end if
end if
end if

next

else
MsgBox "Failed to read dock preferences."
end if

```

Notes: You can use the `CFPreferencesMBS.SetValue` to change a value and `CFPreferencesMBS.Synchronize` to write the values to disc. You may need to restart the `Dock.app` if you modified things.

5.0.56 How do I truncate a file?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: In a `binarystream` you can set the `length` property to truncate.

5.0.57 How do update a Finder's windows after changing some files?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

dim f as folderitem // some file
dim ae as appleevent
ae=newappleevent("fndr","fupd","MACS")
ae.folderitemparam("—")=f
if not ae.send then
//something went wrong

```

end if

Notes: The `folderitem.finderupdate` from the MBS Plugin does something like this.

5.0.58 How to access a USB device directly?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: First, it depends on the device.

Notes: Some devices can be talked directly from user mode code, but some require a kernel driver.

For some devices you can use plugins to access them like:

- Audio and Video sources using the `QTGrabberClassMBS`
- Mass storage devices using the `folderitem` class.
- Serial devices using the `System.SerialPort` function.
- HID USB devices can be used with `MacHIDMBS`, `WinHIDMBS` or `LinuxHIDInterface` class.
- Any USB device may be used with `MacUSBMBS` or `WinUSBMBS` classes.

In general it is always the best to take the most high level access to have others do the work for the details.

5.0.59 How to add icon to file on Mac?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use `FolderItem.AddCustomIcon` or `NSWorkspaceMBS.setIcon` functions.

Notes: Please close any open stream for the file you want to add an icon.

5.0.60 How to ask the Mac for the Name of the Machine?

Plugin Version: all, Platform: macOS.

Answer: Using Apple Events you can use this code:

Example:

Function `Computername()` *As string*

```

dim theEvent as AppleEvent
dim err as boolean

theEvent = newAppleEvent("mchn","getd","MACS")

err = theEvent.send

return theevent.ReplyString

End Function

```

Notes: Code above is for Mac OS 9!

Also the MBS Plugin has a function for this which may be faster and work also on Macs without Filesharing (which handles this event).

5.0.61 How to automatically enable retina in my apps?

Plugin Version: all, Platform: macOS.

Answer: You can run a build script on each build with this code:

Example:

```

Dim App As String = CurrentBuildLocation + "/" + CurrentBuildAppName + ".app"
Call DoShellCommand("/usr/bin/defaults write " + App + "/Contents/Info ""NSHighResolutionCapable""
YES")

```

Notes: This will set the NSHighResolutionCapable flag to YES.

5.0.62 How to avoid leaks with Cocoa functions?

Plugin Version: all, Platform: macOS.

Answer: You can try this code on Mac OS X:

Example:

```

// in a Timer Action event:
Sub Action()
static LastPool as NSAutoreleasePoolMBS = nil
static CurrentPool as NSAutoreleasePoolMBS = nil

```

```

LastPool = CurrentPool
CurrentPool = new NSAutoreleasePoolMBS

```

End Sub

Notes: With Xojo 2009r4 the code above should not be needed as Xojo runtime does automatically handle the NSAutoreleasePools for you. For older Xojo versions you need to use code with a timer with the action event above to avoid memory leaks.

Please do not use Xojo 2009r4 and newer with plugins before version 9.5. You can get crashes there which typically show a line with a objc_msgSend call.

5.0.63 How to avoid trouble connecting to oracle database with SQL Plugin?

Plugin Version: all, Platform: macOS.

Answer: For oracle the most important thing is to point the plugin to the libraries from oracle.

Notes: In environment variables, the paths like ORACLE_HOME must be defined.

On Mac OS X you also need to define DYLD_LIBRARY_PATH to point to the dylib files from oracle.

For that you need to modify /etc/launchd.conf for Mac OS X 10.8 and newer.

In older versions those variables in .MacOSX/environment.plist file in user's home.

Another way for the case you bundle things inside your app is to use the LSEnvironment key in info.plist. In info.plist it looks like this:

```
<key>LSEnvironment</key>
<dict>
<key>test</key>
<string>Hello World</string>
</dict>
```

5.0.64 How to avoid ___NSAutoreleaseNoPool console messages in threads?

Plugin Version: all, Platform: macOS.

Answer: You need to use your own NSAutoreleasePool on a thread like this:

Example:

```
sub MyThread.run
dim pool as new NSAutoreleasePoolMBS
// do work here

pool=nil
```

```
end sub
```

Notes: For more details read here:

http://developer.apple.com/mac/library/documentation/Cocoa/Reference/Foundation/Classes/NSAutoreleasePool_Class/Reference/Reference.html

5.0.65 How to bring app to front?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac you can use this code:

Example:

```
// First way:  
app.FrontMostMBS = true
```

```
// second way:  
dim p as new ProcessMBS  
p.GetCurrentProcess  
p.FrontProcess = true
```

```
// third way:  
NSApplicationMBS.sharedApplication.activateIgnoringOtherApps(true)
```

```
// for Windows:  
RemoteControlMBS.WinBringWindowToTop
```

Notes: This will bring a Mac app to the front layer.

5.0.66 How to bring my application to front?

Plugin Version: all, Platform: macOS.

Answer: This makes SimpleText (Code txt) to the frontmost application:

Example:

```
Dim A As AppleEvent  
A = NewAppleEvent("misc", "actv", "")  
If Not A.Send then  
Beep  
end if
```

Notes: (Code is Mac only)

5.0.67 How to catch Control-C on Mac or Linux in a console app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use SignalHandlerMBS class for this.

Example:

```
// watch for Control-C on Mac
call SignalHandlerMBS.SetFlagHandler(2)

dim ende as boolean = false
do
if SignalHandlerMBS.IsFlagSet(2) then
Print "Flag 2 set. Existing..."
ende = true
end if

DoEvents 1
loop until ende
```

Notes: The signal is caught, a flag is set and you can ask later in your normal application flow for the result.

5.0.68 How to change name of application menu?

Plugin Version: all, Platforms: macOS, Windows.

Answer: Use this code to change the application menu name on Mac OS X:

Example:

```
dim mb as new MenubarMBS
dim m as MenuMBS = mb.item(1) // 1 is in my tests the app menu
if m<>Nil then
m.MenuTitle = "Hello World"
end if
```

Notes: This code is for Carbon only.

5.0.69 How to change the name in the menubar of my app on Mac OS X?

Plugin Version: all, Platform: macOS.

Answer:

You mean it screws up if the file name of the bundle itself is different than the name of the executable file in the MacOS folder within the bundle? If so, you should find something like this within your Info.plist file (or the 'plst' resource that the RB IDE builds for you):

```
<key>CFBundleExecutable</key>
<string>Executable file name here</string>
```

Just make sure that file name matches.

However, if your question involves how you can change the name of the app that appears in the menu and the dock, that's different. You can make this name different from the file name by changing the CFBundleName key:

```
<key>CFBundleName</key>
<string>Name for menu here</string>
```

Note that if you use my free AppBundler program, this second part is taken care of for you – just fill in a custom name in the right field. You can find AppBundler (from Thomas Reed) at <http://www.bitjuggler.com/products/appbundler/>.

5.0.70 How to check if a folder/directory has subfolders?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this to check all items in a folder:

Example:

```
Function HasSubFolder(folder as FolderItem) As Boolean
dim c as Integer = folder.Count
```

```
for i as Integer = 1 to c
dim item as FolderItem = folder.TrueItem(i)
```

```
if item<>Nil and item.Directory then
Return true
end if
```

next

End Function

Notes: We use trueitem() here to avoid resolving alias/link files.
Also we check for nil as we may not have permission to see all items.
And if one is a directory, we return without checking the rest.

5.0.71 How to check if Macbook runs on battery or AC power?

Plugin Version: all, Platform: macOS.

Answer: Please use our IOPowerSourcesMBS class like this:

Example:

```
Function PowerSourceState() as Integer
dim p as new IOPowerSourcesMBS

// check all power sources
dim u as Integer = p.Count-1
for i as Integer = 0 to u
dim d as CFDictionaryMBS = p.Item(i)
if d<>nil then
// check if they have a power source state key:
dim o as CFObjectMBS = d.Value(NewCFStringMBS("Power Source State"))
if o isa CFStringMBS then
dim s as string = CFStringMBS(o).str

'MsgBox s

if s = "AC Power" then
Return 1
elseif s = "Battery Power" then
Return 2
end if
end if
end if
next
Return 0 // unknown
End Function
```

Notes: If you want to check the CFDictionaryMBS content, simply use a line like "dim x as dictionary = d.dictionary" and check the contents in the debugger.

5.0.72 How to check if Microsoft Outlook is installed?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you need Outlook for Scripting, you should simply check registry for the required Outlook.Application class:

Example:

```
Function OutlookInstalled() As Boolean
    #if TargetWin32 then

    try
    dim r as new RegistryItem("HKEY_CLASSES_ROOT\Outlook.Application\CLSID", false)

    Return true

    catch r as RegistryAccessErrorException
    // not installed
    Return false

    end try

    #else

    // Windows only, so false on other platforms
    Return false

    #endif

End Function
```

5.0.73 How to check on Mac OS which country or language is currently selected?

Plugin Version: all, Platform: macOS.

Answer: The code below returns a country value.

Example:

```
dim result as Integer

IF TargetMacOS THEN

CONST smScriptLang = 28
```

```

CONST smSystemScript = -1

DECLARE FUNCTION GetScriptManagerVariable LIB "Carbon" ( selector as Integer) as Integer
DECLARE FUNCTION GetScriptVariable LIB "Carbon" ( script as Integer, selector as Integer) as Integer

result=GetScriptVariable(smSystemScript, smScriptLang)

END IF

```

Notes: Returns values like:

For more values, check "Script.h" in the frameworks.

5.0.74 How to code sign my app with plugins?

Plugin Version: all, Platform: macOS.

Answer: When you try to code sign the application with plugin dylibs on Mac OS X, you may see error message that there is actually a signature included.

Notes: Please use the -f command line parameter with codesign utility to overwrite our MBS signature. We sign our plugins for MacOS, iOS and Windows to make sure they have not been modified.

In terminal, you do like this:

```
cd <Path to folder of app>
```

```

xattr -cr <Appname>.app
codesign -f -s "Developer ID Application: <Your Name>" <Appname>.app/Contents/Frameworks/*.dylib
codesign -f -s "Developer ID Application: <Your Name>" <Appname>.app/Contents/Frameworks/*.framework
codesign -f -s "Developer ID Application: <Your Name>" <Appname>.app

```

Please use the name of your certificate (See keychain), the name of your app and the path to the app folder. If you have helper apps you need to sign them first.

You can use a build step to automatically sign your app on build.

5.0.75 How to collapse a window?

Plugin Version: all, Platform: macOS.

Answer: Use this function (Mac only):

Example:

```
Sub CollapseRBwindow(w as window, CollapseStatus as boolean)
dim state, err as Integer
dim wh as MemoryBlock
```

```
Declare Function CollapseWindow Lib "Carbon" (window as Integer, collapse as Integer) as Integer
```

```
IF CollapseStatus THEN
state = 1
ELSE
state = 0
END IF
```

```
err = CollapseWindow(w.MacWindowPtr, state)
```

```
End Sub
```

Notes: Also the MBS Plugin has a window.collapsedmbs property you can set. For Windows the MBS Plugin has a window.isiconicmbs property.

5.0.76 How to compare two pictures?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can try this code:

Example:

```
Function ComparePictures(p as picture,q as picture) as Integer
dim r,u as RGBSurface
dim x,y,n,m,h,w as Integer
dim w1,w2,h1,h2,d1,d2 as Integer
dim c1,c2 as color
```

```
h1=p.Height
h2=q.Height
w1=p.Width
w2=q.Width
d1=p.Depth
d2=q.Depth
```

```
if d1<>d2 then
Return 1
elseif w1<>w2 then
```

```

return 2
elseif h1<>h2 then
Return 3
else
r=p.RGBSurface
u=q.RGBSurface

if r=nil or u=nil then
Return -1
else
h=h1-1
w=w1-1
m=min(w,h)

for n=0 to m
c1=r.Pixel(n,n)
c2=u.Pixel(n,n)
if c1<>c2 then
Return 4
end if
next

for y=0 to h
for x=0 to w
c1=r.Pixel(x,y)
c2=u.Pixel(x,y)
if c1<>c2 then
Return 5
end if
next
next

// 0 for equal
// -1 for error (no RGBsurface)
// 1 for different depth
// 2 for different width
// 3 for different height
// 4 for different pixels (fast test)
// 5 for different pixels (slow test)
end if
end if

Exception
Return -1
End Function

```

Notes: Remember that this only works on bitmap pictures, so the `picture.BitmapMBS` function may be useful.

5.0.77 How to compile PHP library?

Plugin Version: all, Platform: macOS.

Answer: You have to download the source code and compile a static version of the library.

Notes: This instructions were written based on PHP 5.2.6 on Mac OS X:

- Best take a new Mac with current Xcode version installed.
- Download the source code archive. e.g. "php-5.2.6.tar.bz2"
- Expand that archive on your harddisc.
- Open terminal window
- change directory to the php directory. e.g. "cd /php-5.2.6"
- execute this two lines to define the supported CPU types and the minimum Mac OS X version:
- export CFLAGS="-arch ppc -arch i386 -mmacosx-version-min=10.3"
- export CXXFLAGS="-arch ppc -arch i386 -mmacosx-version-min=10.3"
- the command "./configure help" does show the configure options.
- use configure with a line like this:
- ./configure --enable-embed --with-curl --enable-ftp --enable-zip --enable-sockets --enable-static --enable-soap --with-zlib --with-bz2 --enable-exif --enable-bcmath --enable-calendar
- start the compilation with "make all"
- other option is to use "make install" which first does the same as "make all" and than does some installation scripts.
- you may get an error about a duplicate symbole __yytext. Search the file "zend_ini_scanner.c", search a line with "char *yytext;" and change it to "extern char *yytext;"
- On the end you get a lot of error messages, but you have a working library (named libphp5.so) file in the invisible ".libs" folder inside your php source folder.

Possible problems and solutions:

- If the path to your files has spaces, you can get into trouble. e.g. "/RB Plugins/PHP" is bad as files will be searched sometimes in "/RB".

- If you have in /usr/local/lib libraries which conflict with the default libraries, you can get into trouble.
- If you installed some open source tools which compiled their own libraries, you can get into conflicts.
- if you have to reconfigure or after a problem, you may need to use "make clean" before you start "make all" again.

Feel free to install additional libraries and add more packages to the configure line.

5.0.78 How to convert a BrowserType to a String with WebSession.Browser?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this:

Example:

```
Function GetBrowserName(s as WebSession.BrowserType) As string
Select case s
case WebSession.BrowserType.Android
Return "Andriod"
case WebSession.BrowserType.Blackberry
Return "Blackberry"
case WebSession.BrowserType.Chrome
Return "Chrome"
case WebSession.BrowserType.ChromeOS
Return "ChromeOS"
case WebSession.BrowserType.Firefox
Return "Firefox"
case WebSession.BrowserType.InternetExplorer
Return "InternetExplorer"
case WebSession.BrowserType.Opera
Return "Opera"
case WebSession.BrowserType.Safari
Return "Safari"
case WebSession.BrowserType.SafariMobile
Return "SafariMobile"
case WebSession.BrowserType.Unknown
Return "Unknown"
else
Return "Unkown: "+str(integer(s))
end Select

End Function
```

5.0.79 How to convert a EngineType to a String with WebSession.Engine?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this:

Example:

```
Function GetRenderingEngineName(s as WebSession.EngineType) As string
Select case s
case WebSession.EngineType.Gecko
Return "Gecko"
case WebSession.EngineType.Presto
Return "Presto"
case WebSession.EngineType.Trident
Return "Trident"
case WebSession.EngineType.Unknown
Return "Unknown"
case WebSession.EngineType.WebKit
Return "WebKit"
else
Return "Unkown: "+str(integer(s))
end Select

End Function
```

5.0.80 How to convert a PlatformType to a String with WebSession.Platform?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this:

Example:

```
Function GetPlatformName(s as WebSession.PlatformType) As string
Select case s
case WebSession.PlatformType.Blackberry
Return "Blackberry"
case WebSession.PlatformType.iPad
Return "iPad"
case WebSession.PlatformType.iPhone
Return "iPhone"
case WebSession.PlatformType.iPodTouch
Return "iPodTouch"
case WebSession.PlatformType.Linux
Return "Linux"
case WebSession.PlatformType.Macintosh
Return "Macintosh"
```

```

case WebSession.PlatformType.PS3
Return "PS3"
case WebSession.PlatformType.Unknown
Return "Unknown"
case WebSession.PlatformType.WebOS
Return "WebOS"
case WebSession.PlatformType.Wii
Return "Wii"
case WebSession.PlatformType.Windows
Return "Windows"
else
Return "Unkown: "+str(integer(s))
end Select

```

End Function

5.0.81 How to convert a text to iso-8859-1 using the TextEncoder?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

This code can help you although it's not perfect.
You need to set lc to the current color you use.

Example:

```

dim outstring as string
dim theMac, thePC as textencoding
dim Mac2PC as textconverter

theMac = getTextEncoding(0) // MacRoman
thePC = getTextEncoding(&h0201) // ISOLatin1

Mac2PC = getTextConverter(theMac, thePC)
// if you wanted to do the opposite just create a converter
// PC2Mac = getTextConverter(thePC, theMac)

outstring = Mac2PC.convert("Bj√rn, this text should be converted")
Mac2PC.clear

```

Notes:

You have to call Mac2PC.clear after every conversion to reset the encoding engine.
See also newer TextConverterMBS class.

5.0.82 How to convert ChartTime back to Xojo date?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We have this example code:

Example:

```
Function ChartTimeToDate(ChartTime as Double) As date
static diff as Double = 0.0
```

```
if diff = 0.0 then
dim d2 as Double = CDBaseChartMBS.chartTime(2015, 1, 1)
dim da as new date(2015, 1, 1)
dim ts as Double = da.TotalSeconds
```

```
diff = ts - d2
end if
```

```
dim d as new date
d.TotalSeconds = diff + ChartTime
```

```
Return d
End Function
```

Notes: As you see we calculate the difference in base date from Date and ChartTime and later use difference to convert.

5.0.83 How to convert line endings in text files?

Plugin Version: all, Platform: macOS.

Answer: You can simply read file with TextInputStream and write with new line endings using TextOutputStream class.

Example:

```
dim inputfile as FolderItem = SpecialFolder.Desktop.Child("test.txt")
dim outputfile as FolderItem = SpecialFolder.Desktop.Child("output.txt")
dim it as TextInputStream = TextInputStream.Open(inputfile)
dim ot as TextOutputStream = TextOutputStream.Create(outputfile)
```

```
ot.Delimiter = EndOfLine.Windows // new line ending
while not it.EOF
ot.WriteLine it.ReadLine
wend
```

Notes: `TextInputStream` will read any input line endings and with `delimiter` property in `TextOutputStream` you can easily define your new delimiter.

5.0.84 How to convert picture to string and back?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use this plugin functions:

Notes: JPEG:

`JPEGStringToPictureMBS(buf as string)` as picture
`JPEGStringToPictureMBS(buf as string,allowdamaged as Boolean)` as picture
`PictureToJPEGStringMBS(pic as picture,quality as Integer)` as string

PNG:

`PictureToPNGStringMBS(pic as picture, gamma as single)` as string
`PictureToPNGStringMBS(pic as picture, mask as picture, gamma as single)` as string
`PictureToPNGStringMBS(pic as picture, gamma as single, Interlace as Boolean, FilterType as Integer)` as string
`PictureToPNGStringMBS(pic as picture, mask as picture, gamma as single, Interlace as Boolean, FilterType as Integer)` as string
`PNGStringToPictureMBS(data as string, gamma as single)` as picture
`PNGStringToPNGPictureMBS(data as string, gamma as single)` as PNGpictureMBS

Tiff:

`TIFFStringToPictureMBS(data as string)` as picture
`TIFFStringToTiffPictureMBS(data as string)` as TiffPictureMBS

BMP:

`BMPStringtoPictureMBS(data as string)` as picture
`Picture.BMPDataMBS(ResolutionValueDPI as Integer=72)` as string

GIF:

`GifStringToGifMBS(data as string)` as GIFMBS
`GifStringToPictureMBS(data as string)` as Picture

5.0.85 How to copy an array?

Plugin Version: all, Platform: macOS.

Answer: You can use a function like this to copy an array:

Example:

```
Function CopyArray(a() as Double) as Double()
dim r() as Double
for each v as Double in a
r.Append v
next
Return r
End Function
```

Notes: If needed make several copies of this method with different data types, not just double. For a deep copy of an array of objects, you need to change code to also make a copy of those objects.

5.0.86 How to copy a dictionary?

Plugin Version: all, Platform: macOS.

Answer: You can use a function like this to copy a dictionary:

Example:

```
Function CopyDictionary(d as Dictionary) As Dictionary
dim r as new Dictionary
for each key as Variant in d.keys
r.Value(key) = d.Value(key)
next
Return r
End Function
```

Notes: If needed make several copies of this method with different data types, not just double. For a deep copy of a dictionary of objects, you need to change code to also make a copy of those objects.

5.0.87 How to copy parts of a movie to another one?

Plugin Version: all, Platforms: macOS, Windows.

Answer: The code below copies ten seconds of the snowman movie to the dummy movie starting at the 5th second.

Example:

```

dim f as FolderItem
dim md as EditableMovie
dim ms as EditableMovie

f=SpecialFolder.Desktop.Child("Our First Snowman.mov")
ms=f.OpenEditableMovie

ms.SelectionStartMBS=5
ms.SelectionLengthMBS=10

f=SpecialFolder.Desktop.Child("dummy.mov")
md=f.CreateMovie

msgbox str(md.AddMovieSelectionMBS(ms))

```

Notes: If result is not 0, the method fails.

5.0.88 How to create a birthday like calendar event?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

// start a connection to the calendar database
dim s as new CalCalendarStoreMBS

// needed for the error details
dim e as NSErrorMBS

dim r as CalRecurrenceRuleMBS = CalRecurrenceRuleMBS.initYearlyRecurrence(1, nil) // repeat every
year without end

dim a as new CalAlarmMBS // add alarm
a.action = a.CalAlarmActionDisplay
a.relativeTrigger = -3600*24 // 24 Hours before

// create a new calendar
dim c as new CalEventMBS

dim d as new date(2011, 04, 20) // the date

dim calendars() as CalCalendarMBS = s.calendars

```

```

// set properties
c.Title="Test Birthday"
c.startDate=d
c.recurrenceRule = r
c.calendar=calendars(0) // add to first calendar
c.addAlarm(a)
c.endDate = d
c.isAllDay = true

// save event
call s.saveEvent(c,s.CalSpanAllEvents, e)
if e<>nil then
MsgBox e.localizedDescription
else
MsgBox "New event was created."
end if

```

Notes: This adds an event to iCal for the given date with alarm to remember you and repeats it every year.

5.0.89 How to create a GUID?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the UUIDMBS class for this.

5.0.90 How to create a Mac picture clip file?

Plugin Version: all, Platform: Windows.

Answer: You can use code like this one.

Example:

```

dim f As FolderItem
dim p As Picture

f=SpecialFolder.Desktop.Child("Test.pictClipping")
if f=nil then Return

p=new Picture(300,200,32) 'Make a sample picture
p.Graphics.ForeColor=RGB(0,255,255)
p.Graphics.FillOval 0,0,99,99

```

```
p.Graphics.ForeColor=RGB(255,0,0)
p.Graphics.DrawOval 0,0,99,99
```

```
dim r As ResourceFork 'ResourceFork is needed for a clip file
```

```
// Please define a file type Any
r=f.CreateResourceFork("Any")
```

```
// get PICT data using plugin function
dim pictdata as string = p.PicHandleDataMBS
r.AddResource(pictdata,"PICT",256,"Picture")
```

```
dim m as new MemoryBlock(8)
```

```
m.LittleEndian = false
m.Int16Value(0) = 0
m.Int16Value(2) = 0
m.Int16Value(4) = p.Width
m.Int16Value(6) = p.Height
```

```
r.AddResource(m,"RECT",256,"")
```

```
'Values taken from a sample file and irrelevant to the problem
```

```
dim data as string = DecodeBase64("AQAAAAAAAAAAAAAAAAACAFRDRVIAAABAAAAAAAAAAAAAAAAABUQ0IQAAAAA")
r.AddResource(data,"drag",128,"") 'ditto
r.Close
```

Notes: In general Apple has deprecated this, but a few application still support clippings.

5.0.91 How to create a PDF file in Xojo?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Check our DynaPDF plugin and the examples.

Notes: An alternative can be to use the CoreGraphics and Cocoa functions on Mac OS X. For Windows, we can only suggest our DynaPDF plugin.

5.0.92 How to create EmailAttachment for PDF Data in memory?

Plugin Version: all, Platform: macOS.

Answer: You can use code like the one below:

Example:

Function EmailAttachmentFromPDFData(PDFData as string, filename as string) As EmailAttachment
 dim a as new EmailAttachment

```
a.data = EncodeBase64(PDFData, 76)
a.ContentEncoding = "base64"
a.MIMETYPE = "application/pdf"
a.MacType = "PDF "
a.MacCreator = "prvw"
a.Name = filename
```

Return a

End Function

Notes: Compared to sample code from Xojo documentation, we set the mime type correct for PDF. The MacType/MacCreator codes are deprecated, but you can still include them for older Mac email clients. "prvw" is the creator code for Apple's preview app.

5.0.93 How to create PDF for image files?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use DynaPDF like this:

Example:

```
Function CreatePrintPDF(jpgFiles() as folderitem, pdfFile as FolderItem, PageWidth as Integer, PageHeight as Integer) As Boolean
  // have files?
  If pdfFile = Nil Then Return False
  If jpgFiles = Nil Then Return False

  If jpgFiles.Ubound < 0 Then Return False

  // new DynaPDF
  Dim pdf As New MyDynapdfMBS

  // page width/height in MilliMeter
  Dim pdfWidth as Integer = PageWidth * 72 / 25.4
  Dim pdfHeight as Integer = PageHeight * 72 / 25.4

  // put your license here
  Call pdf.SetLicenseKey "Starter"

  // create pdf
  Call pdf.CreateNewPDF pdfFile
```

```

// set a couple of options
Call pdf.SetPageCoords(MyDynaPDFMBS.kpcTopDown)
Call pdf.SetResolution(300)
Call pdf.SetUseTransparency(False)
Call pdf.SetSaveNewImageFormat(False)
Call pdf.SetGStateFlags(MyDynaPDFMBS.kgfUseImageColorSpace, False)
Call pdf.SetJPEGQuality(100)

// set page size
Call pdf.SetBBox(MyDynaPDFMBS.kpbMediaBox, 0, 0, pdfWidth, pdfHeight)
Call pdf.SetPageWidth(pdfWidth)
Call pdf.SetPageHeight(pdfHeight)

// append pages with one image per page
For i as Integer = 0 To jpgFiles.Ubound
Call pdf.Append
Call pdf.InsertImageEx(0, 0, pdfWidth, pdfHeight, jpgFiles(i), 1)
Call pdf.EndPage
Next

// close
Call pdf.CloseFile

Return True
End Function

```

Notes: This is to join image files in paper size to a new PDF.
e.g. scans in A4 into an A4 PDF.

5.0.94 How to CURL Options translate to Plugin Calls?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Below a few tips on how to translate command line CURL calls to plugin calls.

Notes: `curl -vX PUT http://localhost:5984/appserials/78569238475/DocumentRegister.docx?rev=3-25634563456 -data-binary @DocumentRegister.docx -H "Content-Type: application/msword"`

- The option `-v` means verbose. You can use `OptionVerbose` and listen for messages in the `DebugMessage` event.
- The option `-X PUT` means we want to do a HTTP PUT Request. So set `OptionPut` to true. Also you will want to set `OptionUpload` to true as you upload data.
- We have the URL which you put into `OptionURL` property.

- The `-data-binary` option tells CURL to pass the given data. With the `@` before the data, it is interpreted as a file name, so the data is read from the given file. You'll need to open this file and pass data with the Read event as needed. (See CURLS ftp file upload example project)
- The last option `-H` specifies an additional header for the upload. Pass this additional header with the `SetOptionHTTPHeader` method.

```
curl -X PUT http://127.0.0.1:5984/appserials/f2f4e540bf8bb60f61cfc4328001c59 -d '{ "type": "Product", "description": "Application Serial", "acronym": "AppSerial", "dateAdded": "2011-03-21 14:57:36" } '
```

- Option `-X PUT` like above.
- Pass the URL again in `OptionURL`
- This time data is passed in command line for CURL. You'd put this data in the quotes into a string and make it available in the Read event. (See CURLS ftp upload example project)

5.0.95 How to delete file with ftp and curl plugin?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can set post/pre quotes to have ftp commands executed before or after the download/upload.

Example:

```
dim d as CURLMBS // your curl object
```

```
// delete file
```

```
dim ws() As String
```

```
ws.Append "DELE Temp.txt"
```

```
d.SetOptionPostQuote(ws)
```

Notes: Use `SetOptionPostQuote`, `SetOptionPreQuote` or `SetOptionQuote`.

The ftp commands you pass here are native ftp commands and not the commands you use with ftp applications. To delete use `DELE` and the file path.

5.0.96 How to detect display resolution changed?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac OS X simply listen for display changed notifications.

Notes: Use the "Distribution Notification Center.rbp" example project as a base and use it to listen to notifications with the name "O3DeviceChanged".

5.0.97 How to detect retina?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use `Window.BackingScaleFactorMBS` to query the factor.

Example:

```
msgbox str(window1.BackingScaleFactorMBS)
```

5.0.98 How to disable force quit?

Plugin Version: all, Platform: macOS.

Answer:

Please visit this website and get the control panel for Mac OS 9 there:

<http://www3.sk.sympatico.ca/tinyjohn/DFQ.html>

For Mac OS X use the MBS Plugin with the `SetSystemUIModeMBS` method.

Notes:

Please use `presentationOptions` in `NSApplicationMBS` for Cocoa applications.

5.0.99 How to disable the error dialogs from Internet Explorer on javascript errors?

Plugin Version: all, Platform: Windows.

Answer: You can use this code in the `htmlviewer` open event:

Example:

```
if targetwin32 then
htmlviewer1._ole.Content.value("Silent") = True
end if
```

Notes: This disables the error dialogs from Internet Explorer.

5.0.100 How to display a PDF file in Xojo?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac OS X you can use CoreGraphics or PDFKit to display a PDF.

Notes: An alternative can be to load the PDF into a htmlviewer so the PDF plugin can display it.

On Windows you may need to use the Acrobat ActiveX control from Adobe or launch Acrobat Reader.

5.0.101 How to do a lottery in RB?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this function:

Example:

```

Sub Lotto(max as Integer,count as Integer,z() as Integer)
// Lotto count numbers of max put into the array z beginning at index 0
dim n(0) as Integer ' all the numbers
dim m as Integer ' the highest field in the current array
dim i,a,b,d as Integer ' working variables

'fill the array with the numbers
m=max-1
redim n(m)

for i=0 to m
n(i)=i+1
next

' unsort them by exchanging random ones
m=max*10
for i=1 to m
a=rnd*max
b=rnd*max

d=n(a)
n(a)=n(b)
n(b)=d
next

' get the first count to the dest array
m=count-1
redim z(m)
for i=0 to m
z(i)=n(i)
next

'sort the result
z.sort
End Sub

```



```

b=true
x=x1
while (x<x2) and (y<y2)
  ox=x
  oy=y

  x=x+dx
  y=y+dy

  if b then
    g.DrawLine ox,oy,x,y
  end if

  b=not b
wend

```

End Sub

Notes: It would be possible to add this to the plugin, but I think it's better if you do it in plain Xojo code, so it even works on Windows.

5.0.104 How to draw a nice antialiased line?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

This code can help you although it's not perfect.
You need to set lc to the current color you use.

Example:

```

Sub drawLine(xs as Integer, ys as Integer, xe as Integer, ye as Integer, face as RGBSurface, lineColor as
color)
  dim intX, intY, count, n, xDiff, yDiff as Integer
  dim v, v1, floatX, floatY, xx, yy, xStep, yStep as Double
  dim c as color

  const st=1.0

  xDiff=xe-xs
  yDiff=ye-ys
  count=max(abs(xDiff), abs(yDiff))
  xStep=xDiff/count
  yStep=yDiff/count

```

```

xx=xs
yy=ys
for n=1 to count
intX=xx
intY=yy
floatX=xx-intX
floatY=yy-intY

v=(1-floatX)*(1-floatY)*st
v1=1-v
c=face.pixel(intX, intY)
face.pixel(intX, intY)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)
v=floatX*(1-floatY)*st
v1=1-v
c=face.pixel(intX+1, intY)
face.pixel(intX+1, intY)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)
v=(1-floatX)*floatY*st
v1=1-v
c=face.pixel(intX, intY+1)
face.pixel(intX, intY+1)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)
v=floatX*floatY*st
v1=1-v
c=face.pixel(intX+1, intY+1)
face.pixel(intX+1, intY+1)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)

xx=xx+xStep
yy=yy+yStep
next

End Sub

```

Notes:

PS: st should be 1 and face should be a RGBSurface or a Graphics object.

5.0.105 How to dump java class interface?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: In terminal you can use "javap -s <classname>" to display the class with the method names and parameters.

Notes: For example show ResultSet class: javap -s java.sql.ResultSet

5.0.106 How to duplicate a picture with mask or alpha channel?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this function:

Example:

```
Function Duplicate(extends p as Picture) As Picture
#if RBVersion >= 2011.04 then
if p.HasAlphaChannel then

// create nw picture and copy content:
dim q as new Picture(p.Width, p.Height)
q.Graphics.DrawPicture p,0,0

Return q

end if
#endif

// create new picture
dim q as new Picture(p.Width, p.Height, 32)

// get mask
dim oldMask as Picture = p.mask(false)
if oldMask = nil then
// no mask, so simple copy
q.Graphics.DrawPicture p,0,0
Return q
end if

// remove mask
p.mask = nil

// copy picture and mask
q.Graphics.DrawPicture p, 0, 0
q.mask.Graphics.DrawPicture oldMask,0,0

// restore mask
p.mask = oldmask

Return q
End Function
```

Notes: Simply copy it to a module and call it like this: `q = p.duplicate`.

The code above works with old Xojo versions because of the `#if` even if your RS version does not support alpha channel pictures. This way it's future proof.

5.0.107 How to enable assistive devices?

Plugin Version: all, Platform: macOS.

Answer: You can use AppleScript code like below:

Notes: tell application "System Events"
activate

```
set UI elements enabled to true
```

```
return UI elements enabled
end tell
```

You can run this with AppleScriptMBS class.

5.0.108 How to encrypt a file with Blowfish?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this:

Example:

```
dim fi as FolderItem = SpecialFolder.Desktop.Child("test.xojo_binary_project")
dim fo as FolderItem = SpecialFolder.Desktop.Child("test.encrypted")
```

```
// read input
```

```
dim bi as BinaryStream = BinaryStream.Open(fi)
```

```
dim si as string = bi.Read(bi.Length)
```

```
bi.Close
```

```
// encrypt
```

```
dim so as string = BlowfishMBS.Encrypt("MyKey",si)
```

```
// write output
```

```
dim bo as BinaryStream = BinaryStream.Create(fo)
```

```
bo.Write so
```

```
bo.Close
```

Notes: Of course you can decrypt same way, just use Decrypt function and of course swap files.

5.0.109 How to extract text from HTML?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use both RemoveHTMLTagsMBS and DecodingFromHTMLMBS like this:

Example:

```
dim html as string = "<p><B>Gr&uuml;&szlig;e</B></P>"
dim htmltext as string = RemoveHTMLTagsMBS(html)
dim text as string = DecodingFromHTMLMBS(htmltext)
```

MsgBox text // shows: Gr√üë

Notes: You can use it together with RemoveHTMLTagsMBS to remove html tags. What you get will be the text without tags.

DecodingFromHTMLMBS turns HTML escapes back to unicode characters. Like ä to √§.

5.0.110 How to find empty folders in a folder?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this code:

Example:

```
dim folder as folderitem // your folder

dim c as Integer = folder.count
for i as Integer = 1 to c
dim item as folderitem = folder.trueitem(i)
if item = nil then
// ignore
elseif item.directory then
// folder
if item.count = 0 then
// found empty folder
end if
end if
next
```

5.0.111 How to find iTunes on a Mac OS X machine fast?

Plugin Version: all, Platform: macOS.

Answer: You can try Launch Services.

Example:

```
dim f as FolderItem

f=LaunchServicesFindApplicationForInfoMBS("hook","com.apple.iTunes","iTunes.app")

MsgBox f.NativePath
```

5.0.112 How to find network interface for a socket by it's name?

Plugin Version: all, Platform: macOS.

Answer: You can use our plugin to build a lookup table.

Example:

```
Function FindNetworkInterface(name as string) As NetworkInterface
name = name.trim

if name.len = 0 then Return nil

// search by IP/MAC
dim u as Integer = System.NetworkInterfaceCount-1
for i as Integer = 0 to u
dim n as NetworkInterface = System.GetNetworkInterface(i)
if n.IPAddress = name or n.MACAddress = name then
Return n
end if
next

// use MBS Plugin to build a mapping
dim interfaces() as NetworkInterfaceMBS = NetworkInterfaceMBS.AllInterfaces
dim map as new Dictionary

for each n as NetworkInterfaceMBS in interfaces
dim IPv4s() as string = n.IPv4s
dim IPv6s() as string = n.IPv6s

for each IPv4 as string in IPv4s
map.Value(IPv4) = n.Name
next
for each IPv6 as string in IPv6s
map.Value(IPv6) = n.Name
next
if n.MAC<>>" then
map.Value(n.MAC) = n.Name
```

```

end if
next

// now search interfaces by name, IPv4 or IPv6
for i as Integer = 0 to u
dim n as NetworkInterface = System.GetNetworkInterface(i)
if map.Lookup(n.IPAddress, "") = name then
Return n
end if

if map.Lookup(n.MACAddress, "") = name then
Return n
end if
next

End Function

```

Notes: The code above uses a lookup table build using NetworkInterfaceMBS class to find the network interface by name.

5.0.113 How to find version of Microsoft Word?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this:

Example:

```

// find Word
dim f as FolderItem = LaunchServicesFindApplicationForInfoMBS("", "com.microsoft.Word", "")

// open bundle
dim c as new NSBundleMBS(f)

// read info
dim d as Dictionary = c.infoDictionary

// show version
MsgBox d.Lookup("CFBundleVersion", "")

```

Notes: Older versions of Word can be found with creator code "MSWD".

5.0.114 How to fix CURL error 60/53 on connecting to server?

Plugin Version: all, Platform: macOS.

Answer: You probably connect with SSL and you have no valid certificate.

Example:

```
dim d as new CURLSMBS

// Disable SSL verification
d.OptionSSLVerifyHost = 0 // don't verify server
d.OptionSSLVerifyPeer = 0 // don't proofs certificate is authentic

// With SSL Verification:
dim cacert as FolderItem = Getfolderitem("cacert.pem")
d.OptionCAInfo = cacert.NativePath
d.OptionSSLVerifyHost = 2 // verify server
d.OptionSSLVerifyPeer = 1 // proofs certificate is authentic
```

Notes: You can either use the code above to disable the SSL verification and have no security. Or you use the cacert file and enable the verification. Than you only get a connection if the server has a valid certificate.

see also:

<http://curl.haxx.se/ca/>

5.0.115 How to format double with n digits?

Plugin Version: all, Platform: macOS.

Answer: You can use the FormatMBS function for this.

Example:

```
dim d as Double = 123.4567890
listbox1.AddRow FormatMBS("%f", d)
listbox1.AddRow FormatMBS("%e", d)
listbox1.AddRow FormatMBS("%g", d)

listbox1.AddRow FormatMBS("%5.5f", d)
listbox1.AddRow FormatMBS("%5.5e", d)
listbox1.AddRow FormatMBS("%5.5g", d)

d = 0.000000123456
listbox1.AddRow FormatMBS("%f", d)
listbox1.AddRow FormatMBS("%e", d)
```

```
listbox1.AddRow FormatMBS("%g", d)

listbox1.AddRow FormatMBS("%5.5f", d)
listbox1.AddRow FormatMBS("%5.5e", d)
listbox1.AddRow FormatMBS("%5.5g", d)
```

Notes: see FormatMBS for details.

In general %f is normal style, %e is scientific and %g is whichever gives best result for given space.

5.0.116 How to get a time converted to user time zone in a web app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the WebSession.GMTOffset property.

Example:

```
Sub Open()
// current date on server
dim d as new date
dim s as string = d.LongTime

// adjust to client GMT offset
d.GMTOffset = d.GMTOffset + Session.GMTOffset

dim t as string = D.LongTime

MsgBox s+EndOfLine+t
End Sub
```

5.0.117 How to get an handle to the frontmost window on Windows?

Plugin Version: all, Platform: Windows.

Answer: This function returns a handle for the frontmost window:

Example:

```
Function GetForegroundWindowHandle() as Integer
#if targetwin32 then
declare function GetForegroundWindow Lib "user32.dll" as Integer
Return GetForegroundWindow()
#endif
End Function
```

5.0.118 How to get CFAbsoluteTime from date?

Plugin Version: all, Platforms: macOS, Windows.

Answer: Use code like this:

Example:

```
dim d as new date
dim t as CFTimeZoneMBS = SystemCFTimeZoneMBS
dim g as new CFGregorianCalendarMBS
g.Day = d.Day
g.Month = d.Month
g.Year = d.Year
g.Minute = d.Minute
g.Hour = d.Hour
g.Second = d.Second

dim at as CFAbsoluteTimeMBS = g.AbsoluteTime(t)
dim x as Double = at.Value
```

```
MsgBox str(x)
```

Notes: As you see we need a timezone and put the date values in a gregorian date record. Now we can query absolute time for the given timezone.

5.0.119 How to get client IP address on web app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the WebSession.RemoteAddress property.

Example:

```
Sub Open()
Title = Session.RemoteAddress
End Sub
```

5.0.120 How to get fonts to load in charts on Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use the `SetFontSearchPath` method in the `CDBaseChartMBS` class to specify where your fonts are.

Example:

```

if TargetLinux then
CDBaseChartMBS.SetFontSearchPath "/usr/share/fonts/truetype;/usr/share/fonts/truetype/msttcorefonts"
else
// on Mac and Windows we use system fonts.
end if

// also you can later switch default fonts:

dim Chart as CDBaseChartMBS // your chart

#If TargetARM And TargetLinux Then
// use specific fonts on Linux on Raspberry Pi
Call Chart.setDefaultFonts("/usr/share/fonts/truetype/piboto/PibotoLt-Regular.ttf", "/usr/share/fonts/truetype/piboto/Pi
#EndIf

```

Notes: On macOS, iOS and Windows, the fonts are loaded from the system's font folder.

e.g. if you use ubuntu, you can install the `ttf-mscorefonts-installer` package and call this method with `"/usr/share/fonts/truetype/msttcorefonts"` as the path. No backslash on the end of a path, please.

5.0.121 How to get fonts to load in DynaPDF on Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use the `AddFontSearchPath` method in the `DynaPDFMBS` class to specify where your fonts are.

Example:

```

dim d as new DynaPDFMBS
if TargetLinux then
call d.AddFontSearchPath "/usr/share/fonts/truetype", true
else
// on Mac and Windows we use system fonts.
end if

```

Notes: On Mac OS X and Windows, the fonts are loaded from the system's font folder.

e.g. if you use ubuntu, you can install the `ttf-mscorefonts-installer` package and call this method with `"/usr/share/fonts/truetype/msttcorefonts"` as the path. No backslash on the end of a path, please.

5.0.122 How to get GMT time and back?

Plugin Version: all, Platform: macOS.

Answer: You can use the date class and the GMTOffset property.

Example:

```
// now
dim d as new date

// now in GMT
dim e as new date
e.GMTOffset = 0

// show
MsgBox str(d.TotalSeconds,"0.0")+ " " +str(e.TotalSeconds, "0.0")

dim GMTTimeStamp as Double = e.TotalSeconds

// restore
dim f as new date

// add GMT offset here
f.TotalSeconds = GMTTimeStamp + f.GMTOffset*3600
// because here it's removed
f.GMTOffset = f.GMTOffset

MsgBox d.ShortTime+" (" +str(d.GMTOffset)+") " +str(d.TotalSeconds,"0.0")+EndOfLine+_
e.ShortTime+" (" +str(e.GMTOffset)+") " +str(e.TotalSeconds,"0.0")+EndOfLine+_
f.ShortTime+" (" +str(f.GMTOffset)+") " +str(f.TotalSeconds,"0.0")
```

Notes: It's sometimes a bit tricky with the date class as setting one property often changes the others.

5.0.123 How to get good crash reports?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Check this website from the webkit website:

Notes: <http://webkit.org/quality/crashlogs.html>

5.0.124 How to get list of all threads?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use the runtime module like in this function:

Example:

```
Function Threads() As Thread()
#pragma DisableBackgroundTasks
dim t() as Thread

Dim o as Runtime.ObjectIterator=Runtime.IterateObjects
While o.MoveNext
if o.Current isa Thread then
t.Append thread(o.current)
end if
Wend

Return t
End Function
```

Notes: This returns an array of all thread objects currently in memory.

The pragma is important here as it avoids thread switches which may cause a thread to be created or deleted.

5.0.125 How to get parameters from webpage URL in Xojo Web Edition?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the Webpage.ParametersReceived event.

Example:

```
Sub ParametersReceived(Variables As Dictionary)
for each key as Variant in Variables.keys
MsgBox key+" ->" +Variables.Value(key)
next
End Sub
```

Notes: The text encodings of this strings is not defined in Xojo 2010r5. Please use DefineEncoding.

5.0.126 How to get the color for disabled textcolor?

Plugin Version: all, Platform: macOS.

Answer: Ask the appearance manager:

Example:

```
Function GetThemeTextColor(inColor as Integer, inDepth as Integer, inColorDev as Boolean) As Color
declare function GetThemeTextColor lib "Carbon" (inColor as Integer, inDepth as Integer, inColorDev as
Boolean, outColor as Ptr) as Integer
```

```
dim i as Integer
```

```
dim col as MemoryBlock
```

```
col = newMemoryBlock(6)
```

```
i = GetThemeTextColor(inColor, inDepth, inColorDev, col)
```

```
return RGB(col.UShort(0)\256, col.UShort(2)\256, col.UShort(4)\256)
```

```
End Function
```

Notes: The color for this is:

```
const kThemeTextColorDialogInactive = 2.
```

```
c = GetThemeTextColor(kThemeTextColorDialogInactive, Screen(0).Depth, true)
```

For Mac OS X you should use "CarbonLib" instead of "AppearanceLib" ...

5.0.127 How to get the current free stack space?

Plugin Version: all, Platform: macOS.

Answer: You can something like the code below:

Example:

```
Sub ShowStackSize()
```

```
dim threadid as Integer
```

```
dim size as Integer
```

```
declare function GetCurrentThread lib "Carbon" (byref threadid as Integer) as short
```

```
declare function ThreadCurrentStackSize lib "Carbon" (threadid as Integer, byref size as Integer) as short
```

```
if GetCurrentThread(threadid)=0 then
```

```
if 0=ThreadCurrentStackSize(threadid,size) then
```

```
MsgBox str(size)
```

```
end if
```

```
end if
```

End Sub

Notes: For Mac OS 9, use "ThreadLib" instead of "CarbonLib". You can use #if if you like for that.

5.0.128 How to get the current timezone?

Plugin Version: all, Platforms: macOS, Windows.

Answer:

You can use the TimeZoneMBS class or the CFTimeZoneMBS class.
Or code like below:

Example:

```
Function GMTOffsetInMinutes() as Integer
// Returns the offset of the current time to GMT in minutes.
// supports Mac OS and Windows, but not Linux yet (let me know if
// you have code for that, please)
//
// Note that the offset is not always an even multiple of 60, but
// there are also half hour offsets, even one 5:45h offset

// This version by Thomas Tempelmann (rb@tempel.org) on 25 Nov 2005
// with a fix that should also make it work with future Intel Mac targets.
//
// Using code from various authors found on the RB NUG mailing list

dim result, bias, dayLightbias as Integer
dim info as memoryBlock
dim offset as Integer

#if targetMacOS then

Declare Sub ReadLocation lib "Carbon" (location As ptr)

info = NewMemoryBlock(12)
ReadLocation info
if false then
// bad, because it does not work on Intel Macs:
'offset = info.short(9) * 256 + info.byte(11)
else
offset = BitwiseAnd (info.long(8), &hFFFFFF)
end

offset = info.short(9) * 256 + info.byte(11)
```

```

offset = offset \60
return offset

#endif

#if targetWin32 then

Declare Function GetTimeZoneInformation Lib "Kernel32" ( tzInfoPointer as Ptr ) as Integer
// returns one of
// TIME_ZONE_ID_UNKNOWN 0
// - Note: e.g. New Delhi (GMT+5:30) and Newfoundland (-3:30) return this value 0
// TIME_ZONE_ID_STANDARD 1
// TIME_ZONE_ID_DAYLIGHT 2

info = new MemoryBlock(172)
result = GetTimeZoneInformation(info)

bias = info.Long(0)
// note: the original code I found in the NUG archives used Long(84) and switched to Long(0)
// only for result=1 and result=2, but my tests found that Long(0) is also the right value for result=0

if result = 2 then
daylightBias = info.long(168)
end if
offset = - (bias + dayLightbias)
return offset

#endif

End Function

```

5.0.129 How to get the current window title?

Plugin Version: all, Platform: macOS.

Answer: The code below returns the current window title for the frontmost window on Mac OS X if Accessibility services are

Example:

```

Function CurrentWindowTitle() As string
// your application needs permissions for accessibility to make this work!

dim SystemWideElement,FocusedApplicationElement,FocusedWindowElement as AXUIElementMBS
dim FocusedApplication,FocusedWindow,Title as AXValueMBS
dim s as String
dim cs as CFStringMBS

```

```

SystemWideElement=AccessibilityMBS.SystemWideAXUIElement
if SystemWideElement<>nil then
FocusedApplication=SystemWideElement.AttributeValue(AccessibilityMBS.kAXFocusedApplicationAttribute)
if FocusedApplication.Type=AccessibilityMBS.kAXUIElementMBSTypeID then
FocusedApplicationElement=new AXUIElementMBS
FocusedApplicationElement.Handle=FocusedApplication.Handle
FocusedApplicationElement.RetainObject

FocusedWindow=FocusedApplicationElement.AttributeValue(AccessibilityMBS.kAXFocusedWindowAttribute)

if FocusedWindow<>nil and AccessibilityMBS.kAXUIElementMBSTypeID=FocusedWindow.Type then

FocusedWindowElement=new AXUIElementMBS
FocusedWindowElement.Handle=FocusedWindow.Handle
FocusedWindowElement.RetainObject

Title=FocusedWindowElement.AttributeValue(AccessibilityMBS.kAXTitleAttribute)
if Title<>nil and Title.Type=kCFStringMBSTypeID then
cs=new CFStringMBS
cs.handle=Title.Handle
cs.RetainObject
Return cs.str
end if
end if
end if
end if
End Function

```

5.0.130 How to get the cursor blink interval time?

Plugin Version: all, Platform: macOS.

Answer: On Mac OS you can use GetCaretTime from the toolbox.

Example:

```
declare function GetCaretTime lib "Carbon" () as Integer
```

```
MsgBox str(GetCaretTime()+ " ticks")
```

Notes: 60 ticks make one second.

5.0.131 How to get the list of the current selected files in the Finder?

Plugin Version: all, Platform: macOS.

Answer:

Use the AppleScript like this one:

```
tell application "finder"
return selection
end tell
```

Which translates into this AppleEvent:

```
Process("Finder").SendAE "core,getd,'—':obj { form:prop, want:type(prop), seld:type(sele), from:'null'() }
"
```

and as Xojo code it looks like this:

Example:

```
dim ae as appleEvent
dim o1 as appleEventObjectSpecifier
dim f as folderItem
dim alist as appleEventDescList
dim i as Integer
dim dateiname as string

// setup the AppleEvent
o1=getpropertyObjectDescriptor( nil, "sele")
ae= newappleEvent("core", "getd", "MACS")
ae.objectSpecifierParam("—")=o1

// send it
if ae.send then
// got the list
alist=ae.replyDescList

// now show the list of filename into an editfield:

for i=1 to alist.count
f=alist.folderItem(i)

dateiname=f.name
// editfield1 with property "multiline=true"!
editfield1.text=editfield1.text + dateiname + chr(13)
next
```

end if

5.0.132 How to get the Mac OS system version?

Plugin Version: all, Platform: macOS.

Answer: The following code queries the value and displays the version number:

Example:

```

dim first as Integer
dim second as Integer
dim third as Integer
dim l as Integer

if System.Gestalt("sysv",l) then

Third=Bitwiseand(l,15)
second=Bitwiseand(l\16,15)
first=Bitwiseand(l\256,15)+10*Bitwiseand(l\256\16,15)
end if

if First>=10 then
msgbox "Mac OS X "+str(First)+" "+str(Second)+" "+str(third)
else
msgbox "Mac OS "+str(First)+" "+str(Second)+" "+str(third)
end if

```

5.0.133 How to get the Mac OS Version using System.Gestalt?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

Dim s As String
Dim b As Boolean
Dim i, resp as Integer

// Systemversion
b = System.Gestalt("sysv", resp)
If b then
s = Hex(resp)

```

```

For i =Len(s)-1 DownTo 1
s=Left(s,i)+””+Mid(s,i+1)
Next
MsgBox ”Systemversion: Mac OS ” + s
end if

```

Notes: The MBS Plugin has a SystemInformationMBS.OSVersionString function for this.

5.0.134 How to get the screensize excluding the task bar?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Notes: Use the Screen class with the available* properties.

5.0.135 How to get the size of the frontmost window on Windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Notes: Make yourself a class for the WindowRect with four properties:

```

Bottom as Integer
Left as Integer
Right as Integer
Top as Integer

```

Add the following method to your class:

```

Sub GetWindowRect(windowhandle as Integer)
dim err as Integer
dim mem as memoryBlock
#if targetwin32 then
Declare Function GetWindowRect Lib ”user32.dll” (hwnd as Integer, ipRect As Ptr) as Integer

mem = newmemoryBlock(16)
err = GetWindowRect(windowhandle, mem)
Left = mem.long(0)
Top = mem.Long(4)
Right = mem.Long(8)
Bottom = mem.Long(12)

```

```
#endif  
End Sub
```

Good to use for the MDI Master Window!

5.0.136 How to get the source code of a HTMLViewer?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
// for Windows:
```

```
msgbox HTMLViewer1.IEHTMLTextMBS
```

```
// for MacOS with WebKit 2.x:
```

```
msgbox HTMLViewer1.WKWebViewMBS.HTMLText
```

5.0.137 How to get Xojo apps running Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You need to install some require packages.

Notes: You need CUPS as well as GTK packages. On 64 bit systems also the ia32-libs package.

Please note that you need a x86 compatible Linux. So no PPC, Power, ARM or other CPUs.

5.0.138 How to handle really huge images with GraphicsMagick or ImageMagick?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Sometimes it may be better to use an extra application to process images.

Notes: A typical 32 bit app made with Xojo can use around 1.8 GB on Windows and 3 GB on Mac OS X. Some images may be huge, so that processing them causes several copies of the image to be in memory. With a 500 MB image in memory, doing a scale or rotation may require a temp image. So with source, temp and dest images with each 500 MB plus your normal app memory usage, you may hit the limit of Windows with 1.8 GB.

In that case it may be worth running a tool like gm in the shell class. gm is the command line version of GraphicsMagick. There you can run the 64 bit version which is not limited in memory like your own application. Also you can monitor progress and keep your app responsive.

5.0.139 How to handle tab key for editable cells in listbox?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this function:

Example:

```
Function HandleTabInList(list as listbox, row as Integer, column as Integer, key as String) As Boolean
// Handle tab character in Listbox.CellKeyDown event
```

```
Select case asc(key)
case 9
if Keyboard.AsyncShiftKey then
// back

// look for column left
for i as Integer = column-1 downto 0
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next

// not found, so look in row before
row = row - 1
if row >= 0 then
for i as Integer = list.ColumnCount-1 downto 0
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next
end if
else
// forward

// look for column right
for i as Integer = column+1 to list.ColumnCount-1
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next
```

```

// not found, so look in row below
row = row + 1
if row <list.ListCount then
for i as Integer = 0 to list.ColumnCount-1
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next
end if
end if
end Select
End Function

```

Notes: You call it from CellKeyDown event like this:

```

EventHandler Function CellKeyDown(row as Integer, column as Integer, key as String) As Boolean
if HandleTabInList(me, row, column, key) then Return true
End EventHandler

```

As you see in the code, we handle tab and shift + tab for moving back and forward. Also we wrap to previous/next row if needed. Feel free to extend this to wrap from last to first row or create a new row for editing.

5.0.140 How to hard link MapKit framework?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Our MapKit classes weak link the framework. If you need hard linking it for the App Store, you can add this method to a class:

Example:

```

Sub ReferenceMapKit()
// just put this in window or app class

#if TargetMachO and Target64Bit then
Declare sub testing Lib "MapKit" Selector "test" (id as ptr)
testing(nil)
#endif

End Sub

```

Notes: No need to call the method.

Just having it in a window or app, will cause the compiler to hard link the framework.

5.0.141 How to have a PDF downloaded to the user in a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use a WebHTMLViewer control and load the PDF file with the PDF plugin from the browser.

Example:

```
dim CurrentFile as WebFile // a property of the WebPage

// define the PDF file
CurrentFile = new WebFile
CurrentFile.Filename = "test.pdf"
CurrentFile.MIMEType = "application/pdf"
CurrentFile.Data = "some pdf data" // MyDynaPDF.GetBuffer
CurrentFile.ForceDownload = true

// start the download
showurl(CurrentFile.url)
```

Notes: See our Create PDF example for the Xojo Web Edition.

5.0.142 How to hide all applications except mine?

Platform: macOS.

Answer: The code below will on Mac OS hide all applications except your one:

Example:

```
dim p as new ProcessMBS

p.GetFirstProcess
do
if not p.FrontProcess then
p.Visible=false
end if
loop until not p.GetNextProcess
```

5.0.143 How to hide script errors in HTMLViewer on Windows?

Plugin Version: all, Platform: Windows.

Answer: Set Internet Explorer to silent mode with code like this:

Example:

```
htmlviewer1._ole.Content.value("Silent") = True
```

Notes: Simply put this code in the open event of your htmlviewer control (using me instead of htmlviewer1).

5.0.144 How to hide the grid/background/border in ChartDirector?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you want to hide something in a chart, simply assign the kTransparent constant as color.

5.0.145 How to hide the mouse cursor on Mac?

Plugin Version: all, Platform: macOS.

Answer: Try this declare:

Example:

```
Declare Sub HideCursor Lib "Carbon" () Inline68K("A852")
```

```
HideCursor
```

Notes: The MBS Plugin has this function and supports it on Windows, too.

5.0.146 How to insert image to NSTextView or TextArea?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: With NSTextViewMBS you can use this code to insert file:

Example:

```
// insert a file to textview
```

```
Public Sub InsertFile(textview as NSTextViewMBS, f as FolderItem)
```

```
// read to file
```

```

dim b as BinaryStream = BinaryStream.Open(f)
dim s as string = b.Read(b.Length)

// build wrapper
dim fileWrapper as NSFileWrapperMBS = NSFileWrapperMBS.initRegularFileWithContents(s)
fileWrapper.preferredFilename = f.name

// make attachment
dim fileAttachment as new NSTextAttachmentMBS(fileWrapper)
dim attributedString as NSAttributedStringMBS = NSAttributedStringMBS.attributedStringWithAttachment(fileAttachment)

// add to a NSTextViewMBS
textview.insertText attributedString

End Sub

```

Notes: For TextArea you can query the underlying NSTextViewMBS object via TextArea.NSTextViewMBS method.

5.0.147 How to jump to an anchor in a htmlviewer?

Plugin Version: all, Platforms: macOS, Windows.

Answer: You can use javascript to change the current window's location.

Example:

```

// load website
htmlviewer1.LoadURL "http://www.monkeybreadsoftware.net/addressbook-abpersonmbs.shtml"

// later jump to anchor named "16":

if TargetWin32 then
call HTMLViewer1.IERunJavaScriptMBS "window.location = ""#16""
end if

```

5.0.148 How to keep a movieplayer unclickable?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: To keep the user away from clicking on a playing Movie you can just drop a Canvas in front of the Movieplayer and take the clicks there.

Example:

```
Function Canvas1.MouseDown(X as Integer, Y as Integer) as boolean
return true // take it and do nothing
End Function
```

5.0.149 How to keep my web app from using 100% CPU time?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Linux and MacOS you can use renice command in the terminal. On Windows use the task manager to reduce priority.

Notes: If you launch your app with nohup on Linux or Mac OS X like this from the terminal or a script:

```
nohup /webapps/MyApp/MyApp &
```

you can simply have a second line saying this:

```
renice 20 $ !
```

which tells the system to lower priority to lowest value for the latest background process.

5.0.150 How to kill a process by name?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can kill a process (or application) by name if you loop over all the processes and kill the one you need.

Example:

```
dim p as new ProcessMBS
p.GetfirstProcess ' get first
do
if p.name = "TextEdit" then
call p.KillProcess
Return
end if
loop until not p.GetNextProcess
```

Notes: You may want to check the result of killProcess function. Not every user is allowed to kill every application.

5.0.151 How to know how many CPUs are present?

Plugin Version: all, Platform: macOS.

Answer: Try this function:

Example:

```
Function GetCPUCount() as Integer
Declare Function MPPProcessors Lib "Carbon" () as Integer
```

```
Return MPPProcessors()
End Function
```

Notes: Your app will than need that library to launch on Classic. To avoid this the MBS plugin checks if this library is available and return 1 if it's not available.

5.0.152 How to know the calling function?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac you can use a helper function like this this code:

Example:

```
Public Function CallingFunction() as string
// Query name of calling function of a function
```

```
#Pragma BreakOnExceptions false
```

```
try
```

```
// raise a dummy exception
dim r as new NilObjectException
raise r
```

```
catch x as NilObjectException
```

```
// get stack
dim stack() as string = x.Stack
```

```
// pick function name and return
dim name as string = stack(2)
Return name
```

```
end try
End Function
```

Notes: You need to include function names in your application.

5.0.153 How to launch an app using it's creator code?

Plugin Version: all, Platform: macOS.

Answer: Send an AppleEvent "oapp" with the creator code to the Finder ("MACS"):

Example:

```
Dim a as AppleEvent
dim creator as string

creator = "MSIE" ' here the Internet Explorer

a = NewAppleEvent("aevt", "odoc", "MACS")
a.Timeout = -1

a.ObjectSpecifierParam("—") = GetUniqueIDObjectDescriptor("appf", nil, creator)

if not a.send then
msgBox "An error has occured"
else

end if
```

5.0.154 How to launch disc utility?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use this code:

Example:

```
dim f as FolderItem = LaunchServicesFindApplicationForInfoMBS("", "com.apple.DiskUtility", "")

if f<>Nil then
f.Launch
end if
```

Notes: This works even if people renamed the disc utility or moved it to another folder.

5.0.155 How to make a lot of changes to a REAL SQL Database faster?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You may try to embed your changes to the database between two transaction calls.

Example:

```
dim db as Database // some database

db.SQLiteExecute "BEGIN TRANSACTION"
// Do some Stuff
db.SQLiteExecute "END TRANSACTION"
```

Notes: This can increase speed by some factors.

5.0.156 How to make a NSImage object for my retina enabled app?

Plugin Version: all, Platform: macOS.

Answer: You can use code like this:

Example:

```
Function NewRetinaImage(pic as Picture, mask as Picture = nil) As NSImageMBS
// first make a NSImageMBS from it
dim n as new NSImageMBS(pic, mask)

// now set to half the size, so we have 2x pixels for the image
n.size = new NSSizeMBS(n.width/2, n.height/2)

// and return
Return n
End Function
```

Notes: The thing to do is to have 2x the pixels, but assign a size to the image which gives it the right size in points.

You can pass the NSImageMBS from here to NSMenuItemMBS. For Retina displays, the full resolution is used. For others it will be reduced.

5.0.157 How to make a window borderless on Windows?

Plugin Version: all, Platform: Windows.

Answer: Try this declares:

Example:

```
// Sets window to borderless popup type, and sets its initial dimensions.
// Call this method, then Win32SetBorderlessPos, and then RB's Show
// method. Use RB Frame type 7 (Global Floating Window).

Const SWP_NOMOVE = &H2
Const SWP_FRAMECHANGED = &H20
Const HWND_TOPMOST = -1
Const GWL_STYLE = -16
Const WS_POPUPWINDOW = &H80880000

Dim styleFlags as Integer

#If TargetWin32 Then

Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (hwnd as Integer, nIndex as Integer, dwNewLong as Integer) as Integer
Declare Function SetWindowPos Lib "user32" (hwnd as Integer, hWndInstertAfter as Integer, x as Integer, y as Integer, cx as Integer, cy as Integer, flags as Integer) as Integer

styleFlags = SetWindowLong( w.WinHWND, GWL_STYLE, WS_POPUPWINDOW )
styleFlags = BitwiseOr( SWP_FRAMECHANGED, SWP_NOMOVE )
styleFlags = SetWindowPos( w.WinHWND, HWND_TOPMOST, 0, 0, wd, ht, styleFlags )

#EndIf
```

5.0.158 How to make an alias using AppleEvents?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
Sub MakeAlias(folder as folderitem, target as folderitem, aliasname as string)
dim ev as AppleEvent
dim myResult as boolean
dim properties as AppleEventRecord

ev = NewAppleEvent("core", "crel", "MACS")
ev.MacTypeParam("kocl") = "alis"
ev.FolderItemParam("to ") = target
ev.FolderItemParam("insh") = folder

properties=new AppleEventRecord
```

```

properties.StringParam("pnam")=aliasname

ev.RecordParam("prdt")=properties

myResult = ev.send
// true on success, false on error
End Sub

```

Notes: Call it like this:

```
MakeAlias SpecialFolder.Desktop, SpecialFolder.Desktop.Child("Gif Copy.rb"), "test.rb alias"
```

Seems to not work on Mac OS X 10.6

5.0.159 How to make AppleScripts much faster?

Plugin Version: all, Platform: macOS.

Answer: use "ignoring application responses" like in this example:

```

Notes: on run { fn,fpx,fpy }
ignoring application responses
tell app "Finder" to set the position of folder fn to fpx,fpy
end ignoring
end run

```

5.0.160 How to make double clicks on a canvas?

Plugin Version: all, Platform: macOS.

Answer:

Update: Newer Xojo versions support DoubleClick event, so you don't need this code.

Here's my tip from the tips list on how to add a double-click event to the Canvas control. The technique could easily be used for a window or any Rectcontrol:

Because of its built-in drawing methods, the Canvas control is often used to create custom interface controls. But while the Canvas control has event handlers for most mouse events, it doesn't have an event handler for DoubleClick events. Fortunately, you can add a double-click event handler to a Canvas control easily. Basically, you're going to create a new class based on Canvas and add a double-click event to that. You can then use the new class anytime you need a Canvas with a double-click event.

To create a new Canvas class with a DoubleClick event handler, do this:

1. Add a new class to your project.
2. Set the Super property of the new class to "Canvas".
3. Change the name of this new class to "DoubleClickCanvas".

A double-click occurs when two clicks occur within the users double-click time (set in the Mouse control panel on both Macintosh and Windows) and within five pixels of each other. So, you'll need a few properties to store when and where the last click occurred.

4. Add a new property with the following declaration and mark it as private: lastClickTicks as Integer
5. Add a new property with the following declaration and mark it as private: lastClickX as Integer
6. Add a new property with the following declaration and mark it as private: lastClickY as Integer

Since the Canvas control doesn't have a DoubleClick event, you will need to add one.

7. Add a new event to your class by choosing New Event from the Edit menu and enter "DoubleClick" as the event name.

Double-clicks occur on MouseUp. In order for the mouseUp event to fire, you must return True in the MouseDown event.

8. In the MouseDown event, add the following code:
Return True

In the MouseUp event, you will need to determine what the users double-click time is. This value is represented on both the Mac and Windows in ticks. A tick is 1/60th of a second. Since there isn't a built-in function for this, you'll need to make a toolbox call. The mouseUp event code below makes the appropriate toolbox call for both Macintosh and Windows. It then compares the time of the users last click to the time of the current click and compares the location of the users last click to the location of the current click.

9. Add the following code to the MouseUp event:

```
dim doubleClickTime, currentClickTicks as Integer

#if targetMacOS then
Declare Function GetDbtTime Lib "Carbon" () as Integer
doubleClickTime = GetDbtTime()
#endif

#if targetWin32 then
Declare Function GetDoubleClickTime Lib "User32.DLL" () as Integer
```

```

doubleClickTime = GetDoubleClickTime()/60 // convert to ticks from milliseconds
#endif

currentClickTicks = ticks
//if the two clicks happened close enough together in time
if (currentClickTicks - lastClickTicks) <= doubleClickTime then
//if the two clicks occurred close enough together in space
if abs(X - lastClickX) <= 5 and abs(Y - LastClickY) <= 5 then
DoubleClick //a double click has occurred so call the event
end if
end if
lastClickTicks = currentClickTicks
lastClickX = X
lastClickY = Y

```

10. Now to test out your new DoubleClickCanvas, drag the class from the Project window to a window in your project to create an instance of it.

11. Double-click on the canvas you just added to your window to open the Code Editor. Notice that the canvas has a DoubleClick event handler. In this event handler, add the following code:
BEEP

5.0.161 How to make my Mac not sleeping?

Plugin Version: all, Platform: macOS.

Answer: Just inform the Mac OS about some system activity with code like this:

Example:

```

Sub UpdateSystemActivity()

#if TargetCarbon
declare function myUpdateSystemActivity lib "Carbon" alias "UpdateSystemActivity" (activity as Integer)
as short

const OverallAct = 0 // Delays idle sleep by small amount */
const UsrActivity = 1 // Delays idle sleep and dimming by timeout time */
const NetActivity = 2 // Delays idle sleep and power cycling by small amount */
const HDAActivity = 3 // Delays hard drive spindown and idle sleep by small amount */
const IdleActivity = 4 // Delays idle sleep by timeout time */

dim e as Integer

e=myUpdateSystemActivity(UsrActivity)

```

```
// you may react on an error if e is not 0 after the call.
```

```
#endif
End Sub
```

Notes: You may use another constant if you prefer some different behavior. Call it maybe every second.

5.0.162 How to make my own registration code scheme?

Plugin Version: all, Platform: Windows.

Answer: There are excellent articles about how to make a registration code scheme, but you can also simply use our RegistrationEngineMBS class.

Notes: If you need a license text, why not use the one from Xojo as a starting point?

5.0.163 How to make small controls on Mac OS X?

Plugin Version: all, Platform: macOS.

Answer: You can try this code on Mac OS X:

Example:

```

'/*
** Use the control's default drawing variant. This does not apply to
** Scroll Bars, for which Normal is Large.
**/
const kControlSizeNormal = 0

'/*
** Use the control's small drawing variant. Currently supported by
** the Check Box, Combo Box, Radio Button, Scroll Bar, Slider and Tab
** controls.
**/
const kControlSizeSmall = 1

'/*
** Use the control's small drawing variant. Currently supported by
** the Indeterminate Progress Bar, Progress Bar and Round Button
** controls.
**/
const kControlSizeLarge = 2

```

```

'/*
' * Control drawing variant determined by the control's bounds. This
' * ControlSize is only available with Scroll Bars to support their
' * legacy behavior of drawing differently within different bounds.
' */
const kControlSizeAuto = &hFFFF

const kControlSizeTag = "size"

declare function SetControlData lib "Carbon" (controlhandle as Integer, part as short, tagname as OS-
Type, size as Integer, data as ptr) as short

dim m as MemoryBlock

m=NewMemoryBlock(2)
m.UShort(0)=kControlSizeSmall

Title=str(SetControlData(CheckBox1.Handle, 0, kControlSizeTag, 2, m))

```

5.0.164 How to mark my Mac app as background only?

Plugin Version: all, Platform: macOS.

Answer: You can run a build script on each build with this code:

Example:

```

Dim App As String = CurrentBuildLocation + "/" + CurrentBuildAppName + ".app"
Call DoShellCommand("/usr/bin/defaults write " + App + "/Contents/Info ""NSUIElement"" YES")

```

Notes: This will set the NSUIElement flag to YES.

5.0.165 How to move a file or folder to trash?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like below:

Example:

```

Function MoveToTrash(f as FolderItem) As Boolean
#if TargetMacOS then
dim r as FolderItem
dim e as Integer = MacFileOperationMBS.MoveObjectToTrashSync(f, r, MacFileOperationMBS.kFSFile-
OperationDefaultOptions)

```

```

if e = 0 then
Return true // Ok
end if

#elseif TargetWin32 then
dim w as new WindowsFileCopyMBS

dim flags as Integer = w.FileOperationAllowUndo + w.FileOperationNoErrorUI + w.FileOperationSilent
+ w.FileOperationNoConfirmation
if w.FileOperationDelete(f, flags) then
Return true // OK
end if

flags = w.FileOperationNoErrorUI + w.FileOperationSilent + w.FileOperationNoConfirmation
if w.FileOperationDelete(f, flags) then
Return true // OK
end if
#else
// Target not supported
break
Return false
#endif
End Function

```

Notes: If you want to move a file to trash, you could use `f.movefileto f.trashfolder`, but that will overwrite existing files in the trash. You can use our `MacFileOperationMBS` class to move a file on Mac to the trash. And it uses the same code as the Finder, so files are renamed when the same name is already in use in the trash:

On Windows we use `WindowsFileCopyMBS` class.
Requires Mac OS X 10.5.

5.0.166 How to move an application to the front using the creator code?

Plugin Version: all, Platform: macOS.

Answer: This makes SimpleText (Code ttxt) to the frontmost application:

Example:

```

dim a as appleevent

a=newappleEvent("misc","actv","ttxt")

```

```
if a.send then
end if
```

Notes: (Code is Mac only)

5.0.167 How to move file with ftp and curl plugin?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can set post/pre quotes to have ftp commands executed before or after the download/upload.

Example:

```
dim d as CURLMBS // your curl object

// rename/move file
dim ws() As String
ws.Append "RNFR Temp.txt"
ws.append "RNT0 MyFile.txt"

d.SetOptionPostQuote(ws)
```

Notes: Use SetOptionPostQuote, SetOptionPreQuote or SetOptionQuote.

The ftp commands you pass here are native ftp commands and not the commands you use with ftp applications. So rename is two commands. First RNFR to tell where to rename from and second RNT0 with the new file name. To delete use DELE and the file path.

5.0.168 How to normalize string on Mac?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like below:

Example:

```
Function Normalize(t as string) As string
const kCFStringNormalizationFormD = 0 // Canonical Decomposition
const kCFStringNormalizationFormKD = 1 // Compatibility Decomposition
const kCFStringNormalizationFormC = 2 // Canonical Decomposition followed by Canonical Composition
const kCFStringNormalizationFormKC = 3 // Compatibility Decomposition followed by Canonical Composition

dim s as CFStringMBS = NewCFStringMBS(t)
dim m as CFMutableStringMBS = s.Normalize(kCFStringNormalizationFormD)
```

[Return m.str](#)
[End Function](#)

Notes: This uses Apple's CFString functions to normalize unicode variants.

5.0.169 How to obscure the mouse cursor on Mac?

Plugin Version: all, Platform: macOS.

Answer: Try this declare:

Example:

```
Declare Sub ObscureCursor Lib "Carbon" ()
```

```
ObscureCursor
```

Notes: The MBS Plugin has this function, but it's not supported for Windows.

5.0.170 How to open icon file on Mac?

Plugin Version: all, Platform: macOS.

Answer: Use the NSImageMBS class like this:

Example:

```
dim f as FolderItem = SpecialFolder.Desktop.Child("test.ico")  
dim n as new NSImageMBS(f)
```

```
window1.Backdrop = n.CopyPictureWithMask
```

5.0.171 How to open PDF in acrobat reader?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
dim pdf as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
```

```

// open PDF in Acrobat Reader on Mac:

// find app
dim bundleID as string = "com.adobe.Reader"
dim app as FolderItem = LaunchServicesFindApplicationForInfoMBS("", bundleID, "")

if app<>nil then

// launch app with parameters

dim docs() as FolderItem
docs.Append pdf

dim param as new LaunchServicesLaunchParameterMBS
param.Defaults = true
param.Application = app

dim x as FolderItem = LaunchServicesOpenXMBS(docs, param)

// on failure, simply launch it
if x = nil then
pdf.Launch(true)
end if

else
pdf.Launch(true)
end if

```

Notes: On Windows, simply use pdf.launch or WindowsShellExecuteMBS.

5.0.172 How to open printer preferences on Mac?

Plugin Version: all, Platform: macOS.

Answer: You can use our OpenMacOSXPreferencesPaneMBS function like this:

Example:

```

dim e as Integer = OpenMacOSXPreferencesPaneMBS("PrintAndFax")
if 0 = e then
MsgBox "OK"
elseif e = -43 then
MsgBox "File not found."
else
MsgBox "Error: "+str(e)
end if

```

5.0.173 How to open special characters panel on Mac?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We have functions for that in Cocoa and Carbon.

Example:

```
dim a as new NSApplicationMBS
a.orderFrontCharacterPalette
```

Notes: For Cocoa, you can use `orderFrontCharacterPalette` method in `NSApplicationMBS` class.

Or simply for Carbon and Cocoa the `ShowCharacterPaletteMBS` method.

5.0.174 How to optimize picture loading in Web Edition?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the `WebPicture` class.

Notes: Take your picture and create a `WebPicture` object. Store this `WebPicture` in a property of the `WebPage`, `Session` or `app` (as global as possible). On the first time you use this picture on an user session, the browser will load it. Second time you use it, the browser will most likely pick it from the cache.

Having pictures in `App` or some module reuses the same picture for all sessions which reduces memory footprint.

This does not work well with pictures you change very often or use only for one webpage on one user.

If you like to see an example, check our `Map` example.

5.0.175 How to parse XML?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this:

Example:

```
dim s as string = "<test><test /></test>"
```

```
try
```

```

dim x as new XmlDocument(s)
MsgBox "OK"
catch xe as XmlException
MsgBox "invalid XML"
end try

```

Notes: If you got an exception, you have a parse error.

5.0.176 How to play audio in a web app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use the HTML5 audio tag and control it with javascript.

Notes: This is just another example app I made today. It plays a christmas song. The audio file is provided by the application to the server, so no external web server is needed and this application can run stand alone. To compile and run you need Xojo 2010r5.

In the open event we search the audio files and open them as binarystreams. We create the two webfile objects. Those webfiles are part of the app class, so we have them globally. There we set the data with the content of our streams. We also define file names and mime types. They are needed so browser know what we have here:

```

audioFileM4V = new WebFile
audioFileM4V.Data = bM.Read(BM.Length)
audioFileM4V.Filename = "music.m4a"
audioFileM4V.MIMEType = "audio/m4a"

```

```

audioFileOGG = new WebFile
audioFileOGG.Data = bO.Read(BO.Length)
audioFileOGG.Filename = "music.ogg"
audioFileOGG.MIMEType = "audio/ogg"

```

Next in the open event of the webpage we have a PageSource control. The location is set to be before content. In the open event we define the html code for this. First we pick the URLs for the audio files. Than we build the html to use the audio tag. As you see, we give it an ID for later use and have it preload automatically. If you add an autoplay tag, you can have the audio play right away. Inside the audio tag we have two sources so we provide audio for both Firefox (OGG) and Safari (MPEG4). Finally we have a text to display if HTML5 audio tag is not supported.

You can set the source in the EditSource event:

```
dim urlO as string = app.audioFileOGG.URL
dim urlM as string = app.audioFileM4V.URL
me.Source = "<audio id=""mymusic"" preload=""auto""><source src="""+urlO+""" type=""audio/ogg""
/><source src="""+urlM+""" type=""audio/mpeg"" />Your browser does not support the audio ele-
ment.</audio>"
```

Next in the Play button we execute code to play the audio. This is a short javascript code which searches in the html document for the element with the ID "mymusic" which is the ID of our audio tag above. Once we got the object, we call it's play method to start playback.

```
me.ExecuteJavaScript("document.getElementById('mymusic').play();")
```

same for pause:

```
me.ExecuteJavaScript("document.getElementById('mymusic').pause();")
```

and finally for changing volume:

```
me.ExecuteJavaScript("document.getElementById('mymusic').volume="+str(me.Value/100.0)+"");")
```

5.0.177 How to pretty print xml?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the XML Transform method with the right XLS.

Notes: Learn more here:

<http://docs.xojo.com/index.php/XMLDocument.Transform>

5.0.178 How to print to PDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: This code below shows how to redirect printing to a PDF file on Mac OS X.

Example:

```
// get Xojo printer setup
dim p as new PrinterSetup

// now put it into NSPrintInfo to manipulate
dim n as new NSPrintInfoMBS
n.SetupString = p.SetupString
```

```

// change destination to file
dim f as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
n.SetSaveDestination(f)

// move back
p.SetupString = n.SetupString

// and print as usual
dim g as Graphics = OpenPrinter(p)
g.DrawString "Hello World", 20, 20

```

Notes: And you can use normal graphics class for that.

5.0.179 How to query Spotlight's Last Open Date for a file?

Plugin Version: all, Platform: macOS.

Answer: You can use a MDItemMBS objec to query this value:

Example:

```

Function LastOpenedDate(Extends F As FolderItem, DefaultOtherDates As Boolean = True) As Date
#If TargetMacOS Then
Dim xMDItem as New MDItemMBS(F)
Dim xDate as Variant

If xMDItem <>Nil Then
xDate = xMDItem.GetAttribute(xMDItem.kMDItemLastUsedDate).DateValue
If xDate IsA Date Then Return xDate
Else
If xDate <>Nil Then Break
End If
#EndIf

If DefaultOtherDates Then
If F.ModificationDate <>Nil Then Return F.ModificationDate
If F.CreationDate <>Nil Then Return F.CreationDate
End If
End Function

```

Notes: Thanks for Josh Hoggan for this example code.

5.0.180 How to quit windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
#if targetwin32 then
dim i1,i2,r as Integer
declare function ExitWindowsEx lib "user32" (uFlags as Integer, dwReserved as Integer) as Integer
i1 = 2
i2 = 0
r = ExitWindowsEx(i1,i2)
if r<>0 then
' Error()
end if

#endif
```

Notes: uFlags parameters:

```
'4 = EWX_Force
'0 = EWX_Logoff
'2 = EWX_Reboot
'1 = EWX_shutdown, should shut down computer
```

Also check the ExitWindowsMBS method.

5.0.181 How to read a CSV file correctly?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: With all the rules for quotes and delimiters, you can simply use the SplitCommaSeparatedValuesMBS method in our plugins like this:

Example:

```
dim f as FolderItem = SpecialFolder.Desktop.Child("test.csv")
dim t as TextInputStream = f.OpenAsTextFile

while not t.EOF
dim s as string = t.ReadLine(encodings.ASCII)

dim items() as string = SplitCommaSeparatedValuesMBS(s, ";", """")
```

```
List.AddRow """
dim u as Integer = UBound(items)
for i as Integer = 0 to u
List.Cell(List.LastIndex,i) = items(i)
next

wend
```

Notes: Please make sure you choose the right text encoding.

5.0.182 How to read the command line on windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
#if targetwin32 then
dim line as string
Dim mem as MemoryBlock

Declare Function GetCommandLineA Lib "kernel32" () As Ptr

mem=GetCommandLineA()
s=mem.cstring(0)

#endif
```

Notes: Newer Xojo versions have a system.commandline property.

5.0.183 How to render PDF pages with PDF Kit?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
// choose a file
dim f as FolderItem = SpecialFolder.Desktop.Child("test.pdf")

// open it as PDF Document
dim sourceFile as New PDFDocumentMBS(f)
```

```

if sourceFile.handle <>0 then // it is a PDF file

// get upper bound of pages
dim c as Integer = sourceFile.pageCount-1

// from first to last page
for n as Integer = 0 to c

// pick that page
dim page as PDFPageMBS = sourceFile.pageAtIndex(n)

// render to image
dim p as NSImageMBS = page.Render

// and convert to RB picture and display
Backdrop = p.CopyPictureWithMask

next

end if

```

Notes: PDFKit works only on Mac OS X.

5.0.184 How to restart a Mac?

Plugin Version: all, Platform: macOS.

Answer: Ask the Finder via Apple Events:

Example:

```

dim ae as appleevent
ae=newappleEvent("FNDR","rest","MACS")
if not ae.send then
msgBox "The computer couldn't be restarted."
end if

```

5.0.185 How to resume ftp upload with curl plugin?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: CURL supports that and you simply need to set the right options.

Notes: First of course OptionUpload must be true. Second OptionFTPAppend must be true so the OptionResumeFrom is used. Store there (or in OptionResumeFromLarge) your start value. Don't forget to implement the read event and return data there as requested.

5.0.186 How to rotate a PDF page with CoreGraphics?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: This code opens a PDF and draws the first page into a new PDF with 90–∞ rotation.

Example:

```
// Rotate a PDF page

// our files
dim sourcefile as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
dim destfile as FolderItem = SpecialFolder.Desktop.Child("rotated.pdf")

// open PDF
dim pdf as CGPDFDocumentMBS = sourcefile.OpenAsCGPDFDocumentMBS

// query media size of first page
dim r as CGRectMBS = pdf.MediaBox(1)

// create new PDF
dim c as CGContextMBS = destfile.NewCGPDFDocumentMBS(r,"title","Author","Creator")

// create rotated rectangle
dim nr as new CGRectMBS(0,0,r.Height,r.Width)

// create new page
c.BeginPage nr
c.SaveGState

const pi = 3.14159265

// rotate by 90–∞
c.RotateCTM pi*1.5

// fix origin
c.TranslateCTM -r.width,0

// draw PDF
c.DrawCGPDFDocument pdf,r,1

// cleanup
c.RestoreGState
c.EndPage
```

```
c = nil

// show in PDF viewer
destfile.Launch
```

Notes: This code is Mac only as it needs CoreGraphics.

5.0.187 How to rotate image with CoreImage?

Plugin Version: all, Platform: macOS.

Answer: Use the code like the one below:

Example:

```
// Rotate image with CoreImage

// load image
dim f as FolderItem = SpecialFolder.Desktop.Child("test.png")
dim image as new CIImageMBS(f)

// rotate 45 degree
dim n as new NSAffineTransformMBS
n.rotateByDegrees(45)

dim TransformFilter as new CIFilterAffineTransformMBS
TransformFilter.inputImage = image
TransformFilter.inputTransform = n

// get result
dim resultImage as CIImageMBS = TransformFilter.outputImage

// for saving to file
dim outputImage as NSImageMBS = resultImage.RenderNSImage(false)

f = SpecialFolder.Desktop.Child("output.png")
dim b as BinaryStream = BinaryStream.Create(f, true)
b.Write outputImage.PNGRepresentation

// as Xojo picture object for display
dim pic as Picture = outputImage.CopyPictureWithMask

Backdrop = pic
```

5.0.188 How to run a 32 bit application on a 64 bit Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Install 32 bit compatibility libraries.

Notes: The package is called ia32-libs for ubuntu (and others).

Some applications need to be run on a 32 bit system as they need some hardware related libraries. Like libUSB or libHID for USB devices.

5.0.189 How to save HTMLViewer to PDF with landscape orientation?

Plugin Version: all, Platform: macOS.

Answer: You can use NSPrintInfoMBS to change the options for PrintToPDFFile function.

Example:

```
// make it landscape
dim n as NSPrintInfoMBS = NSPrintInfoMBS.sharedPrintInfo
n.orientation = n.NSLandscapeOrientation

// save html to file
dim f as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
call HTMLViewer1.PrintToPDFFileMBS(f,10,30,10,30)
```

Notes: You may want to reset options later.
This code is only for Mac OS X.

5.0.190 How to save RTFD?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: With NSTextViewMBS you can use this code to save to RTFD:

Example:

```
// save text as RTFD including image attachments
dim f as FolderItem = GetSaveFolderItem(FileTypes1.ApplicationRtfd, "test.rtf")

if f = nil then Return

dim a as NSAttributedStringMBS = textView.textStorage
dim w as NSFileWrapperMBS = a.RTFDFileWrapperFromRange(0, a.length, DocumentAttributes)

dim e as NSErrorMBS
if w.writeToFile(f, e) then
```

```

else
MsgBox e.LocalizedDescription
end if

```

Notes: For TextArea you can query the underlying NSTextViewMBS object via TextArea.NSTextViewMBS method.

5.0.191 How to save RTFD?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: How to load PDF to htmlviewer on desktop?

Example:

```

Public Sub LoadPDFData(viewer as HTMLViewer, PDFData as string)
Dim base64string As String = EncodeBase64(PDFData)

// remove line endings to make it a big line
base64string = ReplaceLineEndings(base64string, "")

// build data URL
// https://en.wikipedia.org/wiki/Data_URI_scheme
Dim dataURL As String = "data:application/pdf;base64," + base64string

// show in webviewer
HTMLViewer1.LoadURL(dataURL)

// may not work everywhere due to URL length limit
// for Web projects, use WebFile instead!
End Sub

```

Notes: This avoids a temporary file, which may also work.
For Web Apps, please use WebFile.

5.0.192 How to scale a picture proportionally with mask?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: For a proportional scaling, we calculate the new picture size relative to the target maximum size.

Example:

```

Function ProportionalScaledWithMask(extends pic as Picture, Width as Integer, Height as Integer) As Picture
// Calculate scale factor

dim faktor as Double = min( Height / Pic.Height, Width / Pic.Width)

// Calculate new size
dim w as Integer = Pic.Width * faktor
dim h as Integer = Pic.Height * faktor

// create new picture
dim NewPic as new Picture(w,h,32)

// check if we have a mask and clear it
dim m as picture = pic.mask(False)
pic.mask = nil

// draw picture in the new size
NewPic.Graphics.DrawPicture Pic, 0, 0, w, h, 0, 0, Pic.Width, Pic.Height

if m <>nil then
// restore mask and scale it
pic.mask = m
NewPic.mask.Graphics.DrawPicture m, 0, 0, w, h, 0, 0, Pic.Width, Pic.Height
end if

// return result
Return NewPic
End Function

```

Notes: This version handles mask. As you see we actually have to remove mask in order to copy the picture part correctly.

5.0.193 How to scale a picture proportionally?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: For a proportional scaling, we calculate the new picture size relative to the target maximum size.

Example:

```

Function ProportionalScaled(extends pic as Picture, Width as Integer, Height as Integer) As Picture
// Calculate scale factor

dim faktor as Double = min( Height / Pic.Height, Width / Pic.Width)

```

```

// Calculate new size
dim w as Integer = Pic.Width * faktor
dim h as Integer = Pic.Height * faktor

// create new picture
dim NewPic as new Picture(w,h,32)

// draw picture in the new size
NewPic.Graphics.DrawPicture Pic, 0, 0, w, h, 0, 0, Pic.Width, Pic.Height

// return result
Return NewPic
End Function

```

Notes: This does not handle mask, but you can scale the mask the same way and assign it to the new picture.
(see other FAQ entry with mask)

5.0.194 How to scale/resize a CIIImageMBS?

Plugin Version: all, Platform: Windows.

Answer: Use the CIFilterLanczosScaleTransform filter to scale down a picture to a specific size.

Example:

```

Dim pic As Picture = LogoMBS(500)
Dim image As CIIImageMBS = CIIImageMBS.imageWithPicture(pic)

Dim filter As New CIFilterLanczosScaleTransformMBS

Const targetWidth = 600.0
Const targetHeight = 400.0

Dim scale As Double = targetHeight / image.Extent.Height
Dim aspect As Double = targetWidth / (image.Extent.Width * scale)

filter.inputImage = image
filter.inputScale = scale
filter.inputAspectRatio = aspect

Dim result As Picture = filter.outputImage.RenderPicture

Backdrop = result

```

Notes: This is same code as our scaleTo convenience method.

5.0.195 How to scale/resize a picture?

Plugin Version: all, Platform: Windows.

Answer: There are several ways to scale or resize a picture. The easiest way may be the ScaleMBS function in the Picture class.

Example:

```
dim Original,Scaled as Picture
```

```
Original=LogoMBS(500)
Scaled=Original.ScaleMBS(100,100,true)
```

Notes: The plugin ways:

- GraphicsMagick can scale/resize.
- CoreImage scale filter may result in the fastest and best images on Mac OS X 10.4.
- NSImageMBS can scale, but is Mac OS X only.
- CGImageMBS can scale, but is Mac OS X only.
- CIImageMBS can scale, but is Mac OS X only.
- QuickTime Graphics exporter and importer can be connected to scale. (this was used more often a few years ago)
- ImageMagick can scale very nice and crossplatform. But the ImageMagick libraries are big.
- The picture.ScaleMBS function is self written and results in equal output on Mac, Windows and Linux without any additional libraries installed.
- Picture.ScalingMBS does crossplatform scaling with several modes.

with pure Xojo:

- make a new picture and draw the old one with new size inside.

5.0.196 How to search with regex and use unicode codepoints?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can specify unicode characters in search string with backslash x and digits.

Example:

```
dim r as RegExMbs
dim s as string
dim c as Integer
```

```

s="123 √$√√° ABC 456"

r=new RegExMBS
if r.Compile("√") then
c=r.Execute(s,0)
MsgBox str(c)+" "+str(r.Offset(0))+" "+str(r.Offset(1))
// shows: 1 4 10
// 1 for ubound of the offset array
// 4 for 4 bytes before the matched pattern
// 10 for the 10 bytes before the end of the matched pattern
end if

r=new RegExMBS
if r.Compile("\xF6") then // finds √ using Unicode codepoint
c=r.Execute(s,0)
MsgBox str(c)+" "+str(r.Offset(0))+" "+str(r.Offset(1))
// shows: 1 4 10
// 1 for ubound of the offset array
// 4 for 4 bytes before the matched pattern
// 10 for the 10 bytes before the end of the matched pattern
end if

```

5.0.197 How to see if a file is invisible for Mac OS X?

Plugin Version: all, Platform: macOS.

Answer: Try this function:

Example:

```

Function Invisible(F As FolderItem) As Boolean
Dim TIS As TextInputStream
Dim S,All As String
Dim I as Integer
dim g as folderitem

If Left(F.Name,1)="." or not f.visible Then
Return True
End If

g=F.Parent.Child(".hidden")
If g.Exists Then
TIS=g.OpenAsTextFile
if tis<>Nil then
All=TIS.ReadAll
For I=1 to CountFields(All,Chr(11))
S=NthField(All, Chr(11), I)

```

```

If S=F.name Then
Return True
End If
Next
end if
End if
End Function

```

5.0.198 How to set cache size for SQLite or REALSQLDatabase?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You use the pragma cache_size command on the database.

Example:

```

// set cache size to 20000 pages which is about 20 MB for default page size
dim db as REALSQLDatabase
db.SQLExecute "PRAGMA cache_size = 20000"

```

Notes: Default cache size is 2000 pages which is not much.

You get best performance if whole database fits in memory.

At least you should try to have a cache big enough so you can do queries in memory.

You only need to call this pragma command once after you opened the database.

5.0.199 How to set the modified dot in the window?

Plugin Version: all, Platform: macOS.

Answer: Try this declares:

Example:

```

window1.ModifiedMBS=true

```

5.0.200 How to show a PDF file to the user in a Web Application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use a WebHTMLViewer control and load the

Example:

```

dim CurrentFile as WebFile // a property of the WebPage

// define the PDF file
CurrentFile = new WebFile
CurrentFile.Filename = "test.pdf"
CurrentFile.MIMEType = "application/pdf"
CurrentFile.Data = "some pdf data" // MyDynaPDF.GetBuffer

// load into html viewer
HTMLViewer1.URL = CurrentFile.URL

```

Notes: See our Create PDF example for the Xojo Web Edition.

5.0.201 How to show Keyboard Viewer programmatically?

Platform: macOS.

Answer: Use Xojo or AppleScript to launch the KeyboardViewerServer.app.

Example:

```

dim a as new AppleScriptMBS
dim text as string
dim lines(-1) as string

lines.append "set theApplication to ""KeyboardViewerServer""
lines.append "set thePath to ""/System/Library/Components/KeyboardViewer.component/Contents/Shared-
Support/KeyboardViewerServer.app""
lines.append ""
lines.append "set POSIXPath to ((POSIX file thePath) as string)"
lines.append "tell application ""System Events"" to set isRunning to 0 <(count (application processes whose
name is theApplication))"
lines.append "if isRunning then tell application POSIXPath to quit"
lines.append "delay 0.15"
lines.append ""
lines.append "ignoring application responses"
lines.append " tell application POSIXPath to run"
lines.append "end ignoring"

text=join(lines,EndOfLine.macintosh)

a.Compile text
a.Execute

```

Notes: AppleScript code:

```
set theApplication to "KeyboardViewerServer"
set thePath to "/System/Library/Components/KeyboardViewer.component/Contents/SharedSupport/KeyboardViewerServer.app"
```

```
set POSIXPath to ((POSIX file thePath) as string)
tell application "System Events" to set isRunning to 0 <(count (application processes whose name is theApplication))
if isRunning then tell application POSIXPath to quit
delay 0.15
```

```
ignoring application responses
tell application POSIXPath to run
end ignoring
```

5.0.202 How to show the mouse cursor on Mac?

Plugin Version: all, Platform: macOS.

Answer: Try this declare:

Example:

```
Declare Sub ShowCursor Lib "Carbon" ()
```

```
ShowCursor
```

Notes: The MBS Plugin has this function and supports it on Windows, too.

5.0.203 How to shutdown a Mac?

Plugin Version: all, Platform: macOS.

Answer: Ask the Finder via Apple Events:

Example:

```
dim ae as appleevent
ae=newappleEvent("FNDR","shut","MACS")
if not ae.send then
msgBox "The computer couldn't be shutdown."
end if
```

Notes: Or toolbox call (Attention: This method will stop the computer immediatly: No document asked to be saved, all applications quitting without knowing).

```
Declare Sub ShutDwnPower Lib "Carbon" ()
ShutDwnPower
```

5.0.204 How to sleep a Mac?

Plugin Version: all, Platform: macOS.

Answer: Ask the Finder via Apple Events:

Example:

```
dim ae as appleevent
ae=newappleEvent("FNDR","slep","MACS")
if not ae.send then
msgBox "The computer doesn't want to sleep."
end if
```

5.0.205 How to speed up rasterizer for displaying PDFs with DynaPDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Here a few speed tips:

Notes:

- Use the DynaPDFRasterizerMBS function instead of our render functions.
- Reuse DynaPDFRasterizerMBS as long as the target picture size doesn't change.
- Import only the PDF pages you want to display.
- Let DynaPDF do zooming, rotating or other effects instead of you change it.

5.0.206 How to use PDFLib in my RB application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The PDFlib plugin was discontinued in favor of our DynaPDF plugin.

Notes: If you need help to move, please contact us.

5.0.207 How to use quotes in a string?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Just double them.

Example:

```
msgbox "This String contains ""quotes"""
```

5.0.208 How to use Sybase in Web App?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use our MBS Xojo SQL Plugin to connect to a Sybase Database in your web application.

Notes: If you see db.Connect giving the error message "cs_ctx_alloc ->CS_MEM_ERROR", than some things are not setup right for Sybase.

The Apache process may not have all the SYBASE environment variables being set when the CGI was launched.

Adding these lines to /etc/httpd/conf/httpd.conf stopped the faux memory errors for us:

```
SetEnv LD_LIBRARY_PATH /opt/sybase/OCS-15_0/lib:/opt/sybase/OCS-15_0/lib3p64:/opt/sybase/OCS-15_0/lib3p:
SetEnv SYBROOT /opt/sybase
SetEnv SYBASE_OCS /opt/sybase
SetEnv SYBASE /opt/sybase
```

5.0.209 How to use the Application Support folder?

Plugin Version: all, Platform: macOS.

Answer:

I was saving a registration code for an app to the Preference folder. People on the list have suggested that it would be better in the ApplicationSupportFolder. How do I save the file called CWWPrefs into that folder using MBS?

I have checked for examples and the docs but can't see how to apply it

```
//f = SpecialFolder.Preferences.child("CWWPrefs")
f = ApplicationSupportFolderMBS(-32768)
```

Example:

```

dim folder,file as FolderItem

folder = createApplicationSupportFolderMBS(-32763)

if folder=nil then
// Some very old Mac OS Versions may not support it
// or the plugin may fail for any reason
folder=SpecialFolder.Preferences
end if

file=folder.Child("CWWPrefs")

MsgBox file.NativePath

```

Notes:

You may not be able to write there with a normal user account!

5.0.210 How to use the IOPMCopyScheduledPowerEvents function in Xojo?

Plugin Version: all, Platform: macOS.

Answer: You can use the following code which does this using the SoftDeclareMBS class.

Example:

```

Sub Open()
dim c as CFDateMBS
dim t as CFAbsoluteTimeMBS

// get current date
c=NewCFDateMBS

// in absolute time (seconds since x)
t=c.AbsoluteTime

// add 600 seconds (= 10 Minutes)
t.Value=t.Value+600

// Make a Date from it
c=t.Date

// Schedule the event
// 0 on success
// E00002C1 for missing root rights

```

```

Title=hex(schedulePowerEvent(c, "wake"))

// Just for information, display the scheduled stuff
CFShowMBS CopyScheduledPowerEvents
End Sub

Function CopyScheduledPowerEvents() As carrayMBS
dim s as SoftDeclareMBS
dim m as MemoryBlock

s=new SoftDeclareMBS

if s.LoadLibrary("IOKit.framework") then
if s.LoadFunction("IOPMCopyScheduledPowerEvents") then
if s.CallFunction(0,nil) then
Return NewCFArrayMBSHandle(s.Result,true)
else
MsgBox "Failed to Call IOPMCopyScheduledPowerEvents."
end if
else
MsgBox "Failed to load IOPMCopyScheduledPowerEvents."
end if
else
MsgBox "Failed to load IOKit."
end if

Return nil
End Function

Function SchedulePowerEvent(time_to_wake as CFDateMBS, Type as CFStringMBS) as Integer
dim s as SoftDeclareMBS
dim m as MemoryBlock

'/*
'* Types of power event
'* These are potential arguments to IOPMSchedulePowerEvent().
'* These are all potential values of the kIOPMPowerEventTypeKey in the CFDictionaryes
'* returned by IOPMCopyScheduledPowerEvents().
'*/
'/*!
'@define kIOPMAutoWake
'@abstract Value for scheduled wake from sleep.
'*/
'#define kIOPMAutoWake "wake"
,
'/*!
'@define kIOPMAutoPowerOn
'@abstract Value for scheduled power on from off state.

```

```

*/
`#define kIOPMAutoPowerOn "poweron"
,
`/*!
`@define kIOPMAutoWakeOrPowerOn
`@abstract Value for scheduled wake from sleep, or power on. The system will either wake OR
`power on, whichever is necessary.
*/
,
`#define kIOPMAutoWakeOrPowerOn "wakepoweron"
`/*!
`@define kIOPMAutoSleep
`@abstract Value for scheduled sleep.
*/
,
`#define kIOPMAutoSleep "sleep"
`/*!
`@define kIOPMAutoShutdown
`@abstract Value for scheduled shutdown.
*/
,
`#define kIOPMAutoShutdown "shutdown"

s=new SoftDeclareMBS

if s.LoadLibrary("IOKit.framework") then
if s.LoadFunction("IOPMSchedulePowerEvent") then

m=NewMemoryBlock(12)
m.Long(0)=time_to_wake.handle
m.Long(4)=0 // nil
m.Long(8)=type.Handle

if s.CallFunction(3,m) then
Return s.Result
end if
end if
end if

End Function

```

Notes: Requires Mac OS X and to execute root rights.

5.0.211 How to validate a GUID?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use this function below which uses a regular expression to verify that the string is a valid UUID/GUID:

Example:

```
Function IsGUID(guid as string) As Boolean
dim r as new RegEx
```

```
r.SearchPattern = "^(\{ { 0,1 } ( [ 0-9a-fA-F ] ) { 8 } -( [ 0-9a-fA-F ] ) { 4 } -( [ 0-9a-fA-F ] ) { 4 } -( [ 0-9a-fA-F ] ) { 4 } -( [ 0-9a-fA-F ] ) { 12 } \} { 0,1 } )$ "
```

```
Return r.Search(guid)<>nil
End Function
```

Notes: Simply parsing the GUID with CFUUIDMBS does not give the same result as CFUUIDMBS will also take a string like "DDDD".

5.0.212 How to walk a folder hierarchie non recursively?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this one:

Example:

```
Sub Walk(folder as FolderItem)
dim folders() as FolderItem
```

```
folders.Append folder
```

```
while UBound(folders)>=0
```

```
dim currentFolder as FolderItem = folders.pop
```

```
dim c as Integer = currentFolder.Count
```

```
for i as Integer = 1 to c
```

```
dim item as FolderItem = currentFolder.TrueItem(i)
```

```
if item = Nil then
```

```
// no permission
```

```
elseif item.Visible then // only visible
```

```
if item.Directory then
```

```
folders.Append item
```

```
else
// work with file here
end if

end if

next

wend
End Sub
```

Notes: As you see we go with a long loop which runs until we don't have more folders to process. We ignore items we can't access due to permission limits. And we only work visible items. If you like, check `folderitem.isBundleMBS` on item to handle packages and applications better on Mac OS X.

5.0.213 I got this error: PropVal, QDPictMBS.Name (property value), Type mismatch error. Expected CGDataProviderMBS, but got Variant, Name:QDPictMBS

Plugin Version: all, Platform: macOS.

Answer: The plugins MacOSX and MacOSXCF belong together. If you use one part, please also install the other part.

Notes: We splitted the plugin because the Xojo IDE on Windows crashed on compilation.

5.0.214 I registered the MBS Plugins in my application, but later the registration dialog is shown.

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: There are two main reasons.

Notes: 1. you may use the plugin before registering them. This is often the case if you register in a window open event and use the plugin in a control open event.

On the console on Mac OS X or Windows, you may see a message like this "MBS Plugins were used by the application before the RegisterMBSPlugin function was called. Please fix this in your code!".

2. you may have mixed different plugin versions which are not compatible.

In this case you can see a message "Internal plugin registration error." on the console on Mac OS X. Newer plugins may show a message dialog reporting this. Older version simply think they are not registered.

If the installer just merges old and new applications, users may have libraries of older and newer plugin versions in the libs folder. If your application loads the wrong version, the registration fails.

If you use remote debugging, make sure you clear the temporary files there, too. Otherwise you may have old DLLs on your hard disc which may disturb your application.

You can run into issues if you use your registration code on different places of your app. Please register only once in app.open (or app Constructor). If you have several codes, simply call them one after the other.

Also check that you only call RegisterMBSPlugin with valid serial number. If you later call RegisterMBSPlugin with Demo like in example code above, you remove the license.

Next check if you can clear the Xojo caches and that helps. This includes the Xojo Scratch folder and the Plugins & Project caches. Simply locate those folders and delete them. For Windows look in hidden AppData folder in your user folder. For Mac, please check textasciitilde /Library/Caches and your temp folders.

Finally make sure you use the right serial number. Not an older one or a misspelled one.

5.0.215 I want to accept Drag & Drop from iTunes

Plugin Version: all, Platform: macOS.

Answer: You need to accept AcceptMacDataDrop "itun" and Handle the DropObject.

Example:

```
Sub Open()
window1.AcceptMacDataDrop "itun"
End Sub
```

```
Sub DropObject(obj As DragItem)
dim s as string
dim f as folderItem
dim d as CFDictionaryMBS
dim o as CFObjectMBS
dim key as CFStringMBS
dim dl as CFDictionaryListMBS
dim i,c as Integer
dim u as CFURLMBS
dim file as FolderItem
```

```
if obj.MacDataAvailable("itun") then
s = obj.MacData("itun")
```

```
// Parse XML
o=NewCFOBJECTMBSFromXML(NewCFBinaryDataMBSStr(s))

// Make dictionary
if o isa CFDictionaryMBS then
d=CFDictionaryMBS(o)

// get Tracks Dictionary
key=NewCFStringMBS("Tracks")
o=d.Value(key)

if o isa CFDictionaryMBS then
d=CFDictionaryMBS(o)
dl=d.List

// Walk over all entries in the Tracks dictionary
c=dl.Count-1
for i=0 to c
o=dl.Value(i)

if o isa CFDictionaryMBS then
d=CFDictionaryMBS(o)

key=NewCFStringMBS("Location")
o=d.Value(key)
if o isa CFStringMBS then
u=NewCFURLMBS CFStringMBS(o),nil

file=u.file
if file<>nil then
MsgBox file.NativePath
end if
end if
end if
next
end if
end if
end if
End Sub
```

Notes: The code above inside a window on Xojo 5.5 with MBS Plugin 5.3 will do it nice and show the paths.

5.0.216 I'm drawing into a listbox but don't see something.

Plugin Version: all.

Answer: If you draw this in a listbox cellbackground, you need to draw on the correct position

Example:

```
Function CellBackgroundPaint(g As Graphics, row as Integer, column as Integer) As Boolean
dim f as FolderItem
f=SpecialFolder.Desktop
f.DrawWideIconMBS(g,listbox1.left,listbox1.top+row*20,16)
Return true
End Function
```

Notes: Try this in a listbox. The Graphics object there has a clipping and an offset which the plugin doesn't know about.

5.0.217 I'm searching for a method or so to move a window from position x.y to somewhere else on the screen.

Platform: macOS.

Answer:

The code I produced in RB isn't smooth enough. Is there a call in MBS, if not, can it be done? The speed of it has to be like the show of a DrawerWindow.

Try the declare below for Carbon. With WindowLib it will work on Mac OS 8.5 and newer.

Notes:

See Window.Transition functions.

5.0.218 If I use one of your plug-ins under windows, would this then impose the use of dll after compilation or my would my compiled soft still be a stand-alone single file software?

Platforms: macOS, Linux, Windows.

Answer: Stand alone.

Notes: Xojo compiles all used plugins into the application binary.

Some plugin parts need external dlls but you will find that in the documentation. (e.g. pdfiib for some classes)

5.0.219 Is the fn key on a powerbook keyboard down?

Plugin Version: all, Platform: macOS.

Answer: I am unable to figure out how or if it is possible to detect if the fn key is down on a powerbook keyboard. Is it possible?

Example:

' Window.Open Event of a blank project:

```
dim i as Integer

for i=0 to 127
if keyboard.asynckeydown(i) then
title=str(i) // found
return
end if
next
title="" // not found
```

Notes: This test application shows the keycode (decimal) 63 for the fn key.

5.0.220 Is there a case sensitive Dictionary?

Plugin Version: all.

Answer: The MBS Plugin has several classes which can work as a replacement.

Notes: First you could use VariantToVariantHashMapMBS or VariantToVariantOrderedMapMBS.

If you know that all keys are Strings or Integers only, you can use the specialized classes which are a little bit faster due to avoiding variants:

```
IntegerToIntegerHashMapMBS class
IntegerToIntegerOrderedMapMBS class
IntegerToStringHashMapMBS class
IntegerToStringOrderedMapMBS class
IntegerToVariantHashMapMBS class
IntegerToVariantOrderedMapMBS class
StringToStringHashMapMBS class
StringToStringOrderedMapMBS class
StringToVariantHashMapMBS class
StringToVariantOrderedMapMBS class
```

5.0.221 Is there a way to use the MBS plugin to get only the visible item and folder count on a volume?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use the DirectorySizeMBS class for this as in the example below:

Example:

```
dim d as DirectorySizeMBS

d=new DirectorySizeMBS

// volume(1) as my boot volume is very full
if d.update(volume(1),true,0) then
MsgBox str(d.VisibleItemCount)+" visible items, "+str(d.HiddenItemCount)+" invisible items."
end if
```

Notes: Complete Question: Is there a way to use the MBS plugin to get only the visible item and folder count on a volume? The FileCount and FolderCount properties of VolumeInformationMBS seem to provide the total # of items including invisible items such as .DS_Store and more importantly .Trashes which is causing me a great amount of difficulty during a recursive scan of a volume. I've got a progress bar which uses the total of the filecount and foldercount properties as the maximum value, but my routine needs to filter out all invisible items, as it is creating a catalog of a volume for archiving purposes. Any thoughts how I could get accurate number.

5.0.222 Is there an easy way I can launch the Displays preferences panel?

Plugin Version: all, Platform: macOS.

Answer: Use the code below:

Example:

```
dim error as Integer

error=OpenMacOSXPreferencesPaneMBS("Displays")
if error<>0 then
MsgBox "Failed to launch QuickTime System Preferences panel."
end if
```

5.0.223 List of Windows Error codes?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We have a list of windows error codes on our website.

Notes: <http://www.monkeybreadsoftware.de/xojo/winerror.shtml>

5.0.224 Midi latency on Windows problem?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The issue is system related, not a problem with RB or the plugin.

Notes: Two things will adversely affect the timing:

(1) latency of the software synthesizer output driver. The default Windows wavetable synthesizer has considerable latency. I don't know how many milliseconds, but it is noticeable.

(2) latency of the digital audio output driver. Different systems have different drivers for different audio hardware. My Dell laptop has a minimum 15ms latency in the audio driver.

These two things put together were causing a very sluggish MIDI response. I was able to verify these as the culprits by routing MIDI directly out of RB into a sample player, which only introduces the latency of (2) and does not include latency of (1).

I don't know how widely known are these facts, if not then you may want to add this information to the documentation, since Windows programmers using the MIDI plugin may not know those problems, and might mistakenly blame your plugin, as I did :) Sorry about that!

(From Aaron Andrew Hunt)

5.0.225 My Xojo Web App does not launch. Why?

Plugin Version: all, Platform: macOS.

Answer: Here is a list of checks to do for linux apache installations with Xojo or Xojo Web applications:

Notes: Just a list of checks to do for linux apache installations:

- You have 64bit linux? Than you need 32 bit compatibility libraries.
- The folder of your app is writable? Set permissions to 777.
- The cgi script is executable? Set permissions to 755.

- The app file itself is executable? Set permissions to 755.
- You uploaded cgi file as text, so it has unix line endings? (this often gives error "Premature end of script headers" in apache log)
- You uploaded config.cfg file and made it writable? Set permissions to 666.
- Your apache allows execution of cgi scripts? You enabled cgi for apache and uncommented addhandler command for CGI on a new apache installation?
- You uploaded the app file and libraries as binary files? Upload as text breaks them.
- You did upload the libs folder?
- You don't have code in app.open, session.open and other events which crashes app right at launch?
- You don't have a print command in your app.open event? (see feedback case 23817)
- You allowed htaccess file to overwrite permissions?

5.0.226 SQLiteDatabase not initialized error?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Before you can use SQLiteDatabaseMBS, it must be initialized.

Example:

```
dim d as new SQLiteDatabaseMBS
```

Notes: This happens normally when you use "new SQLiteDatabaseMBS".

But if you just have a SQLConnectionMBS and get a recordset there, the initialization may not have happened, yet.

So please simply add a line "dim d as new SQLiteDatabaseMBS" to your app.open code after registration, so the plugin part can initialize and late provide recordsets.

5.0.227 Textconverter returns only the first x characters. Why?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

Some older Xojo versions limit the Textconverter to around 1024 characters in input and output. This should be fixed with RB5.

Notes:

Xojo seems not to support Textconverters at all on Windows.

5.0.228 The type translation between CoreFoundation/Foundation and Xojo data types.

Plugin Version: all, Platform: macOS.

Answer: The plugin does conversion between Cocoa/Carbon data types and native Xojo data types. The following list help you knowing what the current plugins support:

Notes: Cocoa NSObject to Variant:

```

nil ->nil
NSDictionary ->Dictionary
NSData ->MemoryBlock
NSString ->String
NSAttributedString ->NSAttributedStringMBS
NSDate ->Date
NSNumber ->double/integer/Int64/UInt64/UInt32/Boolean
NSURL ->String
NSValue with NSRect ->NSRectMBS
NSValue with NSPoint ->NSPointMBS
NSValue with NSSize ->NSSizeMBS
NSValue with NSRange ->NSRangeMBS
NSValue with QTTime ->QTTimeMBS
NSValue with QTTimeRange ->QTTimeRangeMBS
NSArray ->Array of Variant
QuartzFilter ->QuartzFilterMBS

```

- ->*MBS

Variant to Cocoa NSObject:

```

nil ->nil
Dictionary ->NSDictionary
Boolean ->NSNumber
Integer ->NSNumber
Color ->NSColor
Int64 ->NSNumber
Single ->NSNumber
Double ->NSNumber
Date ->NSDate
MemoryBlock ->NSData
String ->NSString
NSImageMBS ->NSImage
NSAttributedStringMBS ->NSAttributedString
NSColorMBS ->NSColor
NSRectMBS ->NSValue with NSRect
NSSizeMBS ->NSValue with NSSize

```

NSPointMBS ->NSValue with NSPoint
 NSRangeMBS ->NSValue with NSRange
 NSBurnMBS ->NSBurn
 NSViewMBS ->NSView
 NSFontMBS ->NSFont
 NSParagraphStyleMBS ->NSParagraphStyle
 NSAttributedStringMBS ->NSAttributedString
 WebPolicyDelegateMBS ->WebPolicyDelegate
 WebUIDelegateMBS ->WebUIDelegate
 WebFrameLoadDelegateMBS ->WebFrameLoadDelegate
 WebResourceLoadDelegateMBS ->WebResourceLoadDelegate
 NSIndexSetMBS ->NSIndexSet
 QTTimeMBS ->QTTime
 QTTimeRangeMBS ->QTTimeRange
 Array of Variant ->NSArray
 Array of String ->NSArray
 CFStringMBS ->NSString
 CFNumberMBS ->NSNumber
 CFDataMBS ->NSData
 CFURLMBS ->NSURL
 CFArrayMBS ->NSArray
 CFDictionaryMBS ->NSDictionary
 CFBinaryDataMBS ->NSData

Carbon CTypeRef to Variant:

CFDictionaryRef ->Dictionary
 CFStringRef ->String
 CFDataRef ->String
 CFURL ->String
 CFNumber ->Integer/Double/Int64
 CFArray ->Array
 CFDate ->date
 nil ->nil
 CGColorSpace ->CGColorSpaceMBS
 CGColor ->CGColorMBS
 CGImage ->CGImageMBS
 CF* ->CF*MBS

Variant to Carbon CTypeRef:

Dictionary ->CFDictionaryRef
 Boolean ->CFBooleanRef
 Color ->CFNumberRef
 Integer ->CFNumberRef

Int64 ->CFNumberRef
 Single ->CFNumberRef
 Double ->CFNumberRef
 String ->CFStringRef
 Color ->CGColorRef
 Date ->CFDateRef
 nil ->nil
 Memoryblock ->CFDataRef
 FolderItem ->CFURLRef
 Dictionary ->CFDictionaryRef
 Array of Variant/String/Date/Double/Single/Int64/Integer ->CFArray
 CGRectMBS ->CGRect as CFDataRef
 CGSizeMBS ->CGSize as CFDataRef
 CGPointMBS ->CGPoint as CFDataRef
 CGColorMBS ->CGColor
 CGColorSpaceMBS ->CGColorSpace
 CGImageMBS ->CGImage
 CGDataConsumerMBS ->CGDataConsumer
 CGDataProviderMBS ->CGDataProvider
 CF*MBS ->CF*

Strings without encodings should be put into dictionaries as memoryblocks.

5.0.229 Uploaded my web app with FTP, but it does not run on the server!

Plugin Version: all, Platform: Windows.

Answer: If you see errors like a simple "Segmentation Fault" on Linux or some other wired errors, you may want to check your FTP upload mode. It must be binary for web apps. ASCII mode corrupts the application.

5.0.230 What classes to use for hotkeys?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use CarbonHotKeyMBS class on Mac and WindowsKeyFilterMBS on Windows.

Notes: CarbonHotKeyMBS will also work fine in Cocoa apps.

5.0.231 What do I need for Linux to get picture functions working?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: In order to get our plugins working on Linux systems without GUI, the plugin loads graphics

libraries dynamically.

Notes: To get it working, the plugin tries to load gtk with this paths:

- libgtk-x11-2.0.so”
- libgtk-x11-2.0.so.0”
- /usr/lib/libgtk-x11-2.0.so”
- /usr/lib32/libgtk-x11-2.0.so”
- /usr/lib/libgtk-x11-2.0.so.0”
- /usr/lib32/libgtk-x11-2.0.so.0”

gdk is loaded with this paths:

- libgdk-x11-2.0.so”
- libgdk-x11-2.0.so.0”
- /usr/lib/libgdk-x11-2.0.so”
- /usr/lib32/libgdk-x11-2.0.so”
- /usr/lib/libgdk-x11-2.0.so.0”
- /usr/lib32/libgdk-x11-2.0.so.0”

For the paths without explicit path, the system will search in /lib, /usr/lib and all directories in the LD_LIBRARY_PATH environment variable.

5.0.232 What does the NAN code mean?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

5.0.233 What font is used as a 'small font' in typical Mac OS X apps?

Plugin Version: all, Platform: macOS.

Answer:

Xojo 4.5 has a constant "SmallSystem" to use for a font name.

For older versions try this code:

Example:

```

Sub GetThemeFont(fontType as Integer, ByRef fontName as String, ByRef fontSize as Integer, ByRef
fontName as Integer)
dim err as Integer
dim theFont, theFontSize, theFontStyle as MemoryBlock

const smSystemScript = -1

Declare Function GetThemeFont Lib "Carbon" (inFontID as Integer, inScript as Integer, outFontName
as Ptr, outFontSize as Ptr, outStyle as Ptr) as Integer

theFont = NewMemoryBlock(256) //Str255
theFontSize = NewMemoryBlock(2) //SInt16
theFontStyle = NewMemoryBlock(1) //Style

err = GetThemeFont(fontType, smSystemScript, theFont, theFontSize, theFontStyle)

if err = 0 then
fontName = theFont.PString(0)
fontSize = theFontSize.UShort(0)
fontStyle = theFontStyle.Byte(0)
else
fontName = ""
fontSize = 0
fontStyle = 0
end if
End Sub

```

5.0.234 What is last plugin version to run on Mac OS X 10.4?

Plugin Version: all, Platform: Windows.

Answer: Last Version with 10.4 support is version 15.4.

Notes: With version 15.4 you can build applications for OS X 10.4 and newer.

For Version 16.0 we disabled 10.4 and moved minimum to 10.5. We may be able to enable it again to build a version of 16.x, but may need to charge for this by hour.

5.0.235 What is last plugin version to run on PPC?

Plugin Version: all, Platform: Windows.

Answer: Last Version with PPC is 15.4.

Notes: With version 15.4 you can build PPC applications for OS X 10.4 and newer.

For Version 16.0 we disabled PPC. We may be able to enable it again to build a PPC version of 16.x, but may need to charge for this by hour.

5.0.236 What is last version of the plugins for macOS 32-bit?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use version 23.0 or older.

Notes: We stopped including 32-bit code for macOS in version 23.1. Please use older versions if you use an old Xojo.

Xojo 2017r3 and newer load our 64-bit plugins.

5.0.237 What is the difference between Timer and WebTimer?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Timer is server side and WebTimer client side.

Notes: Timer is the normal timer class in Xojo. It runs on the server. On the client side the WebTimer runs on the client. It triggers a request to the server to perform the action. So a WebTimer is good to keep the connection running and the website updated regularly. A timer on the server is good to make regular jobs like starting a database backup every 24 hours.

5.0.238 What is the list of Excel functions?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Below a list of function names known by LibXL.

Notes: LibXL parses the functions and writes tokens to the excel file. So even if Excel can do more functions, we can only accept the ones known by LibXL.

ABS, ABSREF, ACOS, ACOSH, ACTIVE.CELL, ADD.BAR, ADD.COMMAND, ADD.MENU, ADD.TOOLBAR, ADDRESS, AND, APP.TITLE, AREAS, ARGUMENT, ASC, ASIN, ASINH, ATAN, ATAN2, ATANH, AVEDEV, AVERAGE, AVERAGEA, BAHTTEXT, BETADIST, BETAINV, BINOMDIST, BREAK, CALL, CALLER, CANCEL.KEY, CEILING, CELL, CHAR, CHECK.COMMAND, CHIDIST, CHIINV, CHITEST, CHOOSE, CLEAN, CODE, COLUMN, COLUMNS, COMBIN, CONCATENATE, CONFIDENCE, CORREL, COS, COSH, COUNT, COUNTA, COUNTBLANK, COUNTIF, COVAR, CREATE.OBJECT, CRITBINOM, CUSTOM.REPEAT, CUSTOM.UNDO, DATE, DATEDIF, DATESTRING, DATEVALUE, DAVERAGE, DAY, DAYS360, DB, DBCS, DCOUNT, DCOUNTA, DDB, DEGREES, DELETE.BAR, DELETE.COMMAND, DELETE.MENU, DELETE.TOOLBAR, Deref, DEVSQ, DGET, DIALOG.BOX, DIRECTORY, DMAX, DMIN, DOCUMENTS, DOLLAR, DPRODUCT, DSTDEV, DSTDEVP, DSUM, DVAR, DVARP, ECHO, ELSE, ELSE.IF, ENABLE.COMMAND, ENABLE.TOOL, END.IF, ERROR, ERROR.TYPE, EVALUATE, EVEN, EXACT, EXEC, EXECUTE, EXP, EXPONDIST, FACT, FALSE, FCLOSE, FDIST, FILES, FIND, FINDB, FINV, FISHER, FISHERINV, FIXED, FLOOR, FOPEN, FOR, FOR.CELL, FORECAST,

FORMULA.CONVERT, FPOS, FREAD, FREADLN, FREQUENCY, FSIZE, FTEST, FV, FWRITE, FWRITELN, GAMMADIST, GAMMAINV, GAMMALN, GEOMEAN, GET.BAR, GET.CELL, GET.CHART.ITEM, GET.DEF, GET.DOCUMENT, GET.FORMULA, GET.LINK.INFO, GET.MOVIE, GET.NAME, GET.NOTE, GET.OBJECT, GET.PIVOT.FIELD, GET.PIVOT.ITEM, GET.PIVOT.TABLE, GET.TOOL, GET.TOOLBAR, GET.WINDOW, GET.WORKBOOK, GET.WORKSPACE, GETPIVOTDATA, GOTO, GROUP, GROWTH, HALT, HARMEAN, HELP, HLOOKUP, HOUR, HYPERLINK, HYPGEOMDIST, IF, INDEX, INDIRECT, INFO, INITIATE, INPUT, INT, INTERCEPT, IPMT, IRR, ISBLANK, ISERR, ISERROR, ISLOGICAL, ISNA, ISNONTEXT, ISNUMBER, ISPMT, ISREF, ISTEXT, ISTHAIDIGIT, KURT, LARGE, LAST.ERROR, LEFT, LEFTB, LEN, LENB, LINEST, LINKS, LN, LOG, LOG10, LOGEST, LOGINV, LOGNORMDIST, LOOKUP, LOWER, MATCH, MAX, MAXA, MDETERM, MEDIAN, MID, MIDB, MIN, MINA, MINUTE, MINVERSE, MIRR, MMULT, MOD, MODE, MONTH, MOVIE.COMMAND, N, NA, NAMES, NEGBINOMDIST, NEXT, NORMDIST, NORMINV, NORMSDIST, NORMSINV, NOT, NOTE, NOW, NPER, NPV, NUMBERSTRING, ODD, OFFSET, OPEN.DIALOG, OPTIONS.LISTS.GET, OR, PAUSE, PEARSON, PERCENTILE, PERCENTRANK, PERMUT, PHONETIC, PI, PIVOT.ADD.DATA, PMT, POISSON, POKE, POWER, PPMT, PRESS.TOOL, PROB, PRODUCT, PROPER, PV, QUARTILE, RADIANS, RAND, RANK, RATE, REFTTEXT, REGISTER, REGISTER.ID, RELREF, RENAME.COMMAND, REPLACE, REPLACEB, REPT, REQUEST, RESET.TOOLBAR, RESTART, RESULT, RESUME, RETURN, RIGHT, RIGHTB, ROMAN, ROUND, ROUNDBAHTDOWN, ROUNDBAHTUP, ROUNDDOWN, ROUNDUP, ROW, ROWS, RSQ, RTD, SAVE.DIALOG, SAVE.TOOLBAR, SCENARIO.GET, SEARCH, SEARCHB, SECOND, SELECTION, SERIES, SET.NAME, SET.VALUE, SHOW.BAR, SIGN, SIN, SINH, SKEW, SLN, SLOPE, SMALL, SPELLING.CHECK, SQRT, STANDARDIZE, STDEV, STDEVA, STDEVP, STDEVPA, STEP, STEYX, SUBSTITUTE, SUBTOTAL, SUM, SUMIF, SUMPRODUCT, SUMSQ, SUMX2MY2, SUMX2PY2, SUMXMY2, SYD, T, TAN, TANH, TDIST, TERMINATE, TEXT, TEXT.BOX, TEXTREF, THAIDAYOFWEEK, THAIDIGIT, THAIMONTHOFYEAR, THAINUMSOUND, THAINUMSTRING, THAISTRINGLENGTH, THAIYEAR, TIME, TIMEVALUE, TINV, TODAY, TRANSPOSE, TREND, TRIM, TRIMMEAN, TRUE, TRUNC, TTEST, TYPE, UNREGISTER, UPPER, USDOLLAR, USERDEFINED, VALUE, VAR, VARA, VARP, VARPA, VDB, VIEW.GET, VLOOKUP, VOLATILE, WEEKDAY, WEIBULL, WHILE, WINDOW.TITLE, WINDOWS, YEAR and ZTEST.

5.0.239 What is the replacement for PluginMBS?

Plugin Version: all, Platform: macOS.

Answer: Use the SoftDeclareMBS class to load libraries dynamically.

5.0.240 What to do on Xojo reporting a conflict?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

I get an error like "This item conflicts with another item of the same name" when using one of the plugin functions.

Xojo just wants to tell you that you dropped something in the plugins folder what is not a plugin.

Notes:

Some users dropped the examples, the documentation or other files into the plugins folder. Don't do it.

5.0.241 What to do with a NSImageCacheException?

Plugin Version: all, Platforms: macOS, Windows.

Answer: You need to add exception handlers for NSExcptionMBS in order to catch this exception.

Notes: You may also add code to write the stack of the exception into a log file for later locating the error source.

A NSImage has several image representations in memory. So basicly you pass in the base image and for whatever size an image is needed, the NSImage class will create a cache image representation of the requested size so on the next query it can use that cache for the same requested size.

5.0.242 What to do with MySQL Error 2014?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can get this error on MySQL if you have a recordset open while you create another one.

5.0.243 What to do with SQL Plugin reporting Malformed string as error?

Plugin Version: all, Platform: macOS.

Answer: Please make sure the table and/or database fields have a text encoding set.

Notes: For Firebird our plugin tries to use UTF-8 encoding if possible and to correctly convert between various tables, the tables and their fields need to have a text encoding defined.

e.g. if the text field in the table is windows-1252 and the other ISO 8859-5, then the Firebird database can convert them to UTF-8 and deliver texts to the plugin.

If encoding is set to none, it may get confused for non-ascii text.

5.0.244 Where is CGGetActiveDisplayListMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetActiveDisplayList.

5.0.245 Where is CGGetDisplaysWithPointMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetDisplaysWithPoint.

5.0.246 Where is CGGetDisplaysWithRectMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetDisplaysWithRect.

5.0.247 Where is CGGetOnlineDisplayListMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetOnlineDisplayList.

5.0.248 Where is GetObjectClassNameMBS?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use this replacement method:

Example:

```
Function GetObjectClassNameMBS(o as Object) As string
dim t as Introspection.TypeInfo = Introspection.GetType(o)
Return t.FullName
End Function
```

Notes: GetObjectClassNameMBS was removed from the plugins.

5.0.249 Where is NetworkAvailableMBS?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We removed NetworkAvailableMBS some versions ago. It was not working right and basically it's not useful. If you want to check whether you have a network, than do a DNS resolve:

Example:

```

// two independent domain names
const domain1 = "www.google.com"
const domain2 = "www.macsw.de"

// resolve IPs
dim ip1 as string = DNSNameToAddressMBS(Domain1)
dim ip2 as string = DNSNameToAddressMBS(Domain2)

// if we got IPs and not the same IPs (error/login pages)
if len(ip1)=0 or len(ip2)=0 or ip1=ip2 then
MsgBox "no connection"
else
MsgBox "have connection"
end if

```

Notes: This way you can detect whether you got something from DNS. And you can make sure that a DNS redirection to a login page won't catch you.

5.0.250 Where is StringHeight function in DynaPDF?

Plugin Version: all, Platform: Windows.

Answer: Use the function GetFTextHeight or GetFTextHeightEx.

Notes: Be aware that GetFTextHeight works with format commands and you may want to escape your text if you don't use them.

5.0.251 Where is XLSDocumentMBS class?

Plugin Version: all, Platform: macOS.

Answer: This class has been removed in favor of XLBookMBS class.

Notes: This classes have been removed XLSCellMBS, XLSDocumentMBS, XLSFormatRecordMBS, XLSMergedCellsMBS, XLSRowMBS and XLSSheetMBS.

5.0.252 Where to get information about file formats?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

Please visit this web page:

<http://www.wotsit.org>

5.0.253 Where to register creator code for my application?

Plugin Version: all, Platform: macOS.

Answer:

Register at Apple:

<http://developer.apple.com/dev/cftype/information.html>

5.0.254 Which Mac OS X frameworks are 64bit only?

Plugin Version: all, Platform: macOS.

Answer: Some frameworks from Mac OS X do not support 32 bit applications, so we can't provide plugins for Xojo until 64bit target is available.

Notes: For Mac OS X 10.8:

- Accounts
- EventKit
- GLKit
- Social

and in 10.9:

- Accounts
- AVKit
- EventKit
- GameController
- GLKit
- MapKit
- MediaLibrary
- Social
- SpriteKit

In general Apple makes all new frameworks being 64 bit only.

5.0.255 Which plugins are 64bit only?

Plugin Version: all, Platform: macOS.

Answer: Some of our plugins work only in 64 bit modes as operation systems do not provide 32 bit code.

Notes: This effects currently: EventKit, Accounts, Social frameworks from Apple and our matching plugins.

5.0.256 Why application doesn't launch because of a missing ddraw.dll!?

Plugin Version: all, Platform: Windows.

Answer: Some RB versions require that you install DirectX from Microsoft on your Windows.

5.0.257 Why application doesn't launch because of a missing shlwapi.dll!?

Plugin Version: all, Platform: Windows.

Answer: Some RB versions require that you install the Internet Explorer from Microsoft on your Windows.

Notes: This bug is for several older Windows 95 editions.

5.0.258 Why do I hear a beep on keydown?

Plugin Version: all, Platform: Windows.

Answer: When the user presses a key, RB goes through all keydown event handlers till on returns true.

Notes: If no keydown event handler returns true for the key, a beep is performed.

5.0.259 Why does folderitem.item return nil?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Because Xojo fails to make a folderitem for you. Reason may be an alias file which can't be resolved or simply that you don't have enough access rights to read the folder content.

Notes: A more rarely reason is that the directory changed and the file with the given index or name does no longer exist.

5.0.260 Why doesn't showurl work?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

There are three main reasons:

1. showurl is not supported by Xojo in 68k applications.
2. there is now application defined for the protocol (e.g. http) in the Internet Control panel.
3. You don't have Internet Config installed.

You can use the InternetConfigMBS class to check for this stuff.

5.0.261 Why don't the picture functions not work on Linux?

Plugin Version: all, Platform: macOS.

Answer: Please make sure libcairo is installed.

Notes: For accessing pictures on Linux, the MBS Plugin relays on the cairo library.

Please install the package if you don't have it already.

Our plugin looks for library called libcairo.so or libcairo.so.2.

5.0.262 Why have I no values in my chart?

Plugin Version: all, Platforms: macOS, Windows.

Answer: You have no data points visible, there may be several reasons:

Notes: For example one of the data values may be infinite or invalid.

Or the scaling may be out of range, so you simply see nothing.

5.0.263 Will application size increase with using plugins?

Plugin Version: all, Platform: Windows.

Answer: All plugins used by your application will be included in the application.

Notes: If you use no plugins, your application will not change size.

And if you use one class from the plugins, your application size will increase by a few kilobytes.

The documentation of the plugins include a list of all plugin parts and their sizes for the different platforms.

5.0.264 XLS: Custom format string guidelines

Plugin Version: all, Platform: macOS.

Answer: You have to download the source code and compile a static version of the library.

Notes: Up to four sections of format codes can be specified. The format codes, separated by semicolons, define the formats for positive numbers, negative numbers, zero values, and text, in that order. If only two sections are specified, the first is used for positive numbers and zeros, and the second is used for negative numbers. If only one section is specified, it is used for all numbers. Four sections example:

```
#,###.00_); [ Red ] (,###.00);0.00;"sales "@
```

The following table describes the different symbols that are available for use in custom number formats.

Specify colors

To set the text color for a section of the format, type the name of one of the following eight colors in square brackets in the section. The color code must be the first item in the section.

Instead of using the name of the color, the color index can be used, like this [Color3] for Red. Valid numeric indexes for color range from 1 to 56, which reference by index to the legacy color palette.

Specify conditions

To set number formats that will be applied only if a number meets a specified condition, enclose the condition in square brackets. The condition consists of a comparison operator and a value. Comparison operators include: = Equal to; >Greater than; <Less than; >= Greater than or equal to, <= Less than or equal to, and <>Not equal to. For example, the following format displays numbers that are less than or equal to 100 in a red font and numbers that are greater than 100 in a blue font.

```
[ Red ] [ <=100 ] ; [ Blue ] [ >100 ]
```

If the cell value does not meet any of the criteria, then pound signs ("##") are displayed across the width of the cell.

Dates and times

Examples

5.0.265 Xojo doesn't work with your plugins on Windows 98.

Plugin Version: all, Platform: Windows.

Answer: Please upgrade your Windows version.

5.0.266 Xojo or my RB application itself crashes on launch on Mac OS Classic. Why?

Plugin Version: all.

Answer:

You may check if the application has enough memory to be loaded.

RB should have on Mac OS Classic more than 20 MB of RAM.

I preferred to use 50 MB and for an application a 10 MB partition is a good way to start.

Parameter	Description
x	The x value of the data point. For an enumerated x-axis (see <code>Axis.setLabels</code> on what is an enumerated axis), the first data point is 0, and the nth data point is (n-1).
xLabel	The bottom x-axis label of the data point.
x2Label	The top x-axis label of the data point.
value	The value of the data point.
accValue	The sum of values of all data points that are in the same x position and same data group as the current data point, and with data set number less than or equal to the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
totalValue	The sum of values of all data points that are in the same x position and same data group as the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
percent	The percentage of the data point based on the total value of all data points that are in the same x position and same data group as the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
accPercent	The accumulated percentage of the data point based on the total value of all data points that are in the same x position and same data group as the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
gpercent	The percentage of the data point based on the total value of all data points in a layer.
dataSet	The data set number to which the data point belongs. The first data set is 0. The nth data set is (n-1).
dataSetName	The name of the data set to which the data point belongs.
dataItem	The data point number within the data set. The first data point is 0. The nth data point is (n-1).
dataGroup	The data group number to which the data point belongs. The first data group is 0. The nth data group is (n-1).
dataGroupName	The name of the data group to which the data point belongs.
layerId	The layer number to which the data point belongs. The first layer is 0. The nth layer is (n-1).
fieldN	The (N + 1)th extra field. For example, { field0 } means the first extra field. An extra field is an array of custom elements added using <code>Layer.addExtraField</code> , <code>Layer.addExtraField2</code> , <code>BaseChart.addExtraField</code> or <code>BaseChart.addExtraField2</code> .

diFieldN	Same as fieldN. See above.
dsFieldN	Similar to fieldN, except that dsFieldN means the extra field is indexed by data set number. The Pth data set corresponds to the Pth element of the extra field.
dsdiFieldN	Similar to fieldN, except that dsdiFieldN means the extra fields are indexed by both the data set number and data point number. The Pth data item of the Qth data set corresponds to the Pth element of the (N + Q)th extra field.

Parameter	Description
zx	The symbol scale in the x dimension. Applicable for layers with symbol scales set by <code>LineStyle.setSymbolScale</code> .
zy	The symbol scale in the y dimension. Applicable for layers with symbol scales set by <code>LineStyle.setSymbolScale</code> .
z	The symbol scale without distinguishing the dimension to use. Applicable for layers with symbol scales set by <code>LineStyle.setSymbolScale</code> .

Parameter	Description
slope	The slope of the trend line.
intercept	The y-intercept of the trend line.
corr	The correlation coefficient in linear regression analysis.
stderr	The standard error in linear regression analysis.

Parameter	Description
top	The value of the top edge of the box-whisker symbol.
bottom	The value of the bottom edge of the box-whisker symbol.
max	The value of the maximum mark of the box-whisker symbol.
min	The value of the minimum mark of the box-whisker symbol.
med	The value of the median mark of the box-whisker symbol.

Parameter	Description
high	The high value.
low	The low value.
open	The open value.
close	The close value.

Parameter	Description
dir	The direction of the vector.
len	The length of the vector.

Parameter	Description
radius	The radial value of the data point.
value	Same as { radius } . See above.
angle	The angular value of the data point.
x	Same as { angle } . See above.
label	The angular label of the data point.
xLabel	Same as { label } . See above.
name	The name of the layer to which the data point belongs.
dataSetName	Same as { name } . See above.
i	The data point number. The first data point is 0. The nth data point is (n-1).
dataItem	Same as { i } . See above.
z	The symbol scale. Applicable for layers with symbol scales set by Polar-Layer.setSymbolScale.
fieldN	The (N + 1)th extra field. For example, { field0 } means the first extra field. An extra field is an array of custom elements added using Layer.addExtraField, Layer.addExtraField2, BaseChart.addExtraField or BaseChart.addExtraField2.

diFieldN	Same as fieldN. See above.
dsFieldN	Similar to fieldN, except that dsFieldN means the extra field is indexed by layer index. The Pth layer corresponds to the Pth element of the extra field.
dsdiFieldN	Similar to fieldN, except that dsdiFieldN means the extra fields are indexed by both the data set number and data point number. The Pth data item of the Qth layer corresponds to the Pth element of the (N + Q)th extra field.

Parameter	Description
dir	The direction of the vector.
len	The length of the vector.

Parameter	Description
value	The axis value at the tick position.
label	The axis label at the tick position.

Parameter	Description
[param]	The name of the parameter
[a]	If this field a number, it specifies the number of decimal places (digits to the right of the decimal point).

[b]

textasciitilde ' for no thousand separator. The default is 'textasciitilde ', which can be modified using `BaseChart.setNumberFormat`.

[c]

The thousand separator. Should be a non-alphanumeric character (not 0-9, A-Z, a-z). Use '.

The decimal point character. The default is '.', which can be modified using `BaseChart.setNumberFormat`.

[d]

textasciitilde ' for no negative sign character. The default is '-', which can be modified using `BaseChart.setNumberFormat`.

The negative sign character. Use '-'

Parameter	Description
yyyy	The year in 4 digits (e.g. 2002)
yyy	The year showing only the least significant 3 digits (e.g. 002 for the year 2002)
yy	The year showing only the least significant 2 digits (e.g. 02 for the year 2002)
y	The year showing only the least significant 1 digits (e.g. 2 for the year 2002)
mmm	The month formatted as its name. The default is to use the first 3 characters of the english month name (Jan, Feb, Mar ...). The names can be configured using <code>BaseChart.setMonthNames</code> .
mm	The month formatted as 2 digits from 01 - 12, adding leading zero if necessary.
m	The month formatted using the minimum number of digits from 1 - 12.
MMM	The first 3 characters of the month name converted to upper case. The names can be configured using <code>BaseChart.setMonthNames</code> .
MM	The first 2 characters of the month name converted to upper case. The names can be configured using <code>BaseChart.setMonthNames</code> .
M	The first character of the month name converted to upper case. The names can be configured using <code>BaseChart.setMonthNames</code> .
dd	The day of month formatted as 2 digits from 01 - 31, adding leading zero if necessary.
d	The day of month formatted using the minimum number of digits from 1 - 31.
w	The name of the day of week. The default is to use the first 3 characters of the english day of week name (Sun, Mon, Tue ...). The names can be configured using <code>BaseChart.setWeekDayNames</code> .
hh	The hour of day formatted as 2 digits, adding leading zero if necessary. The 2 digits will be 00 - 23 if the 'a' option (see below) is not specified, otherwise it will be 01 - 12.
h	The hour of day formatted using the minimum number of digits. The digits will be 0 - 23 if the 'a' option (see below) is not specified, otherwise it will be 01 - 12.
nn	The minute formatted as 2 digits from 00 - 59, adding leading zero if necessary.
n	The minute formatted using the minimum number of digits from 00 - 59.
ss	The second formatted as 2 digits from 00 - 59, adding leading zero if necessary.
s	The second formatted using the minimum number of digits from 00 - 59.
a	Display either 'am' or 'pm', depending on whether the time is in the morning or afternoon. The text 'am' and 'pm' can be modified using <code>BaseChart.setAMPM</code> .

Shape Id	Value	Description
SquareShape	1	Square shape. See (1, 1) above.
DiamondShape	2	Diamond shape. See (2, 1) above.
TriangleShape	3	Triangle shape pointing upwards. See (3, 1) above.
RightTriangleShape	4	Triangle shape pointing rightwards. See (4, 1) above.
LeftTriangleShape	5	Triangle shape pointing leftwards. See (5, 1) above.
InvertedTriangleShape	6	Triangle shape pointing downwards. See (1, 2) above.
CircleShape	7	Circle shape. See (2, 2) above.
StarShape	[Method]	Star shapes of various points. See (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5) above for stars with 3 to 10 points.
PolygonShape	[Method]	Polygon shapes symmetrical about a vertical axis with a vertex at the top center position. See (4, 1), (4, 3), (4, 5), (5, 1) for polygons of 5 to 8 sides.
Polygon2Shape	[Method]	Polygon shapes symmetrical about a vertical axis but without any vertex at the top center position. See (4, 2), (4, 4) for polygons of 5 and 6 sides.
CrossShape	[Method]	'+' shapes. See (5, 2), (5, 3), (5, 4), (5, 5), (6, 1), (6, 2), (6, 3) for '+' shape with arm width of 0.1 - 0.7.
Cross2Shape	[Method]	'X' shapes. See (6, 4), (6, 5), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5) for 'X' shapes with arm width of 0.1 - 0.7.

langEnglish	0	Roman script
langFrench	1	Roman script
langGerman	2	Roman script
langItalian	3	Roman script
langDutch	4	Roman script
langSwedish	5	Roman script
langSpanish	6	Roman script
langDanish	7	Roman script
langPortuguese	8	Roman script
langNorwegian	9	Roman script
langHebrew	10	Hebrew script
langJapanese	11	Japanese script
langArabic	12	Arabic script
langFinnish	13	Roman script
langGreek	14	Greek script using smRoman script code
langIcelandic	15	modified smRoman/Icelandic script
langMaltese	16	Roman script
langTurkish	17	modified smRoman/Turkish script
langCroatian	18	modified smRoman/Croatian script
langTradChinese	19	Chinese (Mandarin) in traditional characters
langUrdu	20	Arabic script
langHindi	21	Devanagari script
langThai	22	Thai script
langKorean	23	Korean script

Nan	Meaning
1	Invalid square root (negative number, usually)
2	Invalid addition (indeterminate such as infinity + (-infinity))
4	Invalid division (indeterminate such as 0/0)
8	Invalid multiplication (indeterminate such as 0*infinity)
9	Invalid modulo such as (a mod 0)
17	Try to convert invalid string to a number like val("x7")
33	Invalid argument in a trig function
34	Invalid argument in an inverse trig function
36	Invalid argument in a log function
37	Invalid argument in Pow function
38	Invalid argument in toolbox financial function
40	Invalid argument in hyperbolic function
42	Invalid argument in a gamma function

Symbol	Description and result
0	Digit placeholder. For example, if the value 8.9 is to be displayed as 8.90, use the format #.00
#	Digit placeholder. This symbol follows the same rules as the 0 symbol. However, the application shall not display extra zeros when the number typed has fewer digits on either side of the decimal than there are # symbols in the format. For example, if the custom format is #.##, and 8.9 is in the cell, the number 8.9 is displayed.
?	Digit placeholder. This symbol follows the same rules as the 0 symbol. However, the application shall put a space for insignificant zeros on either side of the decimal point so that decimal points are aligned in the column. For example, the custom format 0.0? aligns the decimal points for the numbers 8.9 and 88.99 in a column.
. (period)	Decimal point.
%	Percentage. If the cell contains a number between 0 and 1, and the custom format 0% is used, the application shall multiply the number by 100 and add the percentage symbol in the cell.
, (comma)	Thousands separator. The application shall separate thousands by commas if the format contains a comma that is enclosed by number signs (#) or by zeros. A comma that follows a placeholder scales the number by one thousand. For example, if the format is #.0,, and the cell value is 12,200,000 then the number 12.2 is displayed.
E- E+ e- e+	Scientific format. The application shall display a number to the right of the "E" symbol that corresponds to the number of places that the decimal point was moved. For example, if the format is 0.00E+00, and the value 12,200,000 is in the cell, the number 1.22E+07 is displayed. If the number format is #0.0E+0, then the number 12.2E+6 is displayed.
\$ -+/():space	Displays the symbol. If it is desired to display a character that differs from one of these symbols, precede the character with a backslash (\). Alternatively, enclose the character in quotation marks. For example, if the number format is (000), and the value 12 is in the cell, the number (012) is displayed.
\	Display the next character in the format. The application shall not display the backslash. For example, if the number format is 0\!, and the value 3 is in the cell, the value 3! is displayed.
*	Repeat the next character in the format enough times to fill the column to its current width. There shall not be more than one asterisk in one section of the format. If more than one asterisk appears in one section of the format, all but the last asterisk shall be ignored. For example, if the number format is 0*x, and the value 3 is in the cell, the value 3xxxxxx is displayed. The number of x characters that are displayed in the cell varies based on the width of the column.
_ (underline)	Skip the width of the next character. This is useful for lining up negative and positive values in different cells of the same column. For example, the number format _(0.0_);(0.0) aligns the numbers 2.3 and -4.5 in the column even though the negative number is enclosed by parentheses.
"text"	Display whatever text is inside the quotation marks. For example, the format 0.00 "dollars" displays 1.23 dollars when the value 1.23 is in the cell.
@	Text placeholder. If text is typed in the cell, the text from the cell is placed in the format where the at symbol (@) appears. For example, if the number format is "Bob "@ Smith" (including quotation marks), and the value "John" is in the cell, the value Bob John Smith is displayed.

[Black] [Green] [White] [Blue] [Magenta] [Yellow] [Cyan] [Red]

To display	As	Use this code
Months	1-12	m
Months	01-12	mm
Months	Jan-Dec	mmm
Months	January-December	mmmm
Months	J-D	mmmmm
Days	1-31	d
Days	01-31	dd
Days	Sun-Sat	ddd
Days	Sunday-Saturday	dddd
Years	00-99	yy
Years	1900-9999	yyyy
Hours	0-23	h
Hours	00-23	hh
Minutes	0-59	m
Minutes	00-59	mm
Seconds	0-59	s
Seconds	00-59	ss
Time	4 AM	h AM/PM
Time	4:36 PM	h:mm AM/PM
Time	4:36:03 P	h:mm:ss A/P
Time	4:36:03.75	h:mm:ss.00
Elapsed time	1:02	[h] :mm
Elapsed time	62:16	[mm] :ss
Elapsed time	3735.80	[ss] .00

To display	As	Use this code
1234.59	1234.6	#####.#
8.9	8.900	#.000
.631	0.6	0.#
12	12.0	#.0#
1234.568	1234.57	#.0#
44.398	44.398	???.???
102.65	102.65	???.???
2.8	2.8	???.???
5.25	5 1/4	# ??/??
5.3	5 3/10	# ??/??
12000	12,000	#,###
12000	12	#,
12400000	12.4	0.0,,