What's new in xmCHART 5.0

At a glance

- New chart types: Contour plots, density plots, violin plots.
- New diagram types: Sankey diagrams, chord diagrams, Venn diagrams.
- New gauge types: Linear and radial gauges.
- New text engine rewritten from scratch – supports now styled texts and hyphenation.
- New table, group and tag elements.
- Legend for heat maps, tree maps and density plots.
- New and improved arrow and round polygon objects.
- Texts and symbols along arbitrary paths.
- Backgrounds with round corners and paddings.
- Enhanced and improved symbol collection.
- Custom color schemes and color scales.
- Improved and optimized dashed line styles.
- Enhanced fill style for Gantt charts.
- Rotated views.
- New and faster input parser.
- Support of Script Steps.
- Enhanced gallery database.
- Numerous bug fixes.
- Available for macOS, Windows, Linux Ubuntu.
- Running native on Apple Silicon M1, M2 & M3 processors.
- Ready for FileMaker Pro 2024, FileMaker Server 2024, FileMaker WebDirect.

Requirements:
- FileMaker Pro 16 or higher.
- macOS 10.11 (El Capitan) or higher, Windows 7 or higher.
- Windows Server 2008 R2 with Platform Update, Windows Server 2012 or higher.
- Linux Ubuntu 22.04 or higher.

Table Of Contents

New Chart Types

xmCHART 5 comes with seven new chart types:

- Circular Bar Charts
- Linear Meters
- Linear Gauges
- Radial Gauges
- Contour Plots
- Density Plots
- Venn Diagrams

Circular Bar Charts

The `CircularBarChart()` function makes it possible to draw bars arranged in concentric circles. The appearance of the bars can be controlled by the `FillStyle()`, `BorderStyle()` and `SymbolStyle()` functions; the scale by using the `Scaling()` function.

```
CircularBarChart(appearance;startAngle;arcAngle;innerRadiusMin;barWidth;barGap;
                 barBackgroundWidth;barBackgroundOffset;barBackgroundShading;
                 barBackgroundColor1;barBackgroundColor2;...;barBackgroundColor100)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
appearance	int	0..127	0	*Constants: shadow,label*
startAngle	num	-360..360	0	
arcAngle	num	-360..360	360	*If negative, go counter-clockwise.*
innerRadiusMin	num	0..1	0,25	*Relative to outer radius.*
barWidth	num	0..1	0,15	*Relative to outer radius.*
barGap	num	0..1	0	*Relative to outer radius.*
barBackgroundWidth	num	0..10	1	*Relative to bar width.*
barBackgroundOffset	num	0..1	0	*Relative to bar width.*
barBackgroundShading	int	-1..128	solid	
barBackgroundColor1	rgba	0..255	#e6e6e6	
barBackgroundColor2	rgba	0..255	#e6e6e6	
...	rgba	0..255	#e6e6e6	
barBackgroundColor100	rgba	0..255	#e6e6e6	

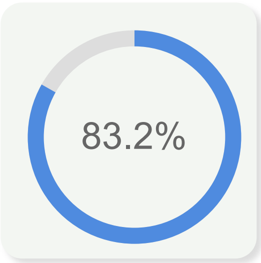
Location of the labels on circular bar charts:

Constant	*Value*	*Positioning*
smartBegin	1	Beginning of bar if enough space is available.
smartCenter	2	Middle of bar if enough space is available.
smartEnd	3	End of bar if enough space is available.
begin	4	Beginning of bar (forced).
center	5	Middle of bar (forced).
end	6	End of bar (forced).
edge	7	Edge on end of bar (forced).
smartOut	8	Outside if enough space is available (default).
out	9	Outside end of bar (forced).
– (no constant)	10	Outside beginning of bar (tangential).
– (no constant)	11	Outside beginning of bar (circular).

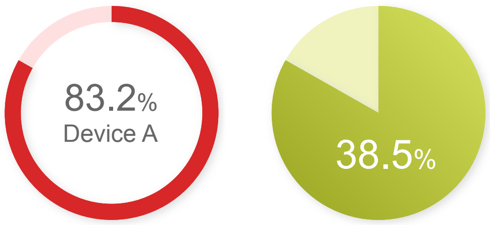
Please note, legend, title and label backgrounds are ignored in combination with circular bar charts.

Examples:

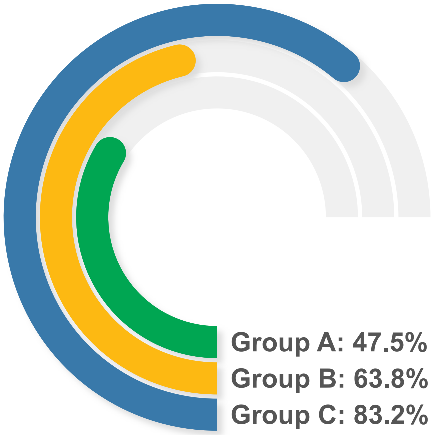
```
OpenDrawing(100;100)
  ChartData(83.2)
  CircularBarChart(default;
                   0;         /* Start angle in [deg].                                  */
                   360;       /* Arc angle in [deg]. If negative, go counter-clockwise. */
                   0.25;      /* Inner radius minimum. Relative to outer radius.        */
                   0.15;      /* Bar width. Relative to outer radius.                   */
                   0;         /* Bar gap. Relative to outer radius.                     */
                   1;         /* Bar background width. Relative to bar width.           */
                   0;         /* Bar background offset. Relative to bar width.          */
                   0;         /* Bar background shading: 0...solid, -1...shaded.        */
                   lightGray) /* Bar background color.                                  */
  /* Use style functions to control the appearance of bar(s). */
  FillStyle(1;79 139 223)
  BorderStyle(1;none)
  /* Use the Scaling() function to control the value range. */
  Scaling(all;linear;0;100)
  AddText(50;50;"83.2%";Arial;14;plain;#666;center;center)
  Background( #f3f6f2;0;0;;;2 2 2;lightGray)
  BackgroundOptions(8)
CloseDrawing()
```



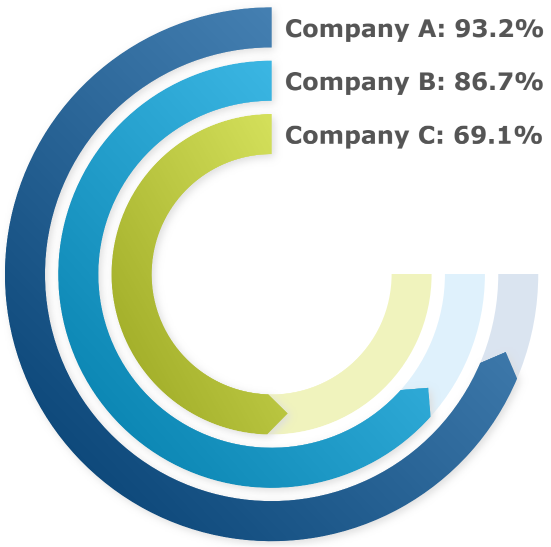
```
OpenDrawing(200;100)
  /* Left Graph. */
  OpenChart(0;0;100;100)
    ChartData(83.2)
    CircularBarChart(shadow;
                     0;       /* Start angle in [deg].                          */
                     360;     /* Arc angle in [deg]. If negative, go counter-clockwise. */
                     0.25;    /* Inner radius minimum. Relative to outer radius.      */
                     0.15;    /* Bar width. Relative to outer radius.                 */
                     0;       /* Bar gap. Relative to outer radius.                   */
                     1;       /* Bar background width. Relative to bar width.         */
                     0;       /* Bar background offset. Relative to bar width.        */
                     0;       /* Bar background shading: 0...solid, -1...shaded.      */
                     #ffe0e0) /* Bar background color.                                */
    /* Use style functions to control the appearance of bar(s). */
    FillStyle(1;#D72729)
    BorderStyle(all;none)
    ShadowStyle(all;1 1 3;lightGray)
    /* Use the Scaling() function to control the value range. */
    Scaling(all;linear;0;100)
    AddText(50;50;"<span size=14>83.2</span>%\nDevice A";Arial;9;plain;#666;center;center)
  CloseChart()
  /* Right Graph. */
  OpenChart(100;0;100;100)
    ChartData(83.2)
    CircularBarChart(shadow;
                     0;       /* Start angle in [deg].                          */
                     360;     /* Arc angle in [deg]. If negative, go counter-clockwise. */
                     0;       /* Inner radius minimum. Relative to outer radius.      */
                     1;       /* Bar width. Relative to outer radius.                 */
                     0;       /* Bar gap. Relative to outer radius.                   */
                     1;       /* Bar background width. Relative to bar width.         */
                     0;       /* Bar background offset. Relative to bar width.        */
                     0;       /* Bar background shading: 0...solid, -1...shaded.      */
                     #f0f3bd) /* Bar background color.                                */
    /* Use style functions to control the appearance of bar(s). */
    FillStyle(1;#cbdb29;shaded)
    BorderStyle(all;none)
    ShadowStyle(all;1 1 3;lightGray)
    /* Use the Scaling() function to control the value range. */
    Scaling(all;linear;0;100)
    AddText(153;66;"38.5<span size=10>%</span>";Arial;15;plain;#fff;center;center)
  CloseChart()
CloseDrawing()
```



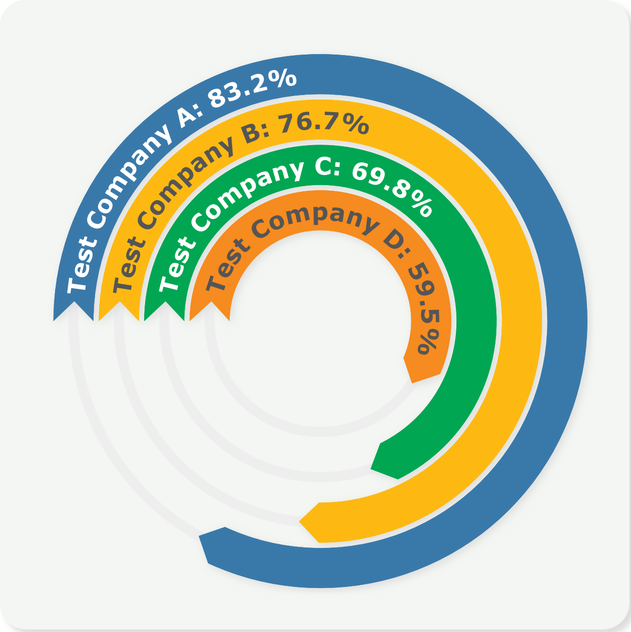
```
OpenDrawing(200;200)
  OpenChart(10;10;180;180)
    ChartData(83.2;63.8;47.5)
    CircularBarChart(shadow+label;
                     180;       /* Start angle in [deg].                                  */
                     270;       /* Arc angle in [deg]. If negative, go counter-clockwise. */
                     0.25;      /* Inner radius minimum. Relative to outer radius.        */
                     0.15;      /* Bar width. Relative to outer radius.                   */
                     0.02;      /* Bar gap. Relative to outer radius.                     */
                     1;         /* Bar background width. Relative to bar width.           */
                     0;         /* Bar background offset. Relative to bar width.          */
                     0;         /* Bar background shading: 0...solid, -1...shaded.        */
                     #f0f0f0 )  /* Bar background color.                                  */
    /* Use style functions to control the appearance of bar(s). */
    FillColorScheme(3)
    BorderStyle(all;none)
    BarStyle(all;0;1) /* Bar caps at start and end of bar [0..7] */
    ShadowStyle(all;2 2 2;lightGray)
    LabelTexts(1;"Group C: |f1|%")
    LabelTexts(2;"Group B: |f1|%")
    LabelTexts(3;"Group A: |f1|%")
    LabelStyle(all;Arial;10;bold;#555)
    LabelOptions(all; /* Bar number.                                        */
                 10;  /* Label location [0..11]                             */
                      /* 10...Labels at the begin of the bars - tangential. */
                      /* 11...Labels at the begin of the bars - circular.   */
                 3;   /* Offset - circular.                                 */
                 0)   /* Offset - radial.                                   */
    /* Use the Scaling() function to control the value range. */
    Scaling(all;linear;0;100)
  CloseChart()
CloseDrawing()
```



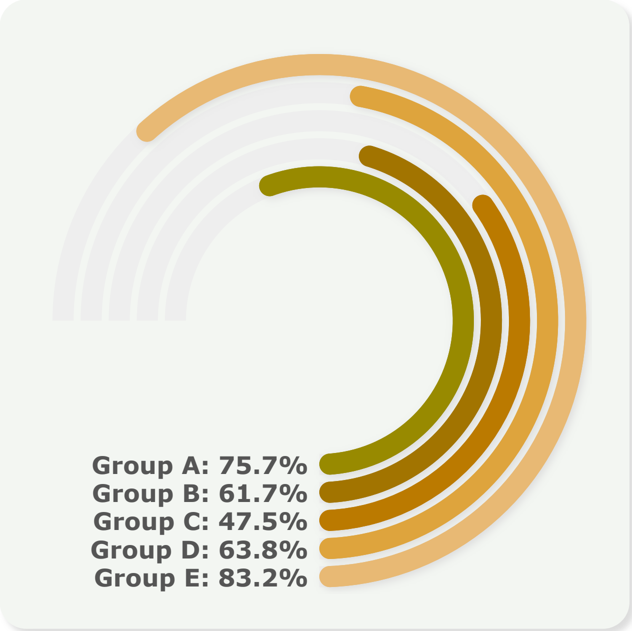
```
OpenDrawing(240;240)
  OpenChart(10;10;220;220)
    ChartData(0.932;0.867;0.691)
    CircularBarChart(shadow+label;
                     0;      /* Start angle in [deg].                                  */
                    -270;    /* Arc angle in [deg]. If negative, go counter-clockwise. */
                     0;      /* Inner radius minimum. Relative to outer radius.        */
                     0.15;   /* Bar width. Relative to outer radius.                   */
                     0.05;   /* Bar gap. Relative to outer radius.                     */
                     1;      /* Bar background width. Relative to bar width.           */
                     0;      /* Bar background offset. Relative to bar width.          */
                     0;      /* Bar background shading: 0...solid, -1...shaded.        */
                     #dae4f0;#dff1fc;#f0f3bd)   /* Bar background colors.              */
    /* Use style functions to control the appearance of bar(s). */
    FillStyle(1;#005ca9;shaded)
    FillStyle(2;#00afef;shaded)
    FillStyle(3;#cbdb29;shaded)
    BorderStyle(all;none)
    BarStyle(all;0;4) /* Bar caps at start and end of bar [0..7] */
    ShadowStyle(all;1 1 3;lightGray)
    LabelTexts(1;"Company A: |2f1|%")
    LabelTexts(2;"Company B: |2f1|%")
    LabelTexts(3;"Company C: |2f1|%")
    LabelStyle(all;Verdana;9;bold;#555)
    LabelOptions(all; /* Bar number.            */
                 10;  /* Label location [0..11]                             */
                      /* 10...Labels at the begin of the bars - tangential. */
                      /* 11...Labels at the begin of the bars - circular.   */
                 3;   /* Offset - circular.                                 */
                 0)   /* Offset - radial.                                   */
    /* Use the Scaling() function to control the value range. */
    Scaling(all;linear;0;1)
  CloseChart()
CloseDrawing()
```



```
OpenDrawing(240;240)
  OpenChart(10;10;220;220)
    ChartData(83.2;76.7;69.8;59.5)
    CircularBarChart(shadow+label;
                     -90;    /* Start angle in [deg].                                  */
                     360;    /* Arc angle in [deg]. If negative, go counter-clockwise. */
                     0.25;   /* Inner radius minimum. Relative to outer radius.        */
                     0.15;   /* Bar width. Relative to outer radius.                   */
                     0.02;   /* Bar gap. Relative to outer radius.                     */
                     0.25;   /* Bar background width. Relative to bar width.           */
                     0;      /* Bar background offset. Relative to bar width.          */
                     0;      /* Bar background shading: 0...solid, -1...shaded.        */
                     #eee)   /* Bar background color.                                  */
    /* Use style functions to control the appearance of bar(s). */
    FillColorScheme(3)
    BorderStyle(all;none)
    BarStyle(all;7;4) /* Bar caps at start and end of bar [0..7] */
    ShadowStyle(all;1 1 3;lightGray)
    LabelTexts(1;"Test Company A: |f1|%")
    LabelTexts(2;"Test Company B: |f1|%")
    LabelTexts(3;"Test Company C: |f1|%")
    LabelTexts(4;"Test Company D: |f1|%")
    LabelStyle(all;Verdana;9;bold;#555)
    LabelStyle(1;Verdana;9;bold;white)
    LabelStyle(3;Verdana;9;bold;white)
    LabelOptions(all;   /* Bar number.            */
                 begin; /* Label location [0..11] */
                 0;     /* Offset - circular.     */
                 0)     /* Offset - radial.       */
    /* Use the Scaling() function to control the value range. */
    Scaling(all;linear;0;100)
  CloseChart()
  Background(#f3f6f2;0;0;;;2 2 2;lightGray)
  BackgroundOptions(10)
CloseDrawing()
```

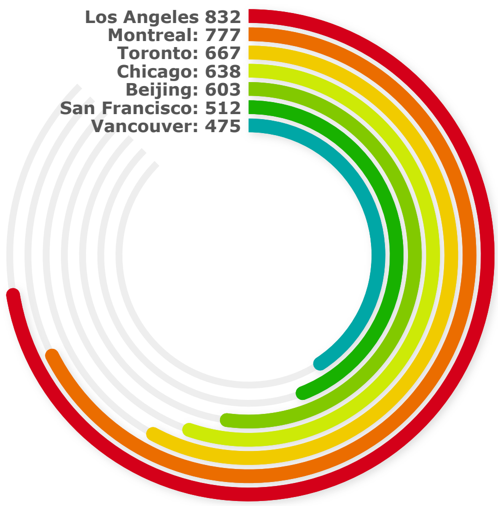


```
OpenDrawing(240;240)
  OpenChart(10;10;220;220)
    ChartData(0.832;0.638;0.475;0.617;0.757)
    CircularBarChart(shadow+label;
                     180;    /* Start angle in [deg].                                  */
                    -270;    /* Arc angle in [deg]. If negative, go counter-clockwise. */
                     0.5;    /* Inner radius minimum. Relative to outer radius.        */
                     0.15;   /* Bar width. Relative to outer radius.                   */
                     0.05;   /* Bar gap. Relative to outer radius.                     */
                     1;      /* Bar background width. Relative to bar width.           */
                     0;      /* Bar background offset. Relative to bar width.          */
                     0;      /* Bar background shading: 0...solid, -1...shaded.        */
                     #eee)   /* Bar background color.                                  */
    /* Use style functions to control the appearance of bar(s). */
    FillColorScheme(19)
    BorderStyle(all;none)
    BarStyle(all;1;1) /* Bar caps at start and end of bar [0..7] */
    ShadowStyle(all;1 1 3;lightGray)
    LabelTexts(5;"Group A: |2f1|%")
    LabelTexts(4;"Group B: |2f1|%")
    LabelTexts(3;"Group C: |2f1|%")
    LabelTexts(2;"Group D: |2f1|%")
    LabelTexts(1;"Group E: |2f1|%")
    LabelStyle(all;Verdana;9;bold;#555)
    LabelOptions(all;  /* Bar number.                                        */
                 10;   /* Label location [0..11]                             */
                       /* 10...Labels at the begin of the bars - tangential. */
                       /* 11...Labels at the begin of the bars - circular.   */
                 2;    /* Offset - circular.                                 */
                 0)    /* Offset - radial.                                   */
    /* Use the Scaling() function to control the value range. */
    Scaling(all;linear;0;1)
  CloseChart()
  Background(#f3f6f2;0;0;;;2 2 2;lightGray)
  BackgroundOptions(10)
CloseDrawing()
```



]

```
OpenDrawing(300;300)
  OpenChart(10;10;280;280)
    ChartData(832;777;667;638;603;512;475)
    CircularBarChart(shadow+label;
                     0;      /* Start angle in [deg].                                  */
                     315;    /* Arc angle in [deg]. If negative, go counter-clockwise. */
                     0.5;    /* Inner radius minimum. Relative to outer radius.        */
                     0.15;   /* Bar width. Relative to outer radius.                   */
                     0.05;   /* Bar gap. Relative to outer radius.                     */
                     0.5;    /* Bar background width. Relative to bar width.           */
                     0;      /* Bar background offset. Relative to bar width.          */
                     0;      /* Bar background shading: 0...solid, -1...shaded.        */
                     #eee)   /* Bar background color.                                  */
    /* Use style functions to control the appearance of bar(s). */
    FillColorScheme(15)
    BorderStyle(all;none)
    BarStyle(all;0;1) /* Bar caps at start and end of bar [0..7] */
    ShadowStyle(all;1 1 3;lightGray)
    LabelTexts(1;"Los Angeles |u|")
    LabelTexts(2;"Montreal: |u|")
    LabelTexts(3;"Toronto: |u|")
    LabelTexts(4;"Chicago: |u|")
    LabelTexts(5;"Beijing: |u|")
    LabelTexts(6;"San Francisco: |u|")
    LabelTexts(7;"Vancouver: |u|")
    LabelStyle(all;Verdana;9;bold;#555)
    LabelOptions(all; /* Bar number.                                        */
                 10;  /* Label location [0..11]                             */
                      /* 10...Labels at the begin of the bars - tangential. */
                      /* 11...Labels at the begin of the bars - circular.   */
                 2;   /* Offset - circular.                                 */
                 0)   /* Offset - radial.                                   */
    /* Use the Scaling() function to control the value range. */
    Scaling(all;linear;0;1000)
  CloseChart()
CloseDrawing()
```



Linear Meters

```
LinearMeter(appearance;thickness;cornerRoundings;scaleLineOffset;majorTickMarkOffset;
           minorTickMarkOffset;symbolOffset;barGap;barShape;meterFillColor;
           meterFillColorVariant;meterBorderStroke;meterBorderColor;
           meterBorderColorVariant)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
appearance	int	0..127	0	*Const: shadow,symbol,horizontal*
thickness	num	0..10000	20 %	*Abs or rel in percent of plot area*
cornerRoundings	num[]	0..1000	0	*Abs or rel in percent of plot area*
scaleLineOffset	num	-1000..1000	0	
majorTickMarkOffset	num	-1000..1000	0	
minorTickMarkOffset	num	-1000..1000	0	
symbolOffset	num[]	-1000..1000	0	
barGap	num	0..100	0,5	
barShape	int	0..16	0	
meterFillColor	rgba	0..255	none	
meterFillColorVariant	int	-1..128	solid	
meterBorderStroke	num[]	0..1000	0	*Dimension: [pt]*
meterBorderColor	rgba	0..255	black	
meterBorderColorVariant	int	-1..128	solid	

A bar is defined by entering two values in `ChartData()`; the first value defines the beginning of the bar, the second value the end. A symbol is defined by entering one value in `ChartData()`. Multiple bars and symbols are separated by a semicolon ";". For example:

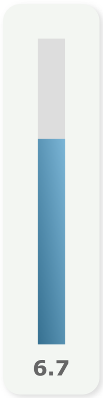
```
ChartData(-10 56; /* Bar1 from -10 to 56 */
            0 87; /* Bar2 from   0 to 87 */
           17.5)  /* Symbol at 17.5      */
```

The appearance of the bars and symbols can be controlled by the `FillStyle()`, `BorderStyle()` and `SymbolStyle()` functions. By using scale and axis functions the appearance of the scale can be controlled. Grid line and grid stripe functions make it possible to define the background behind the bar(s).

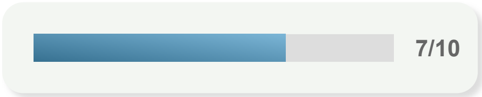
Legend and title are not supported in combination with linear meters.

Examples:

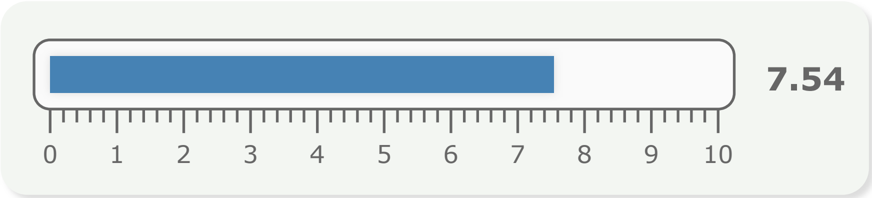
```
OpenDrawing(150;250)
  ChartData(0 6.7)
  LinearMeter(default;
              12%;       /* Thickness relative to plot area width. */
              0;         /* Corner roundings.                      */
              0;         /* Scale line offset.                     */
              0;         /* Major tick mark offset.                */
              0;         /* Minor tick mark offset.                */
              0;         /* Symbol offset.                         */
              0;         /* Bar gap relative to bar thickness.     */
              0;         /* Bar shape. [0..16]                     */
              lightGray) /* Meter fill color.                      */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#499BC9;shaded)
  BorderStyle(1;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;5;4)
  AxisOptions(all;0)
  AddText(75;225;"6.7";Verdana;12;bold;#666;center;;0;150)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;20 20 30 20)
CloseDrawing()
```



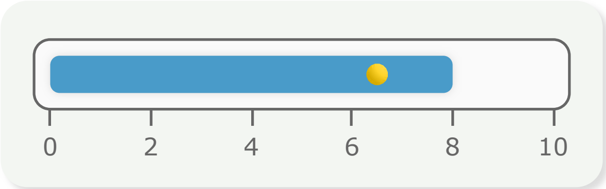
```
OpenDrawing(250;150)
  ChartData(0 7)
  LinearMeter(horizontal;
              10%;       /* Thickness relative to plot area height. */
              0;         /* Corner roundings.                       */
              0;         /* Scale line offset.                      */
              0;         /* Major tick mark offset.                 */
              0;         /* Minor tick mark offset.                 */
              0;         /* Symbol offset.                          */
              0;         /* Bar gap relative to bar thickness.      */
              0;         /* Bar shape. [0..16]                      */
              lightGray) /* Meter fill color.                       */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#499BC9;shaded)
  BorderStyle(1;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;5;4)
  AxisOptions(all;0) /* Hide scale. */
  AddText(217;79;"7/10";Arial;11;bold;#666;center;;0;150)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;15 40 15 15)
CloseDrawing()
```



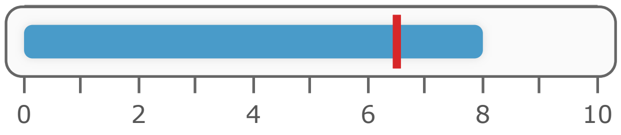
```
OpenDrawing(350;150)
  ChartData(0 7.54)
  LinearMeter(horizontal+shadow;
              20%;        /* Thickness relative to plot area height. */
              6;          /* Corner roundings.                       */
              0;          /* Scale line offset.                      */
              0;          /* Major tick mark offset.                 */
              0;          /* Minor tick mark offset.                 */
              0;          /* Symbol offset.                          */
              0.4;        /* Bar gap relative to bar thickness.      */
              0;          /* Bar shape. [0..16]                      */
              #fafafa;0;  /* Meter fill color.                       */
              1;#666)     /* Meter border.                           */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#4682b4)
  BorderStyle(1;none)
  ShadowStyle(all;0 0 3;lightGray)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;10;5)
  AxisLine(all;0)
  AxisMajorTicks(all;8;1;#666)
  AxisMinorTicks(all;4;1;#666)
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  AddText(297;72;"7.54";Verdana;12;bold;#666)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;14 50 10 12)
CloseDrawing()
```



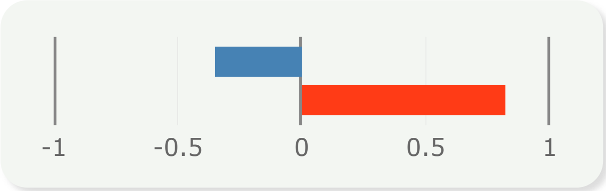
```
OpenDrawing(250;150)
  ChartData(0 8;6.5)
  LinearMeter(horizontal+shadow;
              20%;         /* Thickness relative to plot area height. */
              6;           /* Corner roundings.                       */
              0;           /* Scale line offset.                      */
              0;           /* Major tick mark offset.                 */
              0;           /* Minor tick mark offset.                 */
              0;           /* Symbol offset.                          */
              0.4;         /* Bar gap relative to bar thickness.      */
              6;           /* Bar shape. [0..16]                      */
              #fafafa;0;   /* Meter fill color.                       */
              1;#666)      /* Meter border.                           */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#499BC9)
  BorderStyle(1;none)
  ShadowStyle(1;0 0 3;lightGray)
  SymbolStyle(2;bullet;7;1;darkYellow;shaded)
  ShadowStyle(2;0)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;5)
  AxisLine(all;0)
  AxisMajorTicks(all;5;1;#666)
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;14 12 10 12)
CloseDrawing()
```



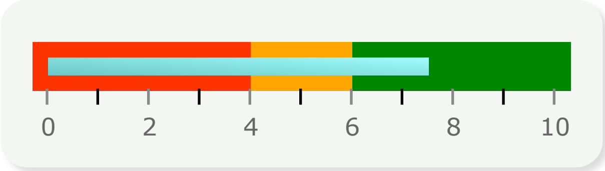
```
OpenDrawing(250;150)
  ChartData(0 8;6.5)
  LinearMeter(horizontal+shadow;
              20%;         /* Thickness relative to plot area height. */
              6;           /* Corner roundings.                       */
              0;           /* Scale line offset.                      */
              0;           /* Major tick mark offset.                 */
              0;           /* Minor tick mark offset.                 */
              0;           /* Symbol offset.                          */
              0.5;         /* Bar gap relative to bar thickness.      */
              6;           /* Bar shape. [0..16]                      */
              #fafafa;0;   /* Meter fill color.                       */
              1;#666)      /* Meter border stroke and color.          */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#499BC9)
  BorderStyle(1;none)
  ShadowStyle(1;0 0 3;lightGray)
  SymbolStyle(2;barVertical;20;3;#D72729)
  ShadowStyle(2;0)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;10)
  AxisLine(all;0)
  AxisMajorTicks(all;5;1;#666)
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#555)
  AxisMajorTickLabelTexts(all;"|u|";"")
CloseDrawing()
```



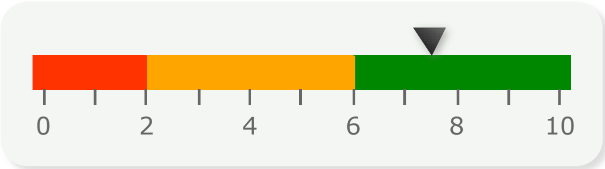
```
OpenDrawing(250;150)
  ChartData(-0.35 0;0 0.82)
  LinearMeter(horizontal;
              25%;         /* Thickness relative to plot area height. */
              8;           /* Corner roundings.                        */
              0;           /* Scale line offset.                       */
              0;           /* Major tick mark offset.                  */
              0;           /* Minor tick mark offset.                  */
              0;           /* Symbol offset.                           */
              0.3)         /* Bar gap relative to bar thickness.       */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#4682b4 )
  FillStyle(2;#FF3B16 )
  BorderStyle(all;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;-1;1;4)
  AxisLine(all;0)
  AxisMajorTicks(all;0)
  AxisMajorTickLabelTexts(all;"|u|")
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  /* Use grid line and stripe functions to control the background behind the bar(s). */
  MajorGridLineWidths(all;all;1;0.25)
  MajorGridLineColors(all;all;gray;lightGray)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;14 12 10 12)
CloseDrawing()
```



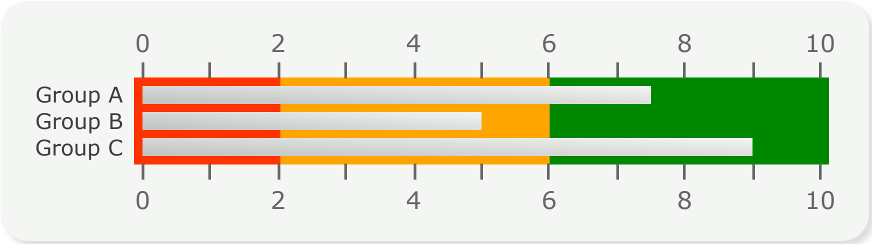
```
OpenDrawing(250;150)
  ChartData(0 7.5)
  LinearMeter(horizontal;
              14%;         /* Thickness relative to plot area height. */
              0;           /* Corner roundings.                       */
              0;           /* Scale line offset.                      */
              0;           /* Major tick mark offset.                 */
              0;           /* Minor tick mark offset.                 */
              0;           /* Symbol offset.                          */
              0.9)         /* Bar gap relative to bar thickness.      */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#8ff;shaded)
  BorderStyle(1;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;5;2)
  AxisLine(all;0)
  AxisMajorTicks(all;5;1;#666)
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  /* Use grid line and stripe functions to control the background behind the bar(s). */
  MajorGridLineWidths(all;all;0)
  MinorGridLineWidths(all;all;0)
  MajorGridStripeColors(all;all;red;red;orange;green;green)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;16 12 10 12)
CloseDrawing()
```



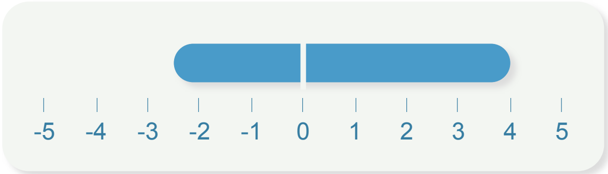
```
OpenDrawing(250;150)
  ChartData(7.5)
  LinearMeter(horizontal+shadow;
              10%;      /* Thickness relative to plot area height. */
              0;        /* Corner roundings.                       */
              0;        /* Scale line offset.                      */
              0;        /* Major tick mark offset.                 */
              0;        /* Minor tick mark offset.                 */
            -12;        /* Symbol offset.                          */
            0.8)        /* Bar gap relative to bar thickness.      */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  SymbolStyle(1;triangleDown;8;3;black;shaded)
  ShadowStyle(1;1 1 3)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;10)
  AxisLine(all;0)
  AxisMajorTicks(all;5;1;#666)
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  AxisMajorTickLabelTexts(all;"|u|";"")
  /* Use grid line and stripe functions to control the background behind the bar(s). */

MajorGridLineColors(all;all;red;red;red;orange;orange;orange;orange;green;green;green;green)
  MajorGridStripeColors(all;all;red;red;orange;orange;orange;orange;green;green;green;green)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;20 12 10 12)
CloseDrawing()
```



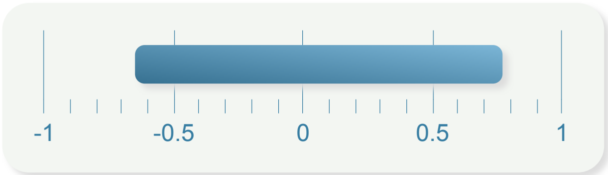
```
OpenDrawing(350;150)
  ChartData(0 7.5;0 5;0 9)
  LinearMeter(horizontal;
              25%;         /* Thickness relative to plot area height. */
              0;           /* Corner roundings.                       */
              0;           /* Scale line offset.                      */
              0;           /* Major tick mark offset.                 */
              0;           /* Minor tick mark offset.                 */
              0;           /* Symbol offset.                          */
              0.5)         /* Bar gap relative to bar thickness.      */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(all;#f3f6f2;shaded)
  BorderStyle(all;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;10)
  AxisLine(all;0)
  AxisMajorTicks(all;5;1;#666)
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  AxisOptions(all;;2) /* Mirror scale. */
  AxisMajorTickLabelTexts(all;"|u|";"")
  /* Use grid line and stripe functions to control the background behind the bar(s). */

MajorGridLineColors(all;all;red;red;red;orange;orange;orange;orange;green;green;green;green)
  MajorGridStripeColors(all;all;red;red;orange;orange;orange;orange;green;green;green;green)
  AddText(56;68;"Group A";Verdana;8;plain;60 60 60;right)
  AddText(56;78;"Group B";Verdana;8;plain;60 60 60;right)
  AddText(56;88;"Group C";Verdana;8;plain;60 60 60;right)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;10 15 10 50)
CloseDrawing()
```



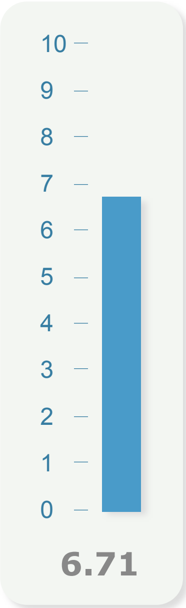
```
OpenDrawing(250;150)
  ChartData(-2.5 4;0)
  LinearMeter(horizontal+shadow;
              20%;         /* Thickness relative to plot area height. */
              0;           /* Corner roundings.                       */
              0;           /* Scale line offset.                      */
              0;           /* Major tick mark offset.                 */
              0;           /* Minor tick mark offset.                 */
              0;           /* Marker offset.                          */
              0.4;         /* Bar gap relative to bar thickness.      */
              3)           /* Bar shape. [0..16]                      */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#499BC9)
  BorderStyle(1;none)
  ShadowStyle(1;2 2 2;lightGray)
  SymbolStyle(2;barVertical;20;2;#f3f6f2)
  ShadowStyle(2;0)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;-5;5;10)
  AxisLine(all;0)
  AxisMajorTicks(all;5;0.25;#367DA2)
  AxisMajorTickLabelStyle(all;;;;#367DA2)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;10)
CloseDrawing()
```



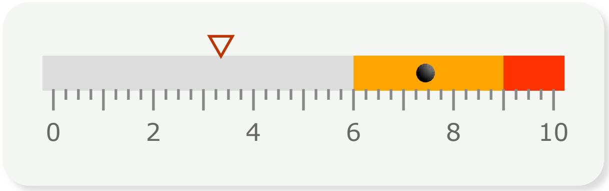
```
OpenDrawing(250;150)
  ChartData(-0.65 0.77)
  LinearMeter(horizontal+shadow;
              20%;         /* Thickness relative to plot area height. */
              0;           /* Corner roundings.                       */
              0;           /* Scale line offset.                      */
              0;           /* Major tick mark offset.                 */
              0;           /* Minor tick mark offset.                 */
              0;           /* Symbol offset.                          */
              0.4;         /* Bar gap relative to bar thickness.      */
              6)           /* Bar shape. [0..16]                      */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#499BC9;shaded)
  BorderStyle(1;none)
  ShadowStyle(1;2 2 2;lightGray)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;-1;1;4;5)
  AxisLine(all;0)
  AxisMajorTicks(all;5;0.25;#367DA2)
  AxisMinorTicks(all;5;0.25;#367DA2)
  AxisMajorTickLabelStyle(all;;;;#367DA2)
  AxisMajorTickLabelTexts(all;"|u|")
  /* Use grid line and stripe functions to control the background behind the bar(s). */
  MajorGridLineWidths(all;all;0.25)
  MinorGridLineWidths(all;all;0)
  MajorGridLineColors(all;all;#367DA2)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;10)
CloseDrawing()
```



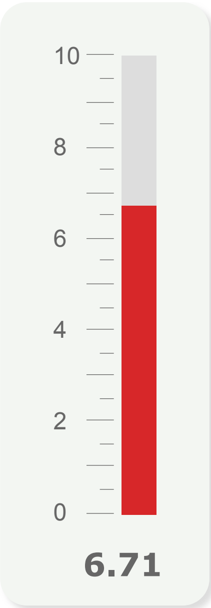
```
OpenDrawing(150;250)
  ChartData(0 6.71)
  LinearMeter(shadow+symbol;
              20%;        /* Thickness relative to plot area width. */
              0;          /* Corner roundings.                      */
              0;          /* Scale line offset.                     */
              0;          /* Major tick mark offset.                */
              0;          /* Minor tick mark offset.                */
              0;          /* Symbol offset.                         */
              0.4)        /* Bar gap relative to bar thickness.     */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#499BC9)
  BorderStyle(1;none)
  ShadowStyle(1;2 2 2;lightGray)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;10)
  AxisLine(all;0)
  AxisMajorTicks(all;5;0.25;#367DA2)
  AxisMajorTickLabelStyle(all;;;;#367DA2)
  AddText(78;225;"6.71";Verdana;12;bold;#888;center;;0;150)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;10 10 30 15)
CloseDrawing()
```



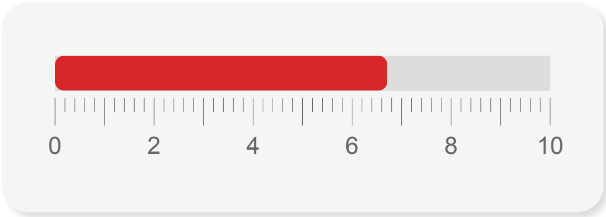
```
OpenDrawing(250;150)
  ChartData(3.35;7.44)
  LinearMeter(horizontal;
              10%;         /* Thickness relative to plot area height. */
              0;           /* Corner roundings.                       */
              0;           /* Scale line offset.                      */
              0;           /* Major tick mark offset.                 */
              0;           /* Minor tick mark offset.                 */
             -10 0;        /* Marker offsets.                         */
              0.8)         /* Bar gap relative to bar thickness.      */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  SymbolStyle(1;triangleDownOutline;10;1;darkRed)
  SymbolStyle(2;bullet;6;1;black;shaded)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;10;4)
  AxisLine(all;0)
  AxisMajorTicks(all;7;1;#888)
  AxisMinorTicks(all;3;1;#888)
  AxisMajorTickLabelTexts(all;"|u|";"")
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  /* Use grid line and stripe functions to control the background behind the bar(s). */
  MajorGridLineWidths(all;all;0)
  MinorGridLineWidths(all;all;0)
  MajorGridStripeColors(all;all;#ddd;#ddd;#ddd;#ddd;#ddd;#ddd;orange;orange;orange;red)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;20 15 15)
CloseDrawing()
```



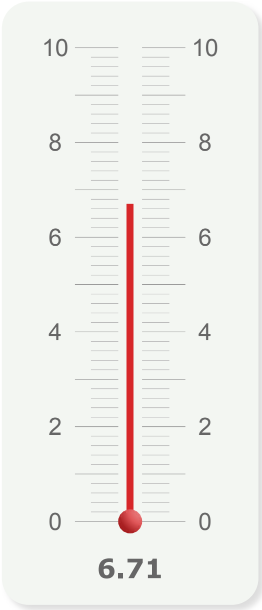
```
OpenDrawing(150;250)
  ChartData(0 6.71)
  LinearMeter(default;
              10%;         /* Thickness relative to plot area width. */
              0;           /* Corner roundings.                      */
              0;           /* Scale line offset.                     */
              3;           /* Major tick mark offset.                */
              3;           /* Minor tick mark offset.                */
              0;           /* Symbol offset.                         */
              0;           /* Bar gap relative to bar thickness.     */
              0;           /* Bar shape. [0..16]                     */
              lightGray)   /* Meter fill color.                      */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#D72729)
  BorderStyle(1;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;10;2)
  AxisLine(all;0)
  AxisMajorTicks(all;10;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelTexts(all;"|u|";"")
  AxisMajorTickLabelStyle(all;;;;#666)
  AddText(82;225;"6.71";Verdana;12;bold;#666;center;;0;150)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;20 20 35 20)
CloseDrawing()
```



```
OpenDrawing(250;150)
  ChartData(0 6.71)
  LinearMeter(horizontal;
              10%;         /* Thickness relative to plot area height. */
              0;           /* Corner roundings.                       */
              0;           /* Scale line offset.                      */
              3;           /* Major tick mark offset.                 */
              3;           /* Minor tick mark offset.                 */
              0;           /* Symbol offset.                          */
              0;           /* Bar gap relative to bar thickness.      */
              6;           /* Bar shape. [0..16]                      */
              lightGray;0; /* Meter fill color.                       */
              0)           /* Meter border.                           */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  FillStyle(1;#D72729)
  BorderStyle(1;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;10;5)
  AxisLine(all;0)
  AxisMajorTicks(all;10;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelTexts(all;"|u|";"")
  AxisMajorTickLabelStyle(all;;;;#666)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;20)
CloseDrawing()
```



```
OpenDrawing(150;250)
  ChartData(0; 0 6.71)
  LinearMeter(default;
              4%;   /* Thickness relative to plot area width. */
              0;    /* Corner roundings.                      */
              0;    /* Scale line offset.                     */
              2;    /* Major tick mark offset.                */
              2)    /* Minor tick mark offset.                */
  /* Use style functions to control the appearance of bar(s) and symbol(s). */
  SymbolStyle(1;bullet;8;1;#D72729;shaded)
  BorderStyle(2;none)
  FillStyle(2;#D72729)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;10;5)
  AxisLine(all;0)
  AxisOptions(all;;2) /* Mirror axis. */
  AxisMajorTicks(all;16;0.25;#999)
  AxisMinorTicks(all;10;0.25;#bbb)
  AxisMajorTickLabelTexts(all;" 0";"";" 2";"";" 4";"";" 6";"";" 8";"";"10")
  AxisMajorTickLabelStyle(all;;;;#666)
  AddText(75;226;"6.71";Verdana;10;bold;#666;center;;0;150)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;16 15 30)
CloseDrawing()
```



Linear Gauges

```
LinearGauge(appearance;thickness;themeID;symbolOffset;scaleLineOffset;
            majorTickLabelOffset;minorTickLabelOffset;attachTickMarkToScaleEnds)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
appearance	int	0..127	0	*Const: shadow,label,symbol,horiz.*
thickness	num	0..10000	20 %	*Abs or rel in percent of plot area*
themeID	int	0..2	1	
symbolOffset	num	-1000..1000	0	
scaleLineOffset	num	-1000..1000	0	
majorTickLabelOffset	num	-1000..1000	0	
minorTickLabelOffset	num	-1000..1000	0	
attachTickMarkToScaleEnds	int	0..1	0	

The position of a symbol is defined by entering the value in `ChartData()`. Values for multiple symbols are separated by a semicolon ";". For example:

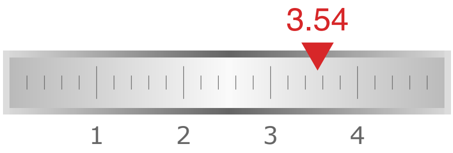
```
ChartData(56;   /* Symbol1 at 56   */
          17.5) /* Symbol2 at 17.5 */
```

The appearance of the symbols can be controlled by the `SymbolStyle()` function; the appearance of the symbol labels by the functions `LabelTexts()`, `LabelStyle()`, `LabelOptions()`, `LabelBackground()` and `LabelBackgroundOptions()`.
By using scale and axis functions the appearance of the scale can be controlled.

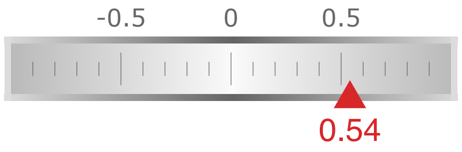
Legend and title are not supported in combination with linear gauges.

Examples:

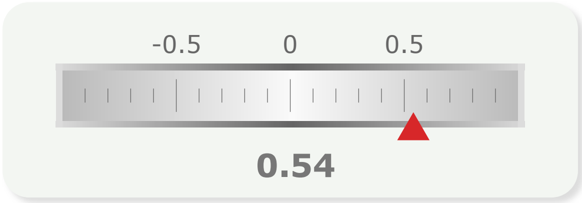
```
OpenDrawing(200;100)
  ChartData(3.54)
  LinearGauge(horizontal+label;
              30%;   /* Thickness relative to plot area height. */
              1;     /* Theme ID. [0..2]                        */
             -10;    /* Symbol offset.                          */
              0;     /* Scale line offset.                      */
              14)    /* Major tick label offset.                */
  /* Use style functions to control the appearance of the symbol(s). */
  SymbolStyle(1;triangleDown;12;1;#D72729)
  LabelTexts(1;"|u|")
  LabelStyle(1;Arial;12;plain;#D72729)
  LabelOptions(1;topCenter;;11)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;5;5;5)
  AxisLine(all;0)
  AxisMajorTicks(all;12;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
CloseDrawing()
```



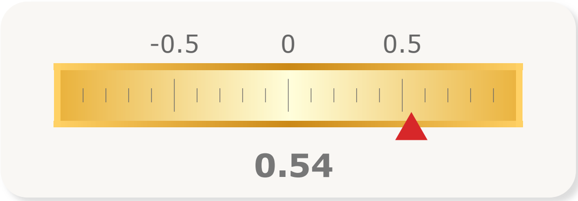
```
OpenDrawing(200;100)
  ChartData(0.54)
  LinearGauge(horizontal+label;
              30% ;          /* Thickness relative to plot area height. */
              1;             /* Theme ID. [0..2]                        */
              10;            /* Symbol offset.                          */
              0;             /* Scale line offset.                      */
             -25)            /* Major tick label offset.                */
  /* Use style functions to control the appearance of the symbol(s). */
  SymbolStyle(1;triangleUp;10;2;#D72729)
  LabelTexts(1;"|u|")
  LabelStyle(1;Arial;12;plain;#D72729)
  LabelOptions(1;bottomCenter;;01)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;-1;1;4;5)
  AxisLine(all;0)
  AxisMajorTicks(all;12;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelTexts(all;"|u|")
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
CloseDrawing()
```



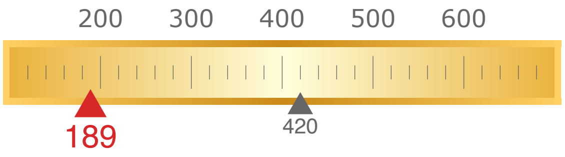
```
OpenDrawing(240;100)
  ChartData(0.54)
  LinearGauge(horizontal;
              30%;         /* Thickness relative to plot area height. */
              1;           /* Theme ID. [0..2]                        */
              12;          /* Symbol offset.                          */
              0;           /* Scale line offset.                      */
             -25)          /* Major tick label offset.                */
  /* Use style functions to control the appearance of the symbol(s). */
  SymbolStyle(1;triangleUp;10;2;#D72729)
  LabelTexts(1;"|u|")
  LabelStyle(1;Arial;12;plain;#D72729)
  LabelOptions(1;bottomCenter;;01)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;-1;1;4;5)
  AxisLine(all;0)
  AxisMajorTicks(all;12;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelTexts(all;"|u|")
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  AddText(120;74;"0.54";Verdana;12;bold;#777;center;center;0;200)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;10 20 20 20)
CloseDrawing()
```



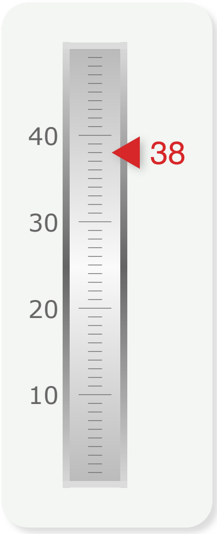
```
OpenDrawing(240;100)
  ChartData(0.54)
  LinearGauge(horizontal;
              30%;         /* Thickness relative to plot area height. */
              2;           /* Theme ID. [0..2]                        */
              12;          /* Symbol offset.                          */
              0;           /* Scale line offset.                      */
             -25)          /* Major tick label offset.                */
  /* Use style functions to control the appearance of the symbol(s). */
  SymbolStyle(1;triangleUp;10;2;#D72729)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;-1;1;4;5)
  AxisLine(all;0)
  AxisMajorTicks(all;12;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelTexts(all;"|u|")
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  AddText(120;74;”0.54";Verdana;12;bold;#777;center;center;0;200)
  ChartBackground(xy;#f9f7f4;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;10 20 20 20)
CloseDrawing()
```



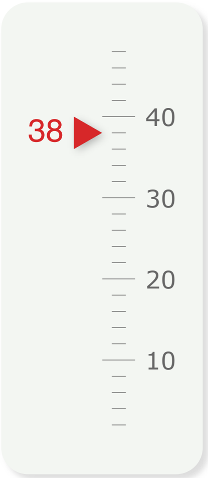
```
OpenDrawing(240;100)
  ChartData(189;420)
  LinearGauge(horizontal+label;
              30%;         /* Thickness relative to plot area height. */
              2;           /* Theme ID. [0..2]                         */
              12;          /* Symbol offset.                           */
              0;           /* Scale line offset.                       */
              -26)         /* Major tick label offset.                 */
  /* Use style functions to control the appearance of the symbol(s). */
  SymbolStyle(1;triangleUp;10;2;#D72729)
  SymbolStyle(2;triangleUp;7;2;#666)
  LabelTexts(all;"|i0|")
  LabelStyle(1;Arial;12;plain;#D72729)
  LabelStyle(2;Arial;8;plain;#666)
  LabelOptions(all;bottomCenter;;00)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;100;700;6;5)
  AxisLine(all;0)
  AxisMajorTicks(all;12;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelTexts(all;"|u|")
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
CloseDrawing()
```



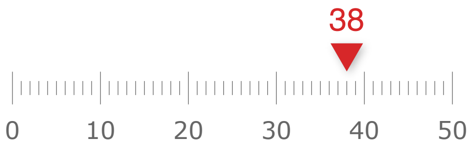
```
OpenDrawing(100;220)
  ChartData(38)
  LinearGauge(label+shadow;
              30%;         /* Thickness relative to plot area width. */
              1;           /* Theme ID. [0..2]                       */
              12;          /* Symbol offset.                         */
              0;           /* Scale line offset.                     */
              -25)         /* Major tick label offset.               */
  /* Use style functions to control the appearance of the symbol(s). */
  SymbolStyle(1;triangleLeft;10;2;#D72729)
  ShadowStyle(1;1 1 3)
  LabelTexts(1;"|u|")
  LabelStyle(1;Arial;12;plain;#D72729)
  LabelOptions(1;centerRight;;01)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;50;5;10)
  AxisLine(all;0)
  AxisMajorTicks(all;12;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelTexts(all;"|u|")
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;15 10)
CloseDrawing()
```



```
OpenDrawing(100;200)
  ChartData(38)
  LinearGauge(label+shadow;
              30%;          /* Thickness relative to plot area width. */
              0;            /* Theme ID. [0..2]                       */
              -12;          /* Symbol offset.                         */
              0;            /* Scale line offset.                     */
              10)           /* Major tick label offset.               */
  /* Use style functions to control the appearance of the symbol(s). */
  SymbolStyle(1;triangleRight;10;2;#D72729)
  ShadowStyle(1;1 1 3)
  LabelTexts(1;"|u|")
  LabelStyle(1;Arial;12;plain;#D72729)
  LabelOptions(1;centerLeft;10)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;50;5;5)
  AxisLine(all;0)
  AxisMajorTicks(all;12;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelTexts(all;"|u|")
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
  ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(10;10)
CloseDrawing()
```



```
OpenDrawing(200;100)
  ChartData(38)
  LinearGauge(horizontal+label+shadow;
              30%;  /* Thickness relative to plot area height. */
              0;    /* Theme ID. [0..2]                        */
              -12;  /* Symbol offset.                          */
              0;    /* Scale line offset.                      */
              10;   /* Major tick label offset.                */
              0;    /* Minor tick label offset.                */
              1)    /* Attach tick mark to scale ends.         */
  /* Use style functions to control the appearance of the symbol(s). */
  SymbolStyle(1;triangleDown;10;2;#D72729)
  ShadowStyle(1;1 1 3)
  LabelTexts(1;"|u|")
  LabelStyle(1;Arial;12;plain;#D72729)
  LabelOptions(1;topCenter;;10)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;50;5;10)
  AxisLine(all;0)
  AxisMajorTicks(all;12;0.25;#666)
  AxisMinorTicks(all;5;0.25;#666)
  AxisMajorTickLabelTexts(all;"|u|")
  AxisMajorTickLabelStyle(all;Verdana;9;plain;#666)
CloseDrawing()
```



Radial Gauges

```
RadialGauge(appearance;startAngle;arcAngle;backgroundAsFullCircle;needleLengths;
            needleThicknesses;needleHeadShape;needleTailShape;needlePivotDiameters;
            needlePivotColor;scaleRadius;scaleBackgroundExtends;scaleBackgroundBounds;
            scaleBackgroundShading;scaleBackgroundColor1;
            scaleBackgroundColor2;...;scaleBackgroundColor100)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
appearance	int	0..127	0	*Constants: shadow*
startAngle	num	-360..360	-90	
arcAngle	num	-360..360	180	*If negative, go counter-clockwise.*
backgroundAsFullCircle	int	0..1	0	
needleLengths	num[]	0..2	1 0 0	*[headLen tailLen offsetFromCenter]*
needleThicknesses	num[]	0..1	0.03 0.03 0	*[atCenter atHead colorSplitFactor]*
needleHeadShape	num[]	1..3	1 1	
needleTailShape	num[]	1..2	1 0	
needlePivotDiameters	num[]	-1..1	0 0	*[outerDiameter innerDiameter]*
needlePivotColor	rgba	0..255	none	
scaleRadius	num	0..1	1	
scaleBackgroundExtends	num[]	0..0.5	0 0	*[atScaleStart atScaleEnd]*
scaleBackgroundBounds	num[]	0..2	1 0.9	
scaleBackgroundShading	int	-1..128	0	
scaleBackgroundColor1	rgba	0..255	none	
scaleBackgroundColor2	rgba	0..255	none	
...	rgba	0..255	none	
scaleBackgroundColor100	rgba	0..255	none	

Needles:

The fill, border and shadow effect of the needle(s) can be controlled by the style functions `FillStyle()`, `BorderStyle()` and `ShadowStyle()`.

The argument *needleLengths* can have up to 3 values, i.e. the length of the needle head, the length of an optional needle tail and an also optional needle offset from the pivot center. All 3 values have to be entered relative to the gauge radius. If the optional needle offset is > 1 the head of the needle is drawn in the opposite direction, i.e. the needle points to the pivot center. This proves useful when depicting a marker on the outer edge of the scale. The needle length values can also be entered repeatedly if more than one needle is be displayed. For example:
... ; 1.2 0 0.2 1 0.2 0 0.85 0 0 ; ...

The argument *needleThicknesses* can have up to 3 values, i.e. the thickness of the needle at the pivot center, the thickness at the needle head and an optional color split factor. For a color split factor > 0 [0..0.5] the needle halves are drawn in slightly different color tones.
The thickness values of the needle have to be entered relative to the gauge radius. The three values can also be entered repeatedly if more than one needle is be displayed. For example:
... ; 0.2 0.01 0.15 0.05 0.01 0 0.01 0.01 0 ; ...

The shape of the needle head is controlled by using the argument *needleHeadShape*.
Three head shapes are supported. As an option, a shape parameter can be added to fine tune the appearance of the head shape.
1...Peak. The length of the peak can be controlled by entering a second value between [-1..1].
2...Round. The rounding can be controlled by entering a second value between [0..1].
3...Arrow. The appearance of the arrowhead can be controlled by entering a second value between [-1..0.5].
The head shape values can also be entered repeatedly if more than one needle is be displayed.
For example: *... ; 1 0 2 0.5 2 0.5 ; ...*

The shape of the needle tail is controlled by using the argument *needleTailShape*.
Two tail shapes are supported. As an option, a shape parameter can be added to fine tune the appearance of the tail shape.
1...Peak. The length of the peak can be controlled by entering a second value between [-1..1].
2...Round. The rounding can be controlled by entering a second value between [0..1].
The tail shape values can also be entered repeatedly if more than one needle is be displayed.
For example: *... ; 2 1 2 1 2 1 ; ...*

The argument *needlePivotDiameters* can have up to 2 values, i.e. the outer diameter of the needle pivot, and an optional inner diameter. Both diameters have to be entered relative to the gauge radius.

The color of the needle pivot can be varied using the argument *needlePivotColor*.

Scale:

The argument *scaleRadius* has to be entered relative to the gauge radius between [0..1].

The argument *scaleBackgroundExtends* can have up to 2 values, i.e. the background extends at the beginning and on the end of the scale. Both background extends have to be entered relative to the *arcAngle* of the gauge between [0..0.5].

The argument *scaleBackgroundBounds* can have up to 5 values:
[1]: The outer radius of the scale background, between [0..1].
[2]: The inner radius of the scale background at the beginning, between [0..1].
[3]: The inner radius of the scale background on the end, between [0..1].
[4]: The start angle of the scale background, between [0..1].
[5]: The end angle of the scale background, between [0..1].

The outer radius and the inner radii have to be entered relative to the gauge radius, the start and end angles relative to the *arcAngle* of the gauge. By entering the inner radii and the start and end angles repeatedly, several background sections can be defined. For example:

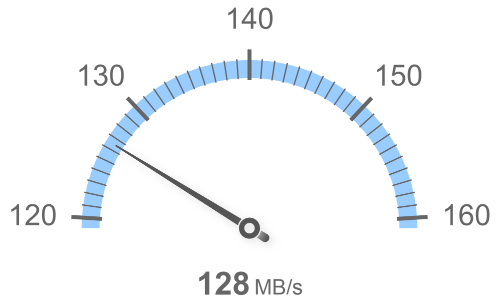
```
... ;
0.65               /* Scale background outerRadius.      */
0.61 0.61 0.0 0.7  /* Scale background section bounds 1. */
0.61 0.61 0.7 0.9  /* Scale background section bounds 2. */
0.00 0.00 0.9 1.0; /* Scale background section bounds 3. */
...
```

The color of the scale background is controlled by using the arguments *scaleBackgroundShading* and *scaleBackgroundColor*. If the number of background sections is larger than the number of defined scale background colors, the colors will be repeated periodically.

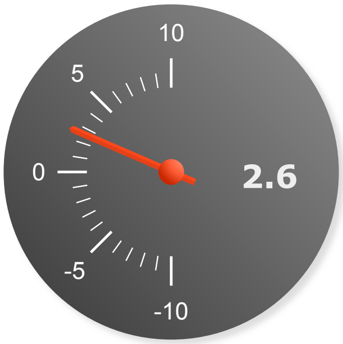
Legend and title are not supported in combination with radial gauges.

Examples:

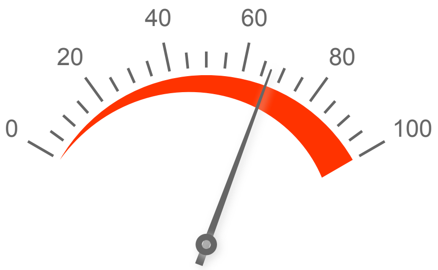
```
OpenDrawing(200;140)
  OpenChart(0;0;200;200)
    ChartData(126.5)
    RadialGauge(shadow;
               -90;             /* Start angle in [deg].                                  */
                180;            /* Arc angle in [deg]. If negative, go counter-clockwise. */
                0;              /* Background as full circle.                             */
                1.05 0.15 0.08; /* Needle length: [head tail offsetFromCenter]            */
                0.05 0.01 0.00; /* Needle thickness: [atCenter atHead colorSplitFactor]   */
                1;              /* Needle head shape. 1...peak, 2...round, 3...arrow      */
                2;              /* Needle tail shape. 1...peak, 2...round                 */
                0.14 0.07;      /* Needle pivot diameters: [out in]                       */
                #555;           /* Needle pivot color.                                    */
                1;              /* Scale radius.                                          */
                0.02;           /* Scale background extend: [atStart atEnd]               */
                1.12            /* Scale background outerRadius.                          */
                1 1 0.0 1;      /* Scale background section bounds.                       */
                0;              /* Scale background shading. 0...solid, -1...shaded       */
                lightBlue)      /* Scale background colors.                               */
    /* Use style functions to control the appearance of the needle. */
    FillStyle(1;#555)
    BorderStyle(1;none)
    ShadowStyle(all;1 1 3;lightGray)
    /* Use scale and axis functions to control the appearance of the scale. */
    Scaling(all;linear;120;160;4;10)
    AxisLine(all;0) /* Hide axis line. */
    AxisMajorTicks(all;10;1.5;#666;;out)
    AxisMinorTicks(all;7;0.5;#666;;out)
    AxisMajorTickLabelStyle(all;Arial;11;plain;#666)
    AxisMajorTickLabelOptions(all;out)
    AddText(100;125;"<b>128</b><span sizeY=8>\xE2\x80\xAFMB/s</
span>”;Arial;12;plain;#666;center)
  CloseChart()
  // Background() /* Uncomment while positioning the gauge. */
CloseDrawing()
```



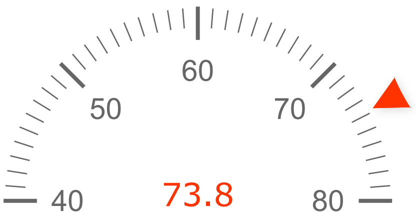
```
OpenDrawing(150;150)
  ChartData(2.6)
  RadialGauge(default;
              180;           /* Start angle in [deg].                                  */
              180;           /* Arc angle in [deg]. If negative, go counter-clockwise. */
              1;             /* Background as full circle.                             */
              1.30 0.3;      /* Needle length: [head tail offsetFromCenter]            */
              0.08 0.08 0.1; /* Needle thickness: [atCenter atHead colorSplitFactor]   */
              1;             /* Needle head shape. 1...peak, 2...round, 3...arrow      */
              1;             /* Needle tail shape. 1...peak, 2...round                 */
              0.3;           /* Needle pivot diameters: [out in]                       */
              red)           /* Needle pivot color.                                    */
  /* Use style functions to control the appearance of the needle. */
  FillStyle(1;red)
  BorderStyle(all;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;-10;10;4;5)
  AxisLine(all;0)
  AxisMajorTicks(all;10;1;white;;out)
  AxisMinorTicks(all;5;0.5;white;;out)
  AxisMajorTickLabelStyle(all;Arial;9;plain;white)
  AddText(110;74;"2.6";Verdana;12;bold;#eee;center;center)
  ChartBackground(xy;#555;shaded;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(0;6)
CloseDrawing()
```



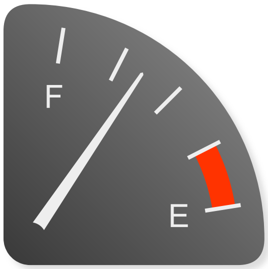
```
OpenDrawing(200;120)
  OpenChart(0;0;200;200)
    ChartData(67)
    RadialGauge(shadow;
               -60;             /* Start angle in [deg].                                  */
                120;            /* Arc angle in [deg]. If negative, go counter-clockwise. */
                0;              /* Background as full circle.                             */
                1.05 0.12;      /* Needle length: [head tail offsetFromCenter]            */
                0.04 0.01;      /* Needle thickness: [atCenter atHead colorSplitFactor]   */
                1;              /* Needle head shape. 1...peak, 2...round, 3...arrow      */
                1;              /* Needle tail shape. 1...peak, 2...round                 */
                0.12 0.05;      /* Needle pivot diameters: [out in]                       */
                #666;           /* Needle pivot color.                                    */
                1;              /* Scale radius.                                          */
                0;              /* Scale background extend: [atStart atEnd]               */
                0.95            /* Scale background outerRadius.                          */
                0.95 0.75 0.0 1.0;/* Scale background section bounds.                     */
                0;              /* Scale background shading. 0...solid, -1...shaded       */
                red)            /* Scale background color.                                */
    /* Use style functions to control the appearance of the needle. */
    FillStyle(1;#666)
    BorderStyle(1;none)
    ShadowStyle(all;1 1 3;lightGray)
    /* Use scale and axis functions to control the appearance of the scale. */
    Scaling(all;linear;0;100;5;4)
    AxisLine(all;0)
    AxisMajorTicks(all;10;1;#666;;out)
    AxisMinorTicks(all;5;1;#666;;out)
    AxisMajorTickLabelStyle(all;Arial;9;plain;#666)
  CloseChart()
  // Background() /* Uncomment while positioning the gauge. */
CloseDrawing()
```



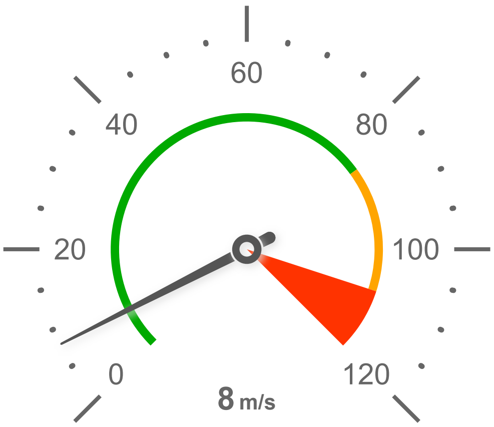
```
OpenDrawing(200;120)
  OpenChart(0;0;200;200)
    ChartData(73.8)
    RadialGauge(shadow;
               -90;             /* Start angle in [deg].                                  */
                180;            /* Arc angle in [deg]. If negative, go counter-clockwise. */
                0;              /* Background as full circle.                             */
                1.20 0.00 1.03; /* Needle length: [head tail offsetFromCenter]            */
                0.18 0.18 0.00; /* Needle thickness: [atCenter atHead colorSplitFactor]   */
                1 1;            /* Needle head shape. 1...peak, 2...round, 3...arrow      */
                1;              /* Needle tail shape. 1...peak, 2...round                 */
                0;              /* Needle pivot diameters: [out in]                       */
                none;           /* Needle pivot color.                                    */
                0.8)            /* Scale radius.                                          */
    /* Use style functions to control the appearance of the marker. */
    FillStyle(1;red)
    BorderStyle(1;none)
    ShadowStyle(all;1 1 3;lightGray)
    /* Use scale and axis functions to control the appearance of the scale. */
    Scaling(all;linear;40;80;4;10)
    AxisLine(all;0) /* Hide axis line. */
    AxisMajorTicks(all;11;1.5;#666;;in)
    AxisMinorTicks(all;7;0.5;#666;;in)
    AxisMajorTickLabelStyle(all;Arial;11;plain;#666)
    AxisMajorTickLabelOptions(all;in)
    AddText(100;102;"73.8";Verdana;12;plain;red;center)
  CloseChart()
  // Background() /* Uncomment while positioning the gauge. */
CloseDrawing()
```



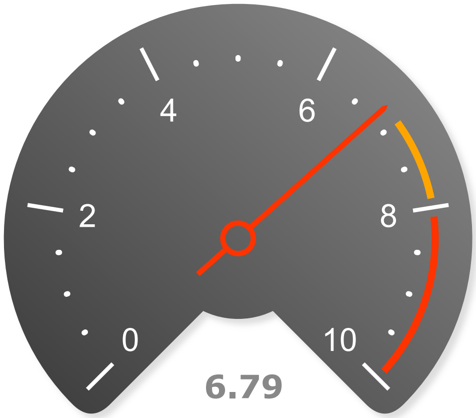
```
OpenDrawing(120;120)
  OpenChart(-75;0;200;200)
    ChartData(0.65)
    RadialGauge(default;
                90;             /* Start angle in [deg].                                  */
                -90;            /* Arc angle in [deg]. If negative, go counter-clockwise. */
                0;              /* Background as full circle.                             */
                1.12 0.00 0.09; /* Needle length: [head tail offsetFromCenter]            */
                0.08 0.02 0;    /* Needle thickness: [atCenter atHead colorSplitFactor]   */
                1;              /* Needle head shape. 1...peak, 2...round, 3...arrow      */
                1;              /* Needle tail shape. 1...peak, 2...round                 */
                0;              /* Needle pivot diameters: [out in]                       */
                none;           /* Needle pivot color.                                    */
                1;              /* Scale radius.                                          */
                0.1;            /* Scale background extend: [atStart atEnd]               */
                1.2             /* Scale background outerRadius.                          */
                1.03 1.03 0.1 0.3; /* Scale background section bounds.                    */
                0;                 /* Scale background shading. 0...solid, -1...shaded    */
                red)               /* Scale background color.                             */
    /* Use style functions to control the appearance of the needle. */
    FillStyle(1;#eee)
    BorderStyle(1;none)
    /* Use scale and axis functions to control the appearance of the scale. */
    Scaling(all;linear;0;1;4)
    AxisLine(all;0)
    AxisMajorTicks(all;12;1.5;#eee;;out)
    AxisMajorTickLabelTexts(all;"E";"";"";"";"F")
    AxisMajorTickLabelStyle(all;Arial;12;plain;#eee)
    AxisMajorTickLabelOptions(all;in)
    ChartBackground(xy;#555;shaded;0;;;2 2 2;lightGray)
    ChartBackgroundOptions(0;10)
   CloseChart()
   // Background() /* Uncomment while positioning the gauge. */
CloseDrawing()
```



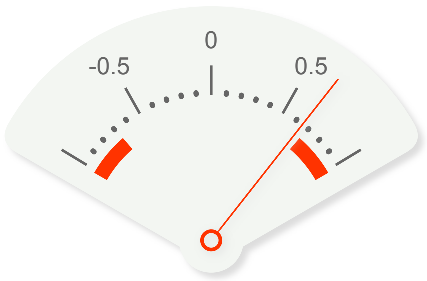
```
OpenDrawing(200;200)
  ChartData(8)
  RadialGauge(shadow;
             -135;              /* Start angle in [deg].                                  */
              270;              /* Arc angle in [deg]. If negative, go counter-clockwise. */
              0;                /* Background as full circle.                             */
              1.00 0.15 0.08;   /* Needle length: [head tail offsetFromCenter]            */
              0.05 0.01 0.00;   /* Needle thickness: [atCenter atHead colorSplitFactor]   */
              1;                /* Needle head shape. 1...peak, 2...round, 3...arrow      */
              2;                /* Needle tail shape. 1...peak, 2...round                 */
              0.14 0.07;        /* Needle pivot diameters: [out in]                       */
              #555;             /* Needle pivot color.                                    */
              1;                /* Scale radius.                                          */
              0;                /* Scale background extend: [atStart atEnd]               */
              0.65              /* Scale background outerRadius.                          */
              0.61 0.61 0.0 0.7 /* Scale background section bounds 1. (green)             */
              0.61 0.61 0.7 0.9 /* Scale background section bounds 2. (orange)            */
              0.00 0.00 0.9 1.0;/* Scale background section bounds 3. (red)               */
              0;                /* Scale background shading. 0...solid, -1...shaded       */
              #0a0;orange;red)  /* Scale background colors.                               */
  /* Use style functions to control the appearance of the needle. */
  FillStyle(1;#555)
  BorderStyle(1;none)
  ShadowStyle(all;1 1 3;lightGray)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;120;6;4)
  AxisLine(all;0) /* Hide axis line. */
  AxisMajorTicks(all;12;1.5;#666;;out)
  AxisMinorTicks(all;0.5;2 2 2;#666;;out)
  AxisMajorTickLabelStyle(all;Arial;11;plain;#666)
  AxisMajorTickLabelOptions(all;in;-0)
  AddText(100;160;"8<span size=8>\xE2\x80\xAFm/s</span>";Arial;12;bold;#666;center)
CloseDrawing()
```



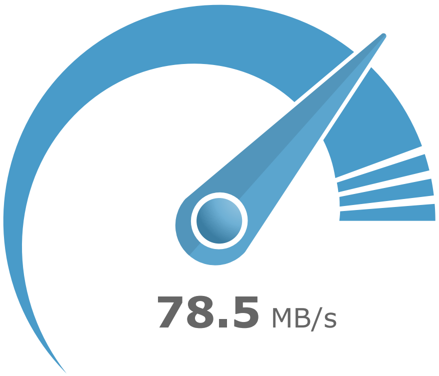
```
OpenDrawing(200;200)
  ChartData(6.79)
  RadialGauge(default;
             -135;              /* Start angle in [deg].                                  */
              270;              /* Arc angle in [deg]. If negative, go counter-clockwise. */
              0;                /* Background as full circle.                             */
              1.12 0.3 0.1;     /* Needle length: [head tail offsetFromCenter]            */
              0.03;             /* Needle thickness: [atCenter atHead colorSplitFactor]   */
              1 1;              /* Needle head shape. 1...peak, 2...round, 3...arrow      */
              1 0;              /* Needle tail shape. 1...peak, 2...round                 */
              0.2 0.14;         /* Needle pivot diameters: [out in]                       */
              red;              /* Needle pivot color.                                    */
              1;                /* Scale radius.                                          */
              0;                /* Scale background extend: [atStart atEnd]               */
              1.12              /* Scale background outerRadius.                          */
              1.08 1.08 0.70 0.79  /* Scale background section bounds 1.                  */
              1.08 1.08 0.81 0.99; /* Scale background section bounds 2.                  */
              0;                   /* Scale background shading. 0...solid, -1...shaded    */
              orange;red)          /* Scale background colors.                            */
  /* Use style functions to control the appearance of the needle. */
  FillStyle(1;red)
  BorderStyle(1;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;10;5;4)
  AxisLine(all;0)
  AxisMajorTicks(all;12;1.5;white;;out)
  AxisMinorTicks(all;0.5;2 2 2;white;;out)
  AxisMajorTickLabelStyle(all;Arial;12;plain;#fff)
  AxisMajorTickLabelOptions(all;in)
  AddText(100;158;"6.79";Verdana;12;bold;#888;center)
  ChartBackground(xy;#555;shaded;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(0;10)
CloseDrawing()
```



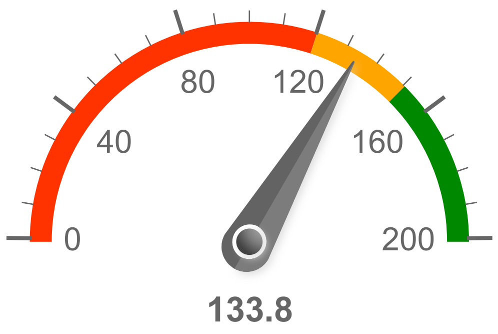
```
OpenDrawing(200;120)
  OpenChart(0;0;200;200)
    ChartData(0.65)
    RadialGauge(shadow;
               -60;             /* Start angle in [deg].                                  */
                120;            /* Arc angle in [deg]. If negative, go counter-clockwise. */
                0;              /* Background as full circle.                             */
                1.4 0.0 0.07;   /* Needle length: [head tail offsetFromCenter]            */
                0.01;           /* Needle thickness: [atCenter atHead colorSplitFactor]   */
                1;              /* Needle head shape. 1...peak, 2...round, 3...arrow      */
                1;              /* Needle tail shape. 1...peak, 2...round                 */
                0.15 0.10;      /* Needle pivot diameters: [out in]                       */
                red;            /* Needle pivot color.                                    */
                1;              /* Scale radius.                                          */
                0.01;           /* Scale background extend: [atStart atEnd]               */
                0.92            /* Scale background outerRadius.                          */
                0.82 0.82 0.00 0.16  /* Scale background section bounds 1.                */
                0.82 0.82 0.84 1.00; /* Scale background section bounds 2.                */
                0;                   /* Scale background shading. 0...solid, -1...shaded  */
                red)                 /* Scale background color.                           */
    /* Use style functions to control the appearance of the needle. */
    FillStyle(1;red)
    BorderStyle(all;none)
    ShadowStyle(all;1 1 3;lightGray)
    /* Use scale and axis functions to control the appearance of the scale. */
    Scaling(all;linear;-1;1;4;5)
    AxisLine(all;0)
    AxisMajorTicks(all;10;1;#666;;out)
    AxisMinorTicks(all;0.5;2 2 2;#666;;out)
    AxisMajorTickLabelTexts(all;"";"-0.5";"0";"0.5";"")
    AxisMajorTickLabelStyle(all;Arial;9;plain;#666)
    ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
    ChartBackgroundOptions(;8)
  CloseChart()
  // Background() /* Uncomment while positioning the gauge. */
CloseDrawing()
```



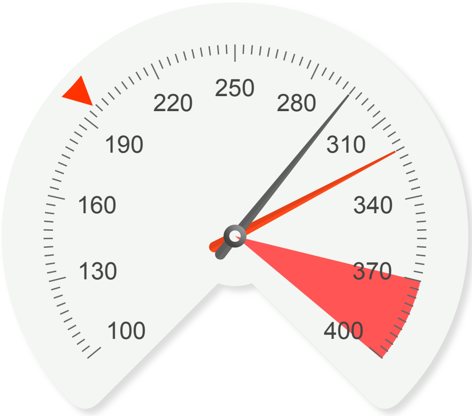
```
OpenDrawing(200;200)
  ChartData(78.5)
  RadialGauge(default;
             -135;              /* Start angle in [deg].                              */
              225;              /* Arc angle in [deg]. If negative, go counter-clockwise.  */
              0;                /* Background as full circle.                         */
              1.05 0.20 0.12;   /* Needle length: [head tail offsetFromCenter]        */
              0.30 0.04 0.10;   /* Needle thickness: [atCenter atHead colorSplitFactor]    */
              2;                /* Needle head shape. 1...peak, 2...round, 3...arrow       */
              2;                /* Needle tail shape. 1...peak, 2...round             */
              0.19;             /* Needle pivot diameters: [out in]                   */
              #499BC9;          /* Needle pivot color.                                */
              1;                /* Scale radius.                                      */
              0;                /* Scale background extend: [atStart atEnd]           */
              0.9               /* Scale background outerRadius.                      */
              0.900 0.515 0.00 0.91   /* Scale background section bounds 1.           */
              0.515 0.510 0.92 0.94   /* Scale background section bounds 2.           */
              0.510 0.505 0.95 0.97   /* Scale background section bounds 3.           */
              0.505 0.500 0.98 1.00;  /* Scale background section bounds 4.           */
              0;                      /* Scale background shading. 0...solid, -1...shaded  */
              #499BC9)                /* Scale background color.                      */
  /* Use style functions to control the appearance of the needle. */
  FillStyle(1;#499BC9)
  BorderStyle(1;;1.5;white)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;100;1)
  AxisOptions(all;none)
  AddText(110;140;"<b>78.5</b><span size=10> MB/s</span>";Verdana;16;plain;#666;center)
CloseDrawing()
```



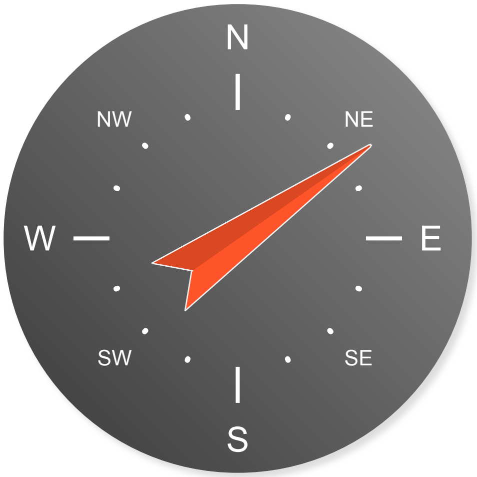
```
OpenDrawing(200;140)
  OpenChart(0;0;200;200)
    ChartData(133.8)
    RadialGauge(shadow;
               -90;             /* Start angle in [deg].                                  */
                180;            /* Arc angle in [deg]. If negative, go counter-clockwise. */
                0;              /* Background as full circle.                             */
                0.95 0.14 0.08; /* Needle length: [head tail offsetFromCenter]            */
                0.20 0.01 0.20; /* Needle thickness: [atCenter atHead colorSplitFactor]   */
                1 .5;           /* Needle head shape. 1...peak, 2...round, 3...arrow      */
                2 1;            /* Needle tail shape. 1...peak, 2...round                 */
                0.12;           /* Needle pivot diameters: [out in]                       */
                #555;           /* Needle pivot color.                                    */
                1;              /* Scale radius.                                          */
                0.005;          /* Scale background extend: [atStart atEnd]               */
                1               /* Scale background outerRadius.                          */
                0.90 0.90 0.00 0.60  /* Scale background section bounds 1.  (red)         */
                0.90 0.90 0.60 0.75  /* Scale background section bounds 2.  (orange)      */
                0.90 0.90 0.75 1.00; /* Scale background section bounds 2.  (green)       */
                0;                   /* Scale background shading. 0...solid, -1...shaded  */
                red;orange;green)    /* Scale background colors.                          */
    /* Use style functions to control the appearance of the needle. */
    FillStyle(1;#555)
    BorderStyle(all;none)
    ShadowStyle(all;1 1 3;lightGray)
    /* Use scale and axis functions to control the appearance of the scale. */
    Scaling(all;linear;0;200;5;4)
    AxisLine(all;0) /* Hide axis line. */
    AxisMajorTicks(all;8;1.5;#666;;out)
    AxisMinorTicks(all;4;0.5;#666;;out)
    AxisMajorTickLabelStyle(all;Arial;12;plain;#666)
    AxisMajorTickLabelOptions(all;in;-6)
    AddText(100;130;"133.8";Arial;13;bold;#666;center)
  CloseChart()
  // Background() /* Uncomment while positioning the gauge. */
CloseDrawing()
```



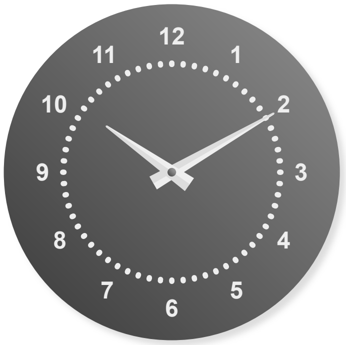
```
OpenDrawing(200;175)
  OpenChart(0;0;200;200)
    ChartData(295;322;195)
    RadialGauge(default;
             -135;             /* Start angle in [deg].                                   */
              270;             /* Arc angle in [deg]. If negative, go counter-clockwise.  */
                0;             /* Background as full circle.                              */
                0.95 0.15 0.05 /* Black needle length: [head tail offsetFromCenter]       */
                0.95 0.15 0.05 /* Red needle length: [head tail offsetFromCenter]         */
              1.16 0.00 1.01;/* Red marker length: [head tail offsetFromCenter]           */
              0.05 0.01 0.1  /* Black needle thickness: [atCenter atHead colorSplitFactor]*/
              0.05 0.01 0.1  /* Red needle thickness: [atCenter atHead colorSplitFactor]  */
              0.15 0.15 0.0; /* Red marker thickness: [atCenter atHead colorSplitFactor]  */
              1 0            /* Red needle head shape. 1...peak, 2...round, 3...arrow     */
              1 0            /* Black needle head shape. 1...peak, 2...round,3...arrow    */
              1 1;           /* Red marker head shape. 1...peak, 2...round, 3...arrow     */
              2 1;           /* Needle tail shape. 1...peak, 2...round                    */
              0.12 0.06;     /* Needle pivot diameters: [out in]                          */
              #555;          /* Needle pivot color.                                       */
              0.9;           /* Scale radius.                                             */
              0.02;          /* Scale background extend: [atStart atEnd]                  */
              1              /* Scale background outerRadius.                             */
              0.0 0.0 0.884 0.981; /* Scale background section.                           */
              0;                   /* Scale background shading. 0...solid, -1...shaded    */
              #f55)                /* Scale background color.                             */
    /* Use style functions to control the appearance of the needles and marker. */
    FillStyle(1;#555)
    FillStyle(2;red)
    FillStyle(3;red)
    BorderStyle(all;none)
    /* Use scale and axis functions to control the appearance of the scale. */
    Scaling(all;linear;100;400;10;10)
    AxisLine(all;0) /* Hide axis line. */
    AxisMajorTicks(all;6;0.5;#666;;in)
    AxisMinorTicks(all;3;0.5;#666;;in)
    AxisMajorTickLabelStyle(all;Arial;9;plain;#444)
    AxisMajorTickLabelOptions(all;in;-0)
    ChartBackground(xy;#f3f6f2;0;0;;;2 2 2;lightGray)
    ChartBackgroundOptions(;8)
  CloseChart()
  // Background() /* Uncomment while positioning the gauge. */
CloseDrawing()
```



```
OpenDrawing(200;200)
  ChartData(55)
  RadialGauge(default;
              0;               /* Start angle in [deg].                          */
              360;             /* Arc angle in [deg]. If negative, go counter-clockwise. */
              1;               /* Background as full circle.                     */
              1.25 0.65 0;     /* Needle length: [head tail offsetFromCenter]    */
              0.30 0.02 0.15;  /* Needle thickness: [atCenter atHead colorSplitFactor]   */
              1  0.5;          /* Needle head shape. 1...peak, 2...round, 3...arrow      */
              1 -0.5;          /* Needle tail shape. 1...peak, 2...round          */
              0.0;             /* Needle pivot diameter.                         */
              none;            /* Needle pivot color.                            */
              0.9)             /* Scale radius.                                  */
  /* Use style functions to control the appearance of the needle. */
  FillStyle(1;red)
  BorderStyle(1;;0.5;#eee)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;360;4;4)
  AxisLine(all;0)
  AxisMajorTicks(all;12;1.5;white;;out)
  AxisMinorTicks(all;0.5;2 2 2;white;;out)
  AxisMajorTickLabelTexts(all;"N";"E";"S";"W")
  AxisMinorTickLabelTexts(all;"";"NE";"";"";"SE";"";"";"SW";"";"";"NW";"")
  AxisMajorTickLabelStyle(all;Arial;13;plain;white)
  AxisMinorTickLabelStyle(all;Arial;8;plain;white)
  AxisMajorTickLabelOptions(all;out)
  AxisMinorTickLabelOptions(all;out;5)
  ChartBackground(xy;#555;shaded;0;;;2 2 2;lightGray)
CloseDrawing()
```



```
OpenDrawing(150;150)
  ChartData(10.17; /* = mod(hour,12)+minute/60 */
            2)     /* = minute/5               */
  RadialGauge(default;
              0;             /* Start angle in [deg].                                  */
              360;           /* Arc angle in [deg]. If negative, go counter-clockwise. */
              1;             /* Background as full circle.                             */
              0.75 0.2 0     /* Needle length 1: [head tail offsetFromCenter]          */
              1.10 0.2 0;    /* Needle length 2: [head tail offsetFromCenter]          */
              0.12 0.02 0.1; /* Needle thickness: [atCenter atHead colorSplitFactor]   */
              1;             /* Needle head shape. 1...peak, 2...round, 3...arrow      */
              1;             /* Needle tail shape. 1...peak, 2...round                 */
              0.08;          /* Needle pivot diameter.                                 */
              #666;          /* Needle pivot color.                                    */
              0.9)           /* Scale radius.                                          */
  /* Use style functions to control the appearance of the needles. */
  FillStyle(1;#eee)
  FillStyle(2;#eee)
  BorderStyle(all;none)
  /* Use scale and axis functions to control the appearance of the scale. */
  Scaling(all;linear;0;12;12;5)
  AxisLine(all;0) /* Hide axis line. */
  AxisMajorTicks(all;0.5;2 2 2;#eee;;out)
  AxisMinorTicks(all;0.5;2 2 2;#eee;;out)
  AxisMajorTickLabelTexts(all;"12";"1";"2";"3";"4";"5";"6";"7";"8";"9";"10";"11")
  AxisMajorTickLabelStyle(all;Arial;9;bold;#eee)
  ChartBackground(xy;#555;shaded;0;;;2 2 2;lightGray)
  ChartBackgroundOptions(0;3)
CloseDrawing()
```



Contour Plots

Two functions are available to set up contour plots: `ContourPlot()` and `ContourPlotOptions()`. The appearance of the contour lines can be controlled by the `LineStyle()` and `LineColorScheme()` functions, the labels by using the five style functions `LabelTexts()`, `LabelStyle()`, `LabelBackground()`, `LabelBackgroundOptions()` and `LabelOptions()`. By using the function `BorderStyle()` the appearance of the border around the contour plot can be controlled.

```
ContourPlot(appearance;numOfGridPointsHorizontal;numOfGridPointsVertical;
          arrangeColumnByColumn;margins)
```

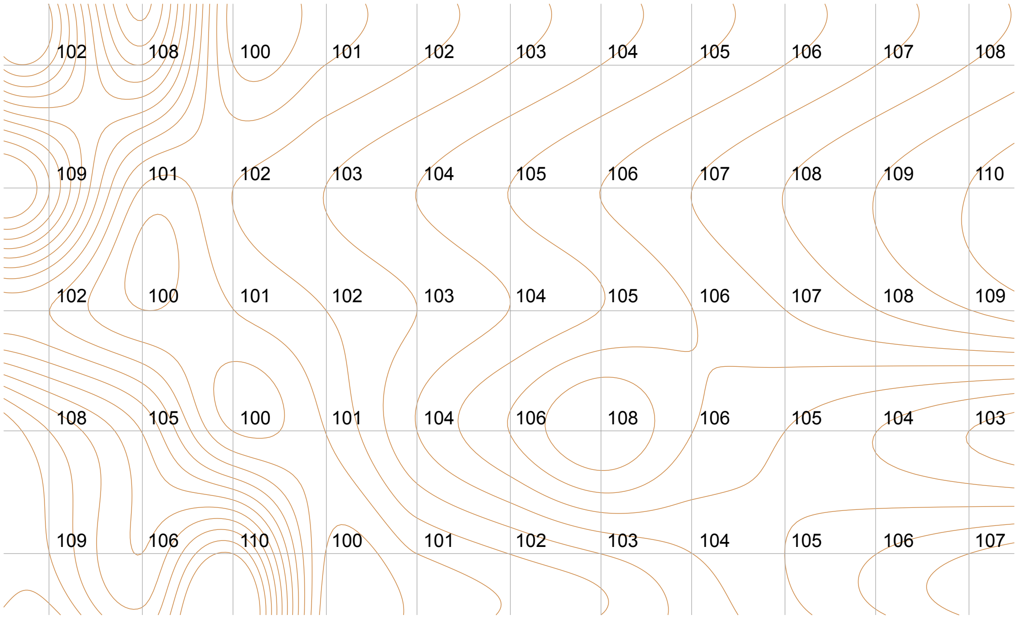
Arguments	*Type*	*Range*	*Default*	*Notes*
appearance	int	0..127	0	Constants: shadow,label,symbol
numOfGridPointsHorizontal	int	2..100000	(autom.)	
numOfGridPointsVertical	int	2..100000	(autom.)	
arrangeColumnByColumn	int	0..1	0	
margins	num[]	0..0.5	0,5	

```
ContourPlotOptions(contourValues;contourSmoothingFactor;meshStroke;meshColor)
```

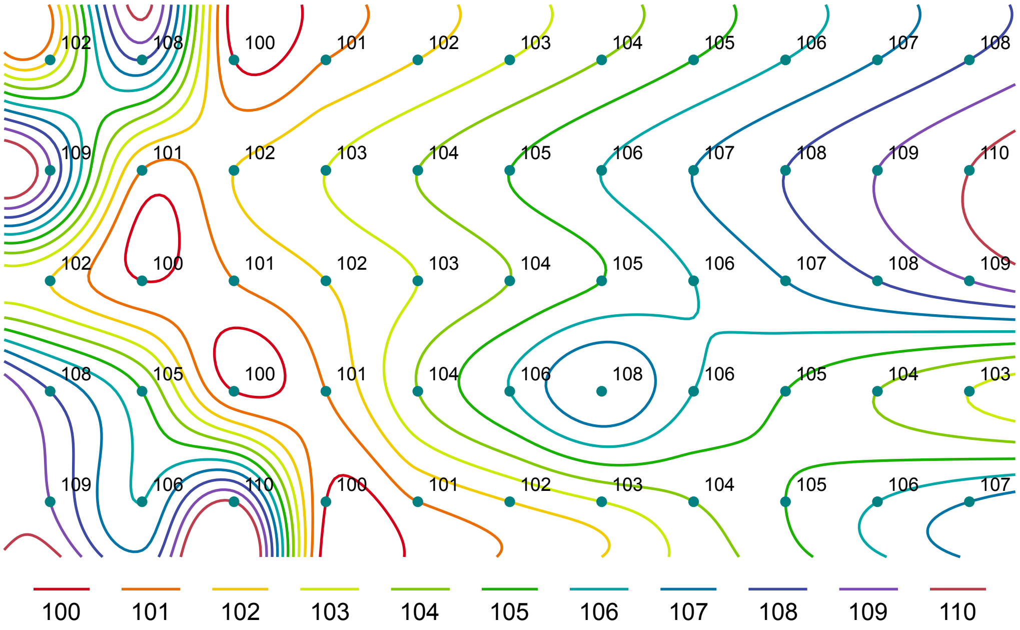
Arguments	*Type*	*Range*	*Default*	*Notes*
contourValues	num[]	-inf..+inf	(autom.)	
contourSmoothingFactor	num	0..1	0,5	
meshStroke	num[]	0..1000	1	
meshColor	rgba	0..255	#aaaf	

Examples:

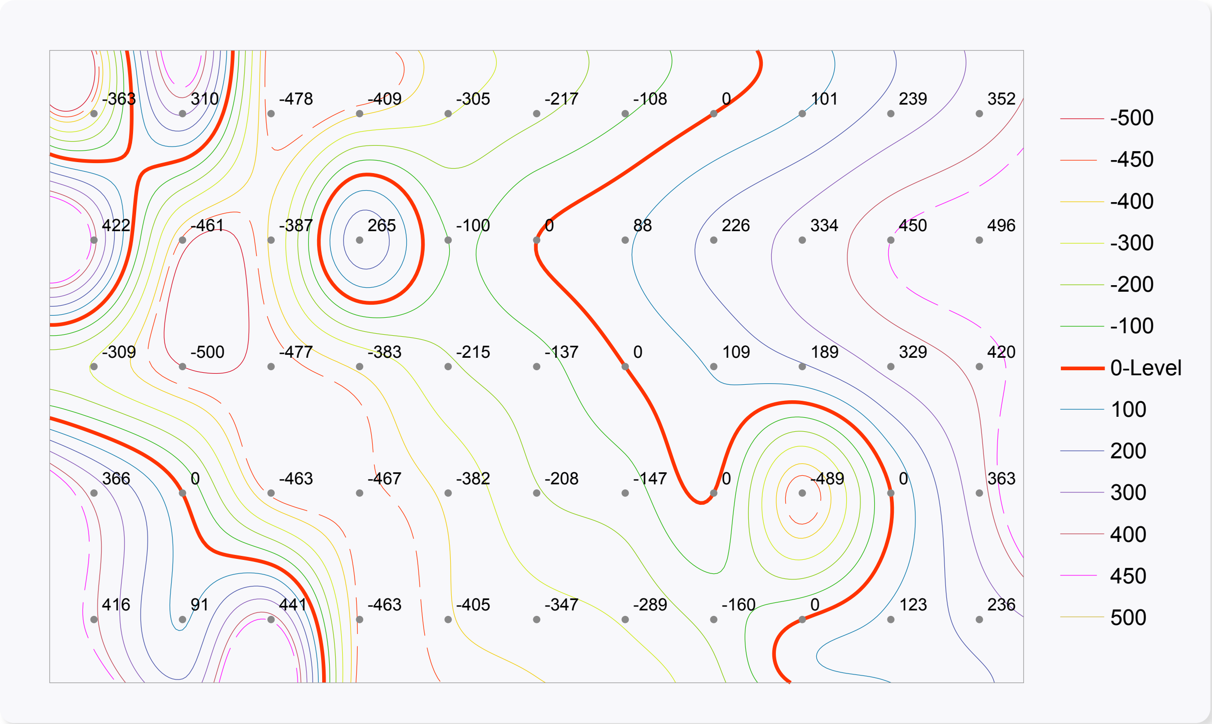
```
OpenDrawing(400;250)
  ChartData(102 108 100 101 102 103 104 105 106 107 108
            109 101 102 103 104 105 106 107 108 109 110
            102 100 101 102 103 104 105 106 107 108 109
            108 105 100 101 104 106 108 106 105 104 103
            109 106 110 100 101 102 103 104 105 106 107)
  ContourPlot(label;11;5)
  ContourPlotOptions(;;0.25)
  Borderstyle(1;none)
  LineStyle(all;;0.25;peru)
  LabelStyle(1;Arial;7)
CloseDrawing()
```



```
OpenDrawing(400;250)
  ChartData(102 108 100 101 102 103 104 105 106 107 108
            109 101 102 103 104 105 106 107 108 109 110
            102 100 101 102 103 104 105 106 107 108 109
            108 105 100 101 104 106 108 106 105 104 103
            109 106 110 100 101 102 103 104 105 106 107)
  ContourPlot(label+symbol;11;5)
  ContourPlotOptions(;;0) /* Hide grid. */
  SymbolStyle(1;bullet;3;1;teal)
  Borderstyle(1;none)
  LineColorScheme(15)
  LabelStyle(1;Arial;7)
  LegendStyle(Arial;9)
  LegendOptions(bottomCenter;
                 0;  /* Place legend inside of plot area. */
                 0;  /* Horizontal offset.                */
                 3;  /* Vertical offset.                  */
                 1;  /* Number of rows.                   */
                 0;  /* Legend item type (ignored).       */
                12;  /* Legend item width.                */
                 1;  /* Legend item height.               */
                 3;  /* Gap between text and line.        */
                 0;  /* Row gap (ignored).                */
                12;  /* Column gap.                       */
     bottomCenter)  /* Text location.                    */
CloseDrawing()
```



```
OpenDrawing(500;300)
  OpenChart(20;20;400;260;on)
    ChartData(-363  310 -478 -409 -305 -217 -108    0  101 239 352
               422 -461 -387  265 -100    0   88  226  334 450 496
              -309 -500 -477 -383 -215 -137    0  109  189 329 420
               366    0 -463 -467 -382 -208 -147    0 -489   0 363
               416   91  441 -463 -405 -347 -289 -160    0 123 236)
    ContourPlot(label+symbol;11;5;0;)
    ContourPlotOptions(-500 -450 -400 -300 -200 -100 0 100 200 300 400 450 500;;0)
    Borderstyle(1;1;0.25;#999)
    SymbolStyle(1;bullet;2;1;#888)
    LabelStyle(1;Arial;7)
    LineColorscheme(15)
    LineStyle(all;;0.25)
    LineStyle( 2;;0.25 15 5)          /* Contour line: -450 */
    LineStyle(12;;0.25 15 5;magenta)  /* Contour line:  450 */
    LineStyle( 7;;1.5;red)            /* Contour line:    0 */
    LegendTexts(;;;;;;"0-Level")
    LegendStyle(Arial;9)
    LegendOptions(centerRight;
              0;  /* Place legend inside of plot area. */
              6;  /* Horizontal offset.                */
              0;  /* Vertical offset.                  */
           -1 1;  /* Unlimited rows, 1 column.         */
              0;  /* Legend item type (ignored).       */
             10;  /* Legend item width.                */
             10;  /* Legend item height.               */
              3;  /* Gap between text and line.        */
              7;  /* Row gap.                          */
              0;  /* Column gap.                       */
      centerRight)  /* Text location.                  */
  CloseChart()
  Background(#f8f8fb;0;0;;;2 2 2;lightGray)
  BackgroundOptions(6)
CloseDrawing()
```



Density Plots

Two functions are available to set up density plots: `DensityPlot()` and `DensityPlotOptions()`. The appearance of the density can be controlled by the `FillColorScale()` functions, the labels by using the five style functions `LabelTexts()`, `LabelStyle()`, `LabelBackground()`, `LabelBackgroundOptions()` and `LabelOptions()`. By using the function `BorderStyle()` the appearance of the border around the density plot can be controlled.

```
DensityPlot(appearance;numOfGridPointsHorizontal;numOfGridPointsVertical;
            arrangeColumnByColumn;margins)
```

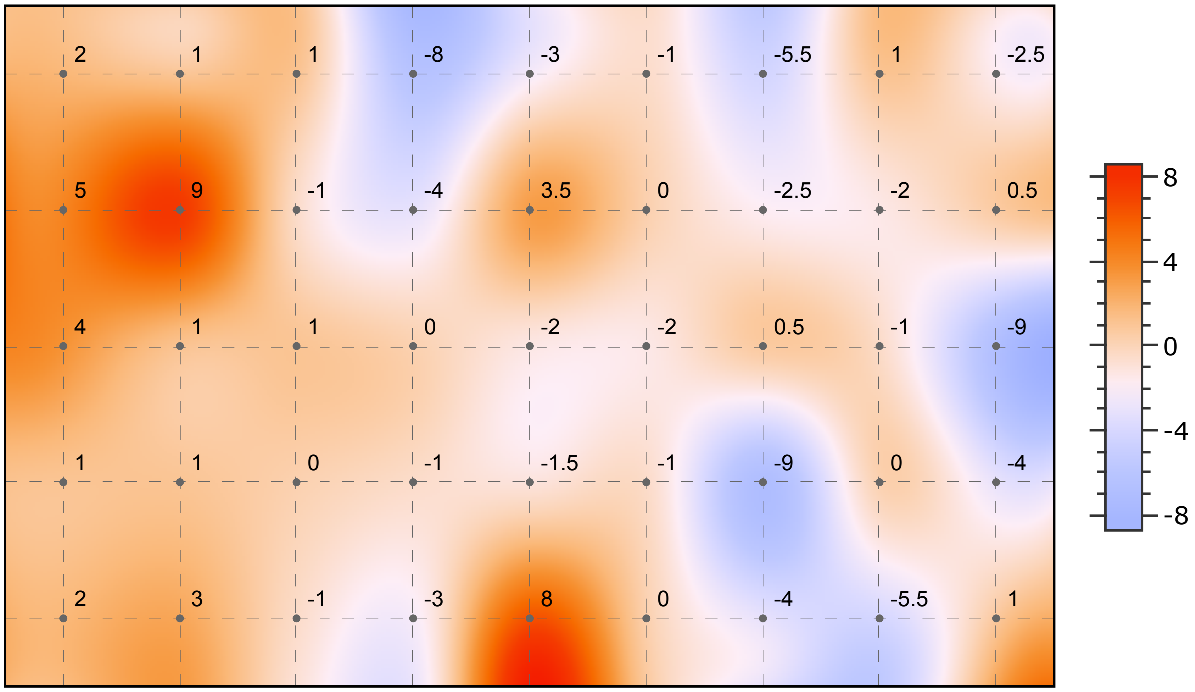
Arguments	*Type*	*Range*	*Default*	*Notes*
appearance	int	0..127	0	Constants: shadow,label,symbol
numOfGridPointsHorizontal	int	2..100000	(autom.)	
numOfGridPointsVertical	int	2..100000	(autom.)	
arrangeColumnByColumn	int	0..1	0	
margins	num[]	0..0.5	0,5	

```
DensityPlotOptions(contourValues;contourSmoothingFactor;meshStroke;meshColor)
```

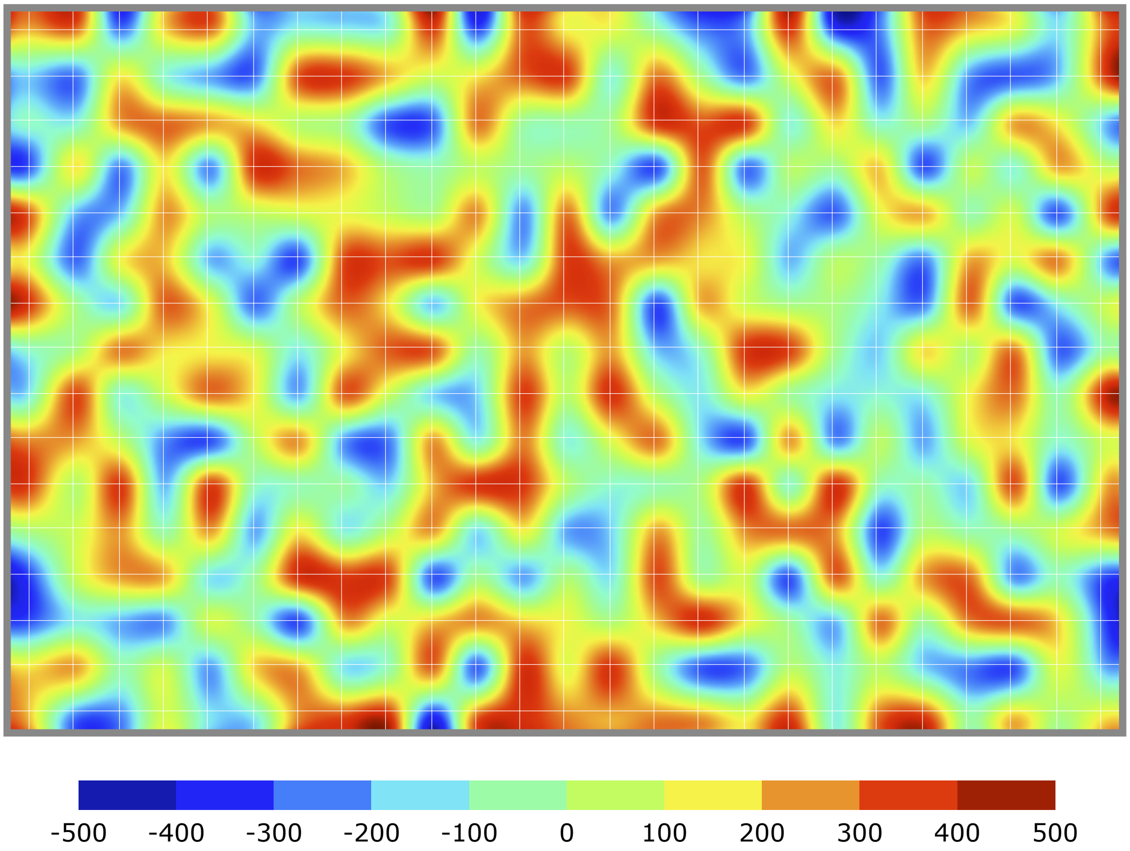
Arguments	*Type*	*Range*	*Default*	*Notes*
meshStroke	num[]	0..1000	1	
meshColor	rgba	0..255	#aaaf	

Examples:

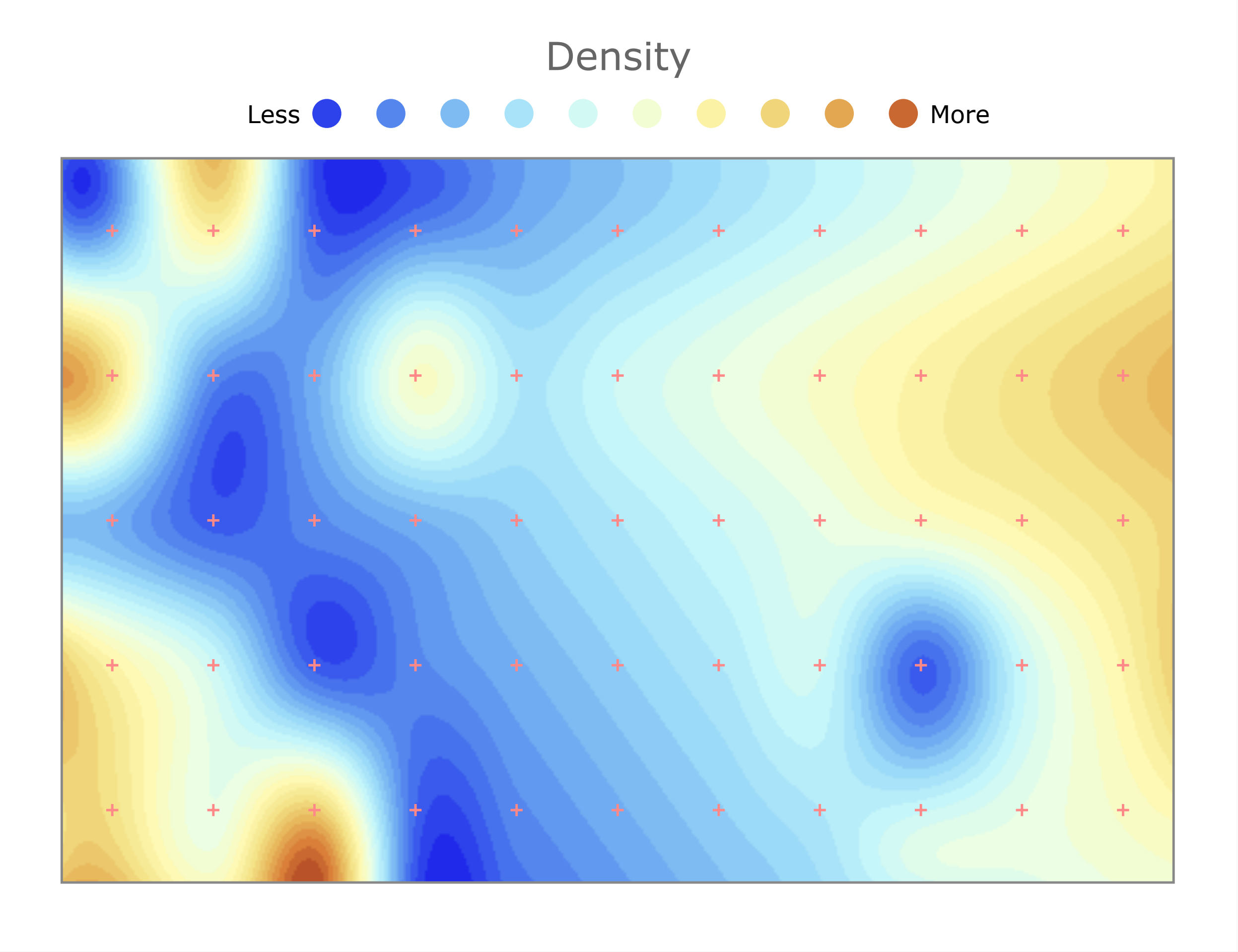
```
OpenDrawing(500;300)
  ChartData(2  5  4  1    2   1    9  1 1
            3  1 -1  1    0  -1   -8 -4 0
           -1 -3 -3  3.5 -2  -1.5  8 -1 0
           -2 -1  0 -5.5 -2.5 0.5 -9 -4 1
           -2 -1  0 -5.5 -2.5 0.5 -9 -4 1)
  DensityPlot(symbol+label;9;5;1;0.5)
  DensityPlotOptions(0.25 5 5;#666)
  FillColorScale(1;-14)
  SymbolStyle(1;bullet;3;0.25;#666)
  LabelStyle(1;Arial;9)
  LegendStyle(Verdana;10)
  LegendOptions(centerRight;
                0;  /* Place legend inside of plot area. */
                6;  /* Horizontal offset.                */
                0;  /* Vertical offset.                  */
                1;  /* Number of rows (ignored).         */
                0;  /* Legend item type (ignored).       */
               15;  /* Color scale width.                */
              150;  /* Color scale height.               */
                3;  /* Gap between text and ticks.       */
                0;  /* Row gap (ignored).                */
                0;  /* Column gap (ignored).             */
                6)  /* Text location.                    */
  LegendColorScale(-8 -4 0 4 8;  /* Scale values.     */
                   1;            /* Do reverse scale. */
                   5;            /* Scale inset.      */
                   6;            /* Tick length.      */
                   0;            /* Tick offset.      */
                   1;            /* Tick stroke.      */
                   #333;         /* Tick color.       */
                   1;            /* Do mirror ticks.  */
                   3;            /* Num of sub-ticks. */
                   3;            /* Sub-tick length.  */
                   "|u|";        /* Label format.     */
                   1;            /* Border stroke.    */
                   #333)         /* Border color.     */
CloseDrawing()
```


2
1
1
-8
-3
-1
-5.5
1
-2.5
5
9
-1
-4
3.5
0
-2.5
-2
0.5
4
1
1
0
-2
-2
0.5
-1
-9
1
1
0
-1
-1.5
-1
-9
0
-4
2
3
-1
-3
8
0
-4
-5.5
1
8
4
0
-4
-8

```
OpenDrawing(500;380)
  OpenChart(20;20;460;300;on)
    ChartData(236 310 -391 181 318 -338 -176 -194 -169 429 -455 401 174 98 -114 -405 -372
441 -494 -423 361 187 54 -269 330 -311 -428 172 -200 -359 -436 415 472 238 50 180 389 445
-209 358 -70 -410 72 352 -464 216 -395 -473 -314 391 -192 -212 289 379 280 193 -10 -85 -492
-461 349 -110 -134 -73 480 422 459 -213 198 -203 -67 -306 278 160 -285 -434 174 -411 176
-392 487 370 249 -65 -147 16 -109 1 -201 -481 399 -430 -16 -87 222 -484 20 -210 272 46 345
-342 -312 306 -36 12 76 132 33 -32 314 -390 354 -393 302 331 -24 -222 -460 110 262 -138 59
-469 317 83 -450 134 241 -335 -225 -491 439 419 467 74 -275 459 288 276 187 126 -332 -15
-136 -414 275 91 329 -307 432 -81 -261 390 117 -440 -16 388 177 -310 134 338 415 358 -493
260 66 -59 -60 -252 -423 375 -460 -239 20 -184 -36 344 180 157 83 -244 158 401 399 -120 293
0 292 -321 -170 471 444 -83 -296 189 -12 345 -456 -132 -205 434 -203 44 361 158 -344 396 40
-289 -295 459 -1 489 -11 -253 133 -88 -260 -212 -235 154 365 -149 496 331 286 25 -395 -484 5
301 -382 -418 297 -224 339 -218 98 354 -272 -479 292 -394 0 -356 95 180 -203 49 427 -7 463
-332 447 -168 -155 -128 -286 257 484 471 -7 -198 -147 -33 467 -181 491 -156 -133 -270 415
-476 207 -43 61 306 -136 289 -352 168 -228 9 293 -273 150 -359 -292 292 -97 317 367 297 -500
-34 -144 -100 66 283 -488 37 289 261 -266 -104 486 475 423 -473 -80 -359 -43 -255 429 -80 56
-480 387 -191 294 365 -385 -162 -483 -473 -243 -341 -385 63 -127 -493 296 77 234 317 220 134
-31 247 495 164 -101 -321 359 -96 345 400 207 -462 191 285 -142 45 -368 211 273 -267 -142
372 -458 494 113 478 -100 -475 -438 -8 -234 1 -293 -424 -485 99 -316 194 -375 -386 66 -277
-204 319 379 445 -495 313 497 310 286 297 228 84 413 -233 347 402 -149 177 -34 114)
    DensityPlot(default;25;16)
    DensityPlotOptions(0.25;white)
    Borderstyle(1;;3;#888)
    FillColorScale(1;25)
    /* Set up legend. */
    LegendStyle(Verdana;10)
    LegendOptions(bottomCenter;
                  0;  /* Place legend inside of plot area. */
                  0;  /* Horizontal offset.                */
                 10;  /* Vertical offset.                  */
                  1;  /* Number of rows (ignored).         */
                  0;  /* Legend item type (ignored).       */
                400;  /* Color scale width.                */
                 12;  /* Color scale height.               */
                  3;  /* Gap between text and ticks.       */
                  0;  /* Row gap (ignored).                */
                  0;  /* Column gap (ignored).             */
       bottomCenter)  /* Text location.                    */
    LegendColorTones(-500;     /* Scale value Min.         */
                      500;     /* Scale value Max.         */
                       10;     /* Number of color tones.   */
                        0;     /* Do reverse scale.        */
                        0)     /* Gap between color tones. */
  CloseChart()
CloseDrawing()
```


-500
-400
-300
-200
-100
0
100
200
300
400
500

```
OpenDrawing(500;380)
  OpenChart(20;60;460;300;on)
    ChartData(-300  300 -500 -400 -300 -200 -100    0  100 200 300
               400 -400 -300  200 -100    0  100 200  300 400 500
              -300 -500 -400 -300 -200 -100    0 100  200 300 400
               300    0 -500 -400 -300 -200 -100   0 -500   0 300
               400  100  500 -500 -400 -300 -200 -100   0 100 200)
    DensityPlot(symbol;11;5)
    DensityPlotOptions(0) /* Hide grid. */
    Borderstyle(1;;1;#888)
    SymbolStyle(1;plus;5;1;#f88)
    FillColorScale(1;11;;;30)
    /* Set up legend. */
    LegendTitle("Density";Verdana;16;plain;#666;;8)
    LegendTexts("Less";;;;;;;;;"More")
    LegendStyle(Verdana;10)
    LegendOptions(topCenter;
                  0;  /* Place legend inside of plot area. */
                  0;  /* Horizontal offset.                */
                 -3;  /* Vertical offset.                  */
                  1;  /* Number of rows (ignored).         */
                  0;  /* Legend item type (ignored).       */
                250;  /* Color scale width.                */
                 12;  /* Color scale height.               */
                  5;  /* Gap between text and ticks.       */
                  0;  /* Row gap (ignored).                */
                  0;  /* Column gap (ignored).             */
         centerLeft;  /* Text location.                    */
                  2)  /* Shape: 1...rect, 2..oval.         */
    LegendColorTones(-500;     /* Scale value Min.         */
                      500;     /* Scale value Max.         */
                       10;     /* Number of color tones.   */
                        0;     /* Do reverse scale.        */
                        1)     /* Gap between color tones. */
  CloseChart()
CloseDrawing()
```


Density
Less
More

Venn Diagrams

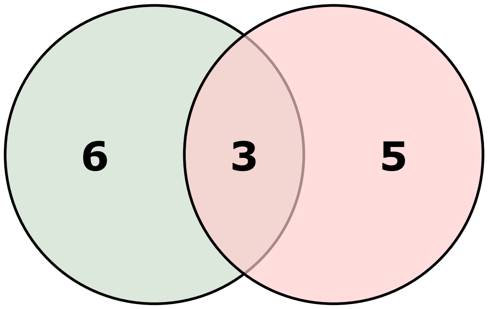
Currently, only 1, 2 and 3 area diagrams are supported. The appearance of the areas can be controlled by the `FillStyle()` and `BorderStyle()` functions, the labels by using the five style functions `LabelTexts()`, `LabelStyle()`, `LabelBackground()`, `LabelBackgroundOptions()` and `LabelOptions()`.

VennDiagram(appearance;overlapFactor)

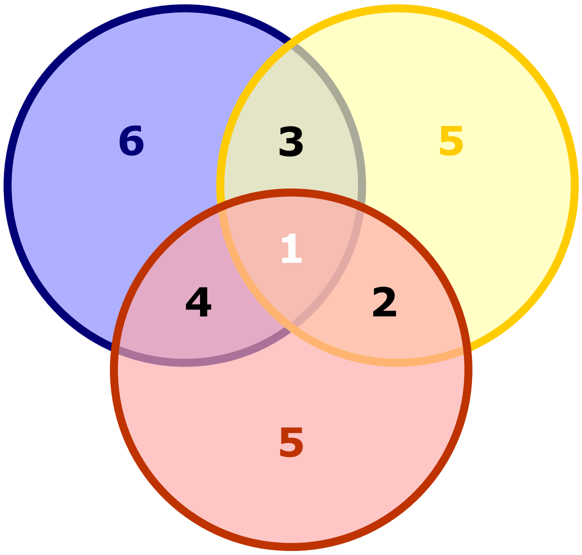
Arguments	*Type*	*Range*	*Default*	*Notes*
appearance	int	0..127	0	*Constants:* label,horizontal
overlapFactor	num	0..1	0,25	

Examples:

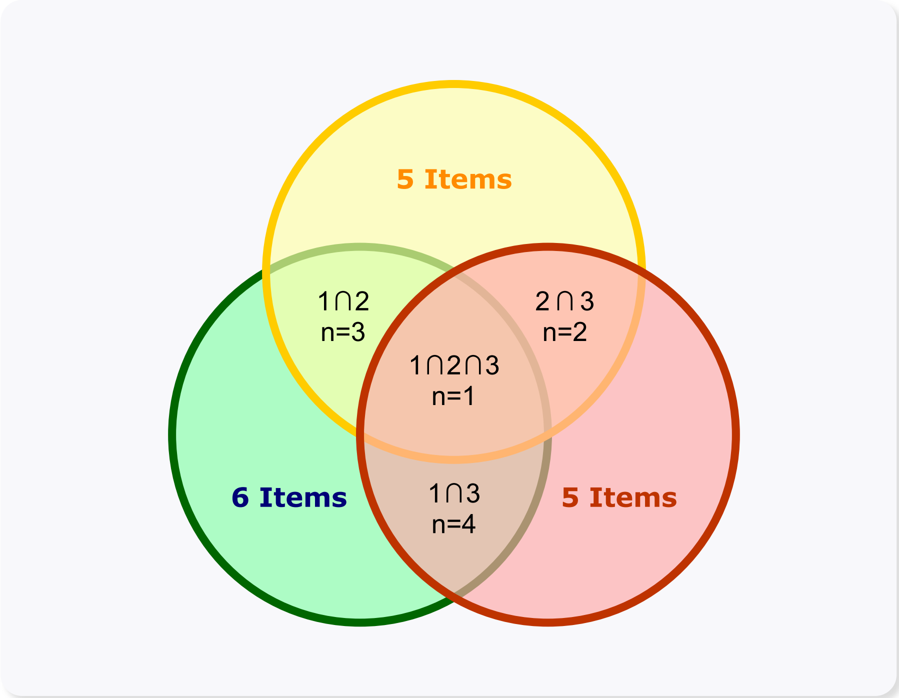
```
OpenDrawing(200;150)
  VennDiagram(label;
              0.4) /* Overlap factor. */
  ChartData(1 1 0 4 5 5; 1 1 3 0 2 )
  FillStyle(1;#cdca)
  FillStyle(2;#fcca)
  LabelStyle(all;Verdana;15;bold;black)
CloseDrawing()
```



```
OpenDrawing(300;225)
  VennDiagram(label+horizontal;
              0.4) /* Overlap factor. */
  ChartData(1 1 0 4 5 5; 1 1 3 0 2; 5 2 4 0 5)
  BorderStyle(1;;3;darkBlue)
  BorderStyle(2;;3;darkYellow)
  BorderStyle(3;;3;darkRed)
  FillStyle(1;#8888ffaa)
  FillStyle(2;#ffffaaaa)
  FillStyle(3;#ffaaaaaa)
  LabelStyle(1;Verdana;15;bold;darkBlue)
  LabelStyle(2;Verdana;15;bold;darkYellow)
  LabelStyle(3;Verdana;15;bold;darkRed)
  LabelStyle(4;Verdana;15;bold;black) /* Set 1+2   */
  LabelStyle(5;Verdana;15;bold;black) /* Set 1+3   */
  LabelStyle(6;Verdana;15;bold;black) /* Set 2+3   */
  LabelStyle(7;Verdana;15;bold;white) /* Set 1+2+3 */
CloseDrawing()
```



```
OpenDrawing(340;265)
  OpenChart(20;20;300;225)
    VennDiagram(label;
                0.5) /* Overlap factor. */
    ChartData(1 1 0 4 5 5; 1 1 3 0 2; 5 2 4 0 5)
    BorderStyle(1;;3;darkGreen)
    BorderStyle(2;;3;darkYellow)
    BorderStyle(3;;3;darkRed)
    FillStyle(1;#88ffaaaa)
    FillStyle(2;#ffffaaaa)
    FillStyle(3;#ffaaaaaa)
    LabelTexts(all;"|u| Items")
    LabelStyle(1;Verdana;10;bold;darkBlue)
    LabelStyle(2;Verdana;10;bold;darkOrange)
    LabelStyle(3;Verdana;10;bold;darkRed)
    LabelTexts(4;"1\xE2\x80\x8A∩\xE2\x80\x8A2\nn=|u|") /* Set 1+2 */
    LabelStyle(4;Arial;10;plain;black)
    LabelTexts(5;"1\xE2\x80\x8A∩\xE2\x80\x8A3\nn=|u|") /* Set 1+3 */
    LabelStyle(5;Arial;10;plain;black)
    LabelTexts(6;"2\xE2\x80\x8A∩\xE2\x80\x8A3\nn=|u|") /* Set 2+3 */
    LabelStyle(6;Arial;10;plain;black)
    LabelTexts(7;"1\xE2\x80\x8A∩\xE2\x80\x8A2\xE2\x80\x8A∩\xE2\x80\x8A3\nn=|u|") /* Set
1+2+3 */
    LabelStyle(7;Arial;10;plain;black)
  CloseChart()
  Background(#f8f8fb;0;0;;;2 2 2;lightGray)
  BackgroundOptions(8)
CloseDrawing()
```



Tables

A total of 19 functions are available for designing tables. They make it possible to:

- define tables with variable column widths and variable row heights.
- design cells with individual text styles, borders and backgrounds.
- alternate row and column backgrounds.
- add gaps between rows and columns.

Table structure:

```
OpenTable(xAnchor;yAnchor;hAlignment;vAlignment)
   ...
   Sequence of table functions.
   ...
   Background(...)        /* Optional. */
   BackgroundOptions(...) /* Optional. */
CloseTable()
```

Table functions:

```
TableCellBackground(cellIndices;fillColor;fillColorVariant)
TableCellBorderBottom(rowIndices;columnIndices;borderStroke;borderColor;borderColorVariant)
TableCellBorderLeft(rowIndices;columnIndices;borderStroke;borderColor;borderColorVariant)
TableCellBorderRect(cellIndices;borderStroke;borderColor;borderColorVariant)
TableCellBorderRight(rowIndices;columnIndices;borderStroke;borderColor;borderColorVariant)
TableCellBorderTop(rowIndices;columnIndices;borderStroke;borderColor;borderColorVariant)
TableCellPadding(cellIndices;padding)
TableCellStyle(cellIndices;fontName;textSize;textStyle;textColor;hAlignment;vAlignment)
TableColumnAlignments(horizontalAlignment1;horizontalAlignment2;...)
TableColumnAlternateBackgrounds(color1;color2;...)
TableColumnGap(columnIndices;gap)
TableColumnWidths(width1;width2;…)
TableRowAlignments(verticalAlignment1;verticalAlignment2;...)
TableRowAlternateBackgrounds(color1;color2;...)
TableRowGap(rowIndices;gap)
TableRowHeights(height1;height2;…)
TableTexts(options;texts)
```

Data entry: Row texts are separated by a line feed "\n" and the individual cell texts by a tab character “\t". For example:

```
TableTexts("";"Row1 Text 1\tRow1 Text 2\tRow1 Text 3\nRow2 Text 1\tRow2 Text 2\tRow2 Text 3")
```

Cells are referenced by the row and column index. For example:

```
TableCellBackground( 5  3;yellow) /* Cell in row 5, column 3.                  */
TableCellBackground( 5 -1;yellow) /* Cell in row 5, last column.               */
TableCellBackground(-1 -1;yellow) /* Cell in last row, last column.            */
TableCellBackground( 5 -2;yellow) /* Cell in row 5, second last column.        */
TableCellBackground(-3 -2;yellow) /* Cell in third last row, second last column. */
```

A range of cells is referenced by listing the upper-left cell of the range, followed by the lower-right cell, i.e. [*fromRow fromColumn toRow toColumn* ...]. For example:

```
TableCellBackground( 1 2  3  2;yellow) /* Range from cell(1,2) to cell(3,2).   */
TableCellBackground( 1 1  1 -1;yellow) /* Row 1.                               */
TableCellBackground( 2 1  2 -1;yellow) /* Row 2.                               */
TableCellBackground( 2 1  5 -1;yellow) /* Rows 2 to 5.                         */
TableCellBackground(-1 1 -1 -1;yellow) /* Last row.                            */
TableCellBackground(-2 1 -1 -1;yellow) /* Last 2 rows.                         */

TableCellBackground(1  1 -1  1;yellow) /* Column 1.                            */
TableCellBackground(1  2 -1  2;yellow) /* Column 2.                            */
TableCellBackground(1  2 -1  5;yellow) /* Column 2 to 5.                       */
TableCellBackground(1 -1 -1 -1;yellow) /* Last column.                         */
TableCellBackground(1 -2 -1 -1;yellow) /* Last 2 columns.                      */
```

Multiple cell ranges can be defined in one function call. For example:

```
TableCellBackground(1 1  1 -1         /* First row.    */
                   -1 1 -1 -1;yellow)  /* Last row.     */
TableCellBackground(1  1 -1  1        /* First column. */
                    1 -1 -1 -1;yellow) /* Last column.  */
```

Short form:
0…means all cells. For example:

```
TableCellBackground(0;yellow) /* All rows and columns. */
```

OpenTable(xAnchor;yAnchor;hAlignment;vAlignment)

Arguments	*Type*	*Range*	*Default*	*Notes*
xAnchor	num	-inf..+inf	(required)	
yAnchor	num	-inf..+inf	(required)	
hAlignment	int	1..3	left	
vAlignment	int	1..3	top	

TableCellBackground(cellIndices;fillColor;fillColorVariant)

Arguments	*Type*	*Range*	*Default*	*Notes*
cellIndices	int[]	-1000..1000	0	*0...all*
fillColor	rgba	0..255	white	
fillColorVariant	int	-1..128	0	

TableCellBorderBottom(rowIndices;columnIndices;borderStroke;borderColor;borderColorVariant)

Arguments	*Type*	*Range*	*Default*	*Notes*
rowIndices	int[]	-1000..1000	0	*0...all*
columnIndices	int[]	-1000..1000	0	*0...all*
borderStroke	num[]	0..1000	1	
borderColor	rgba	0..255	black	
borderColorVariant	int	-1..128	0	

TableCellBorderLeft(rowIndices;columnIndices;borderStroke;borderColor;borderColorVariant)

Arguments	*Type*	*Range*	*Default*	*Notes*
rowIndices	int[]	-1000..1000	0	*0...all*
columnIndices	int[]	-1000..1000	0	*0...all*
borderStroke	num[]	0..1000	1	
borderColor	rgba	0..255	black	
borderColorVariant	int	-1..128	0	

TableCellBorderRect(cellIndices;borderStroke;borderColor;borderColorVariant)

Arguments	*Type*	*Range*	*Default*	*Notes*
cellIndices	int[]	-1000..1000	0	*0...all*
borderStroke	num[]	0..1000	1	
borderColor	rgba	0..255	black	
borderColorVariant	int	-1..128	0	

TableCellBorderRight(rowIndices;columnIndices;borderStroke;borderColor;borderColorVariant)

Arguments	*Type*	*Range*	*Default*	*Notes*
rowIndices	int[]	-1000..1000	0	*0...all*
columnIndices	int[]	-1000..1000	0	*0...all*
borderStroke	num[]	0..1000	1	
borderColor	rgba	0..255	black	
borderColorVariant	int	-1..128	0	

TableCellBorderTop(rowIndices;columnIndices;borderStroke;borderColor;borderColorVariant)

Arguments	*Type*	*Range*	*Default*	*Notes*
rowIndices	int[]	-1000..1000	0	*0...all*
columnIndices	int[]	-1000..1000	0	*0...all*
borderStroke	num[]	0..1000	1	
borderColor	rgba	0..255	black	
borderColorVariant	int	-1..128	0	

TableCellPadding(cellIndices;padding)

Arguments	*Type*	*Range*	*Default*	*Notes*
cellIndices	int[]	-1000..1000	0	*0...all*
padding	num[]	0..1000	(autom.)	

TableCellStyle(cellIndices;fontName;textSize;textStyle;textColor;hAlignment;vAlignment)

Arguments	*Type*	*Range*	*Default*	*Notes*
cellIndices	int[]	-1000..1000	0	*0...all*
fontName	str	0..1000	Arial	
textSize	num	0..1000	9	
textStyle	int	0..7	0	
textColor	rgba	0..255	black	
hAlignment	int	0..3	0	
vAlignment	int	0..3	0	

TableColumnAlignments(hAlignment1;hAlignment2;...;hAlignment1000)

Arguments	*Type*	*Range*	*Default*	*Notes*
hAlignment1	int	1..3	(autom.)	
hAlignment2	int	1..3	(autom.)	
...	int	1..3	(autom.)	
hAlignment1000	int	1..3	(autom.)	

TableColumnAlternateBackgrounds(color1;color2;...;color1000)

Arguments	*Type*	*Range*	*Default*	*Notes*
color1	rgba	0..255	none	
color2	rgba	0..255	none	
...	rgba	0..255	none	
color1000	rgba	0..255	none	

TableColumnGap(columnIndices;gap)

Arguments	*Type*	*Range*	*Default*	*Notes*
columnIndices	int[]	-1000..1000	0	*0...all*
gap	num	-1000..1000	0	

TableColumnWidths(width1;width2;...;width1000)

Arguments	*Type*	*Range*	*Default*	*Notes*
width1	num	0..1000	(autom.)	*0...automatic*
width2	num	0..1000	(autom.)	*0...automatic*
...	num	0..1000	(autom.)	*0...automatic*
width1000	num	0..1000	(autom.)	*0...automatic*

TableTexts(options;texts)

Arguments	*Type*	*Range*	*Default*	*Notes*
options	str	0..1000	""	*Currently not used.*
texts	styt	0..100000	""	

TableRowAlignments(vAlignment1;vAlignment2;...;vAlignment1000)

Arguments	*Type*	*Range*	*Default*	*Notes*
vAlignment1	int	1..3	(autom.)	
vAlignment2	int	1..3	(autom.)	
...	int	1..3	(autom.)	
vAlignment1000	int	1..3	(autom.)	

TableRowAlternateBackgrounds(color1;color2;...;color1000)

Arguments	*Type*	*Range*	*Default*	*Notes*
color1	rgba	0..255	none	
color2	rgba	0..255	none	
...	rgba	0..255	none	
color1000	rgba	0..255	none	

TableRowGap(rowIndices;gap)

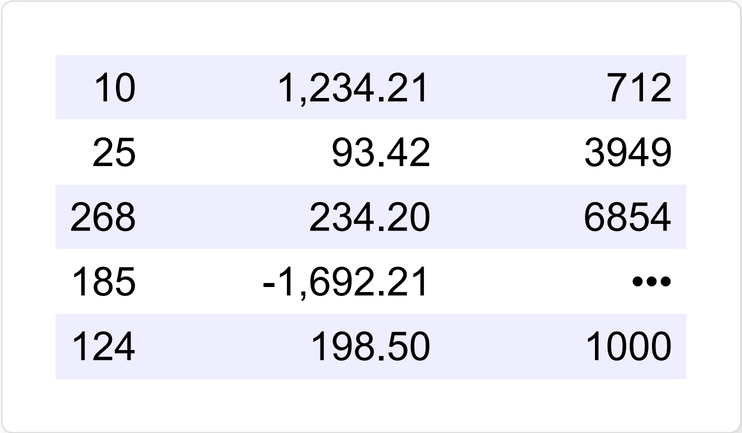
Arguments	*Type*	*Range*	*Default*	*Notes*
rowIndices	int[]	-1000..1000	0	*0...all*
gap	num	-1000..1000	0	

TableRowHeights(height1;height2;...;height1000)

Arguments	*Type*	*Range*	*Default*	*Notes*
height1	num	0..1000	(autom.)	*0...automatic*
height2	num	0..1000	(autom.)	*0...automatic*
...	num	0..1000	(autom.)	*0...automatic*
height1000	num	0..1000	(autom.)	*0...automatic*

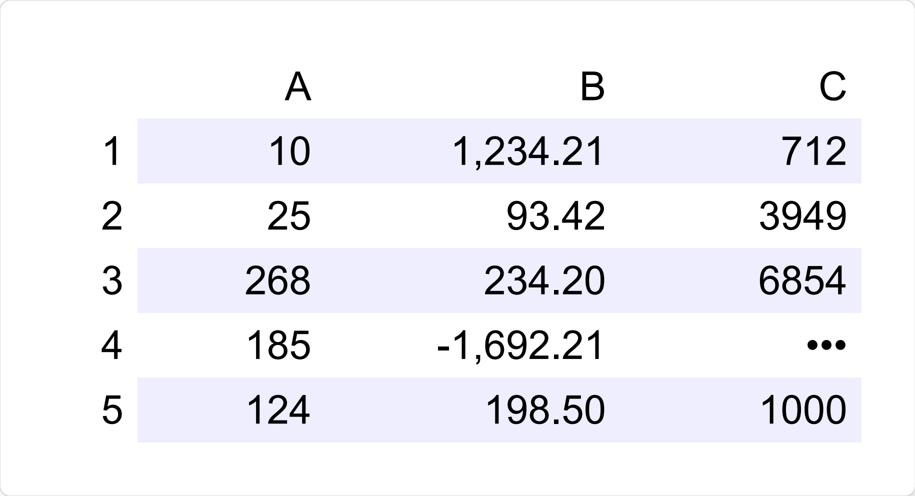
Examples:

```
OpenDrawing(400;200)
  OpenTable(0;0;left;top)

TableTexts("";"10\t1,234.21\t712\n25\t93.42\t3949\n268\t234.20\t6854\n185\t-1,692.21\t•••\n1
24\t198.50\t1000")
    TableCellStyle(0;Verdana;15;plain;black) /* 0..all cells. */
    TableColumnAlignments(right;right;right)
    TableColumnWidths(0;100;80) /* 0...automatic. */
    TableRowAlternateBackgrounds(#eef;white) /* Color of row1 row2 ... */
    Background(white;;0.5;lightGray;;2 2 5;lightGray)
    BackgroundOptions(4;20)
  CloseTable()
CloseDrawing()
```



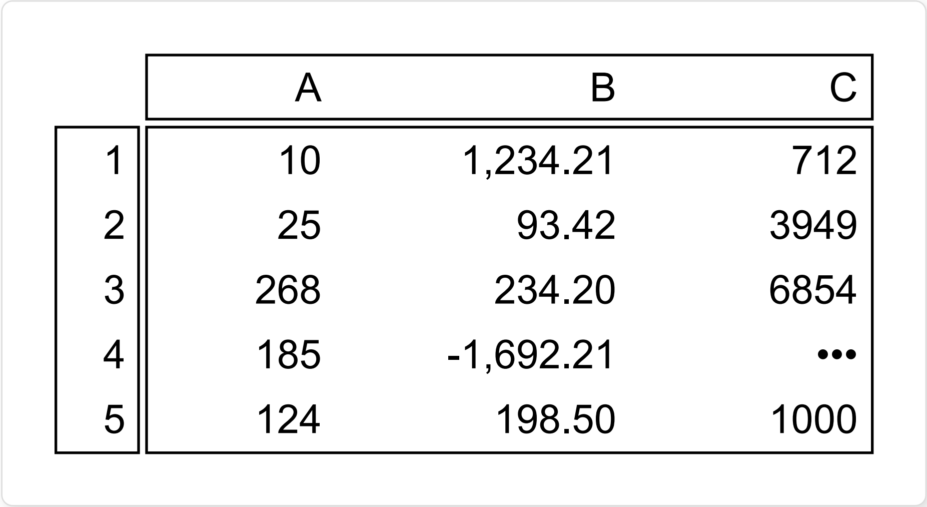
10	1,234.21	712
25	93.42	3949
268	234.20	6854
185	-1,692.21	•••
124	198.50	1000

```
OpenDrawing(400;200)
  OpenTable(0;0;left;top)
    TableTexts("";"\tA\tB\tC\r   1\t10\t1,234.21\t712\n   2\t25\t93.42\t3949\n
3\t268\t234.20\t6854\n   4\t185\t-1,692.21\t•••\n   5\t124\t198.50\t1000")
    TableCellStyle(0;Verdana;15;plain;black) /* 0..all cells. */
    TableColumnAlignments(left;right;right;right)
    TableColumnWidths(0;60;100;80) /* 0...automatic. */

    /* [fromRow fromColumn toRow toColumn] -1...lastRow or lastColumn */
    TableCellBackground(2 2 2 -1  /* Line 2. */
                        4 2 4 -1  /* Line 4. */
                        6 2 6 -1; /* Line 6. */
                        #eef)
    Background(white;;0.5;lightGray;;2 2 5;lightGray)
    BackgroundOptions(4;20)
  CloseTable()
CloseDrawing()
```



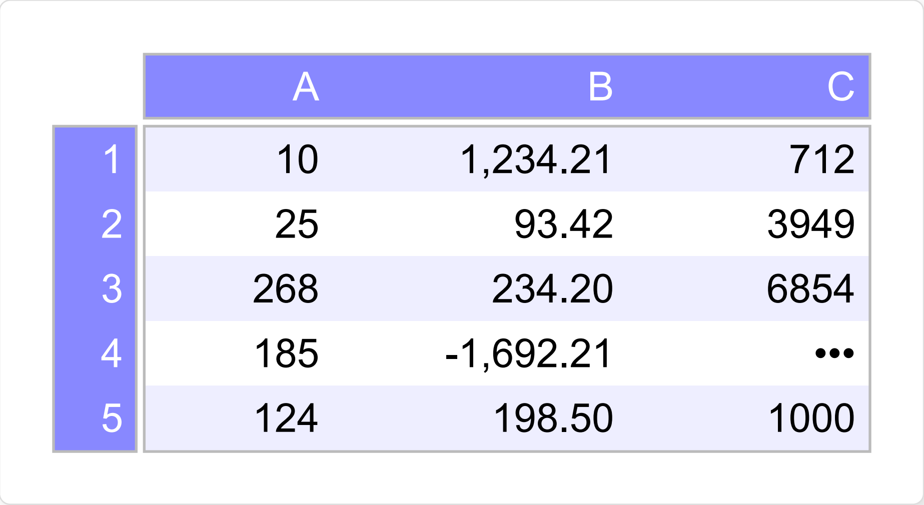
	A	B	C
1	10	1,234.21	712
2	25	93.42	3949
3	268	234.20	6854
4	185	-1,692.21	•••
5	124	198.50	1000

```
OpenDrawing(400;200)
  OpenTable(0;0;left;top)
    TableTexts("";"\tA\tB\tC\r   1\t10\t1,234.21\t712\n   2\t25\t93.42\t3949\n
3\t268\t234.20\t6854\n   4\t185\t-1,692.21\t•••\n   5\t124\t198.50\t1000")
    TableCellStyle(0;Verdana;15;plain;black) /* 0...all cells. */
    TableColumnAlignments(left;right;right;right)
    TableColumnWidths(0;60;100;80) /* 0...automatic. */
    /* Set up column headers. */
    /* [fromRow fromColumn toRow toColumn] -1...lastRow or lastColumn */
    TableCellBorderRect(1 2 1 -1;1)
    TableRowGap(1;3)
    /* Set up rows headers. */
    TableCellBorderRect(2 1 -1 1;1)
    TableColumnGap(1;3)
    /* Set up table. */
    TableCellBorderRect(2 2 -1 -1;1)
    Background(white;;0.5;lightGray;;2 2 5;lightGray)
    BackgroundOptions(4;20)
  CloseTable()
CloseDrawing()
```



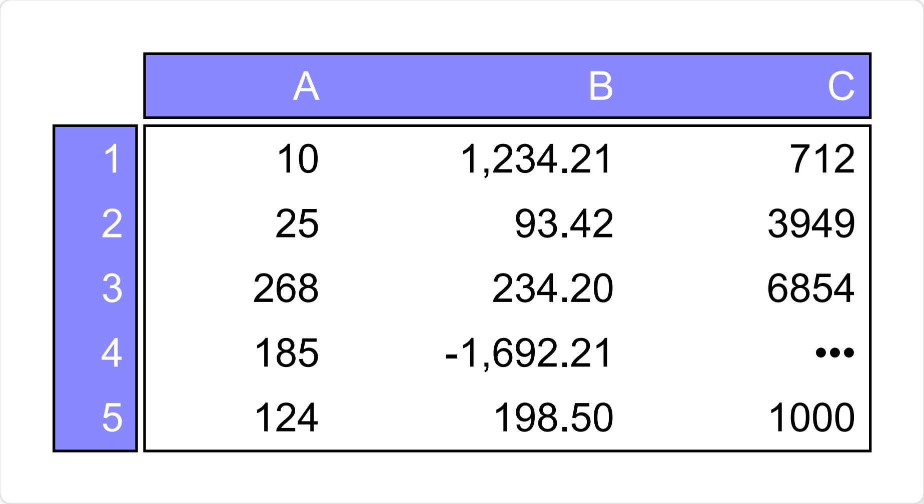
	A	B	C
1	10	1,234.21	712
2	25	93.42	3949
3	268	234.20	6854
4	185	-1,692.21	•••
5	124	198.50	1000

```
OpenDrawing(400;200)
  OpenTable(0;0;left;top)
    TableTexts("";"\tA\tB\tC\r   1\t10\t1,234.21\t712\n   2\t25\t93.42\t3949\n
3\t268\t234.20\t6854\n   4\t185\t-1,692.21\t•••\n   5\t124\t198.50\t1000")
    TableCellStyle(0;Verdana;15;plain;black) /* 0...all cells. */
    TableColumnAlignments(left;right;right;right)
    TableColumnWidths(0;60;100;80) /* 0...automatic. */
    /* Set up column headers. */
    /* [fromRow fromColumn toRow toColumn] -1...lastRow or lastColumn */
    TableCellStyle(1 2 1 -1;Arial;15;plain;white)
    TableCellBackground(1 2 1 -1;#88f)
    TableCellBorderRect(1 2 1 -1;1;#bbb)
    TableRowGap(1;3)
    /* Set up rows headers. */
    TableCellStyle(2 1 -1 1;Arial;15;plain;white)
    TableCellBackground(2 1 -1 1;#88f)
    TableCellBorderRect(2 1 -1 1;1;#bbb)
    TableColumnGap(1;3)
    /* Set up table. */
    TableCellBorderRect(2 2 -1 -1;1;#bbb)
    TableCellBackground(2 2 2 -1  /* Line 2. */
                        4 2 4 -1  /* Line 4. */
                        6 2 6 -1; /* Line 6. */
                        #eef)
    Background(white;;0.5;lightGray;;2 2 5;lightGray)
    BackgroundOptions(4;20)
  CloseTable()
CloseDrawing()
```



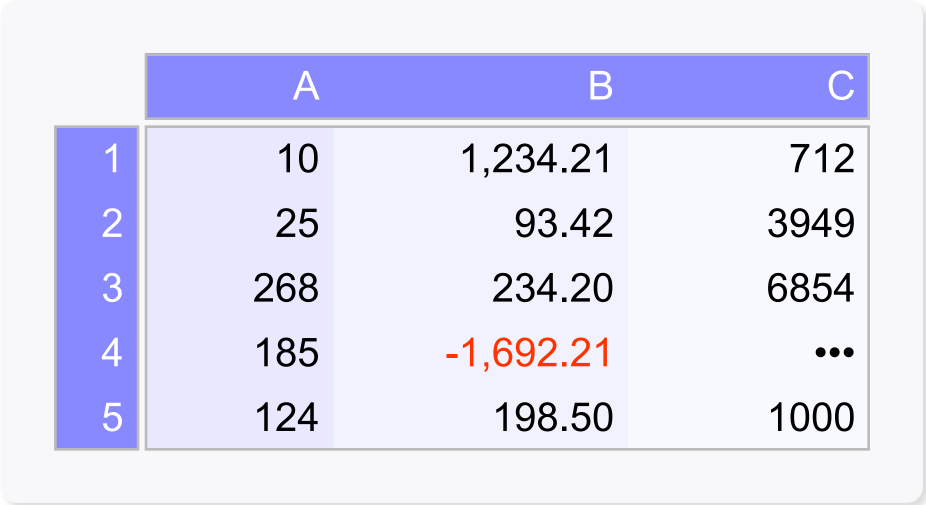
	A	B	C
1	10	1,234.21	712
2	25	93.42	3949
3	268	234.20	6854
4	185	-1,692.21	•••
5	124	198.50	1000

```
OpenDrawing(400;200)
  OpenTable(0;0;left;top)
    TableTexts("";"\tA\tB\tC\r   1\t10\t1,234.21\t712\n   2\t25\t93.42\t3949\n
3\t268\t234.20\t6854\n   4\t185\t-1,692.21\t•••\n   5\t124\t198.50\t1000")
    TableCellStyle(0;Verdana;15;plain;black) /* 0...all cells. */
    TableColumnAlignments(left;right;right;right)
    TableColumnWidths(0;60;100;80) /* 0...automatic. */
    /* Set up column headers. */
    /* [fromRow fromColumn toRow toColumn] -1...lastRow or lastColumn */
    TableCellStyle(1 2 1 -1;Arial;15;plain;white)
    TableCellBackground(1 2 1 -1;#88f)
    TableCellBorderRect(1 2 1 -1;1)
    TableRowGap(1;3)
    /* Set up rows headers. */
    TableCellStyle(2 1 -1 1;Arial;15;plain;white)
    TableCellBackground(2 1 -1 1;#88f)
    TableCellBorderRect(2 1 -1 1;1)
    TableColumnGap(1;3)
    /* Set up table. */
    TableCellBorderRect(2 2 -1 -1;1)
    Background(white;;0.5;lightGray;;2 2 5;lightGray)
    BackgroundOptions(4;20)
  CloseTable()AAAA
CloseDrawing()
```



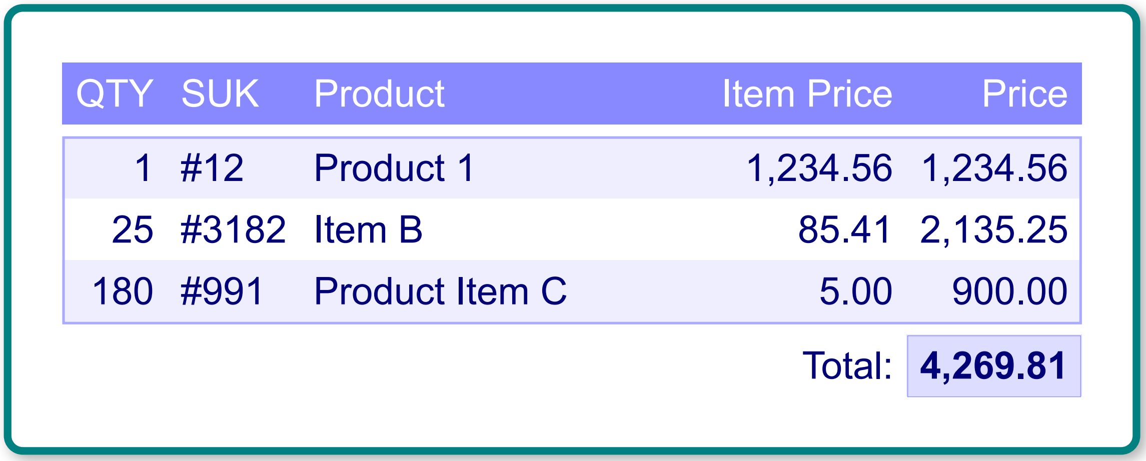
	A	B	C
1	10	1,234.21	712
2	25	93.42	3949
3	268	234.20	6854
4	185	-1,692.21	•••
5	124	198.50	1000

```
OpenDrawing(400;200)
  OpenTable(0;0;left;top)
    TableTexts("";"\tA\tB\tC\r   1\t10\t1,234.21\t712\n   2\t25\t93.42\t3949\n
3\t268\t234.20\t6854\n   4\t185\t-1,692.21\t•••\n   5\t124\t198.50\t1000")
    TableCellStyle(0;Verdana;15;plain;black) /* 0...all cells. */
    TableColumnAlignments(left;right;right;right)
    TableColumnWidths(0;60;100;80) /* 0...automatic. */
    /* Set up column headers. */
    /* [fromRow fromColumn toRow toColumn] -1...lastRow or lastColumn */
    TableCellStyle(1 2 1 -1;Arial;15;plain;white)
    TableCellBackground(1 2 1 -1;#88f)
    TableCellBorderRect(1 2 1 -1;1;#bbb)
    TableRowGap(1;3)
    /* Set up rows headers. */
    TableCellStyle(2 1 -1 1;Arial;15;plain;white)
    TableCellBackground(2 1 -1 1;#88f)
    TableCellBorderRect(2 1 -1 1;1;#bbb)
    TableColumnGap(1;3)
    /* Set up table. */
    TableCellBorderRect(2 2 -1 -1;1;#bbb)
    TableCellBackground(2 2 -1 2;#e8e8ff)
    TableCellBackground(2 3 -1 3;#f2f2ff)
    TableCellBackground(2 4 -1 4;#f8f8ff)
    TableCellStyle(5 3;Verdana;15;plain;red)
    Background(#f8f8fb;;0.5;lightGray;;2 2 5;lightGray)
    BackgroundOptions(4;20)
  CloseTable()
CloseDrawing()
```



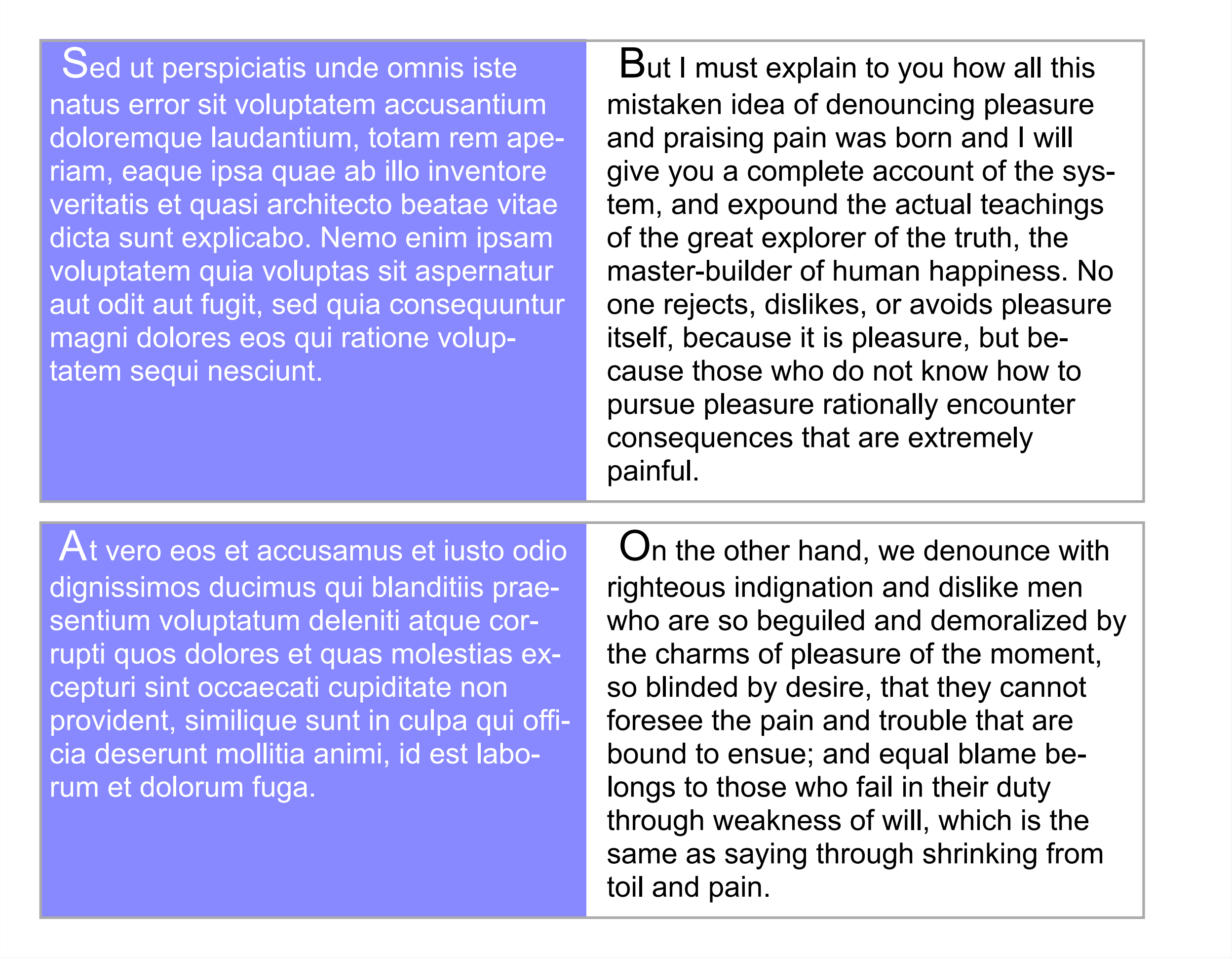
	A	B	C
1	10	1,234.21	712
2	25	93.42	3949
3	268	234.20	6854
4	185	-1,692.21	•••
5	124	198.50	1000

```
OpenDrawing(480;200)
  OpenTable(0;0;left;top)
    TableTexts("";"QTY\tSUK\tProduct\tItem Price\tPrice\n1\t#12\tProduct
1\t1,234.56\t1,234.56\n25\t#3182\tItem B\t85.41\t2,135.25\n180\t#991\tProduct Item
C\t5.00\t900.00\n\t\t\tTotal:\t4,269.81")
    TableColumnWidths(0;0;150;0;0) /* 0...automatic */
    TableColumnAlignments(right;left;left;right;right)
    TableRowAlternateBackgrounds(white;#eef) /* Row1 row2 ... */
    TableRowGap(1 4;5) /* Gap between row 1&2 and row 4&5. */
    /* Set up row 1: Header */
    /* 1 1 1 -1...fromRow fromColumn toRow toColumn. -1...lastRow or lastColumn */
    TableCellStyle(1 1 1 -1;Arial;15;plain;white)
    TableCellBackground(1 1 1 -1;#88f)
    /* Set up row 2..4: Body */
    /* 2 1 -1 -1...fromRow fromColumn toRow toColumn. -1...lastRow or lastColumn */
    TableCellStyle(2 1 -1 -1;Arial;15;plain;darkBlue)
    /* 2 1 4 5...fromRow fromColumn toRow toColumn */
    TableCellBorderRect(2 1 4 5;1;#aaf)
    /* Set up row 5: Footer */
    /* -1 -1...lastRow lastColumn */
    TableCellStyle(-1 -1;Arial;15;bold;darkBlue)
    TableCellBorderRect(-1 -1;0.25;#88f)
    TableCellBackground(-1 -1;#ddf)
    Background(white;0;3;teal;solid;2 2 5;  )
    BackgroundOptions(6;20)
  CloseTable()
CloseDrawing()
```



QTY	SUK	Product	Item Price	Price
1	#12	Product 1	1,234.56	1,234.56
25	#3182	Item B	85.41	2,135.25
180	#991	Product Item C	5.00	900.00
			Total:	**4,269.81**

```
OpenDrawing(450;350)
  OpenTable(10;10;left;top)
    TableTexts("";"<span size=16> S</span>ed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo
inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam
voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni
dolores eos qui ratione voluptatem sequi nesciunt.\t<span size=16> B</span>ut I must explain
to you how all this mistaken idea of denouncing pleasure and praising pain was born and I
will give you a complete account of the system, and expound the actual teachings of the
great explorer of the truth, the master-builder of human happiness. No one rejects,
dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not
know how to pursue pleasure rationally encounter consequences that are extremely painful.
\n<span size=16> A</span>t vero eos et accusamus et iusto odio dignissimos ducimus qui
blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias
excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia
deserunt mollitia animi, id est laborum et dolorum fuga.\t<span size=16> O</span>n the other
hand, we denounce with righteous indignation and dislike men who are so beguiled and
demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot
foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who
fail in their duty through weakness of will, which is the same as saying through shrinking
from toil and pain.")
    TableRowGap(1;8)
    TableColumnGap(1;4)
    TableColumnWidths(200;200)
    TableCellBorderRect(1 1 1 2;1;#aaa)
    TableCellBorderRect(2 1 2 2;1;#aaa)
    /* Left column. */
    TableCellStyle(1 1 -1 1;Verdana;11;plain;white;left;top)
    TableCellBackground(1 1;#88f)
    TableCellBackground(2 1;#88f)
    /* Right column. */
    TableCellStyle(1 2 -1 2;Verdana;11;plain;black;left;top)
  CloseTable()
CloseDrawing()
```



Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful.

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga.

On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain.

Tags

A tag consists of a text object and an optional symbol or picture object and an optional background.

Tag structure:

```
OpenTag(xAnchor;yAnchor;hAlignment;vAlignment)
   TagText(...)
   TagPicture(...) or TagSymbol(...) /* Optional. */
   Background(...)                   /* Optional. */
   BackgroundOptions(...)            /* Optional. */
CloseTag()
```

Tag functions:

```
TagPicture(width;height;position;alignment;gap;sourceType;sourceData;
           borderStroke;borderColor;borderColorVariant;shadowEffect;shadowColor)
TagSymbol(width;height;position;alignment;gap;symbolType;symbolStroke;symbolColor;
          symbolColorVariant;symbolBackgroundColor;symbolBackgroundColorVariant;
          shadowEffect;shadowColor)
TagText(text;fontName;textSize;textStyle;textColor;hAlignment;textWidthMax;textHeightMax;
        ellipsisPosition;shadowEffect;shadowColor)
```

OpenTag(xAnchor;yAnchor;hAlignment;vAlignment)

Arguments	*Type*	*Range*	*Default*	*Notes*
xAnchor	num	-inf..+inf	(required)	
yAnchor	num	-inf..+inf	(required)	
hAlignment	int	1..3	left	
vAlignment	int	1..3	top	

```
TagPicture(width;height;position;alignment;gap;sourceType;sourceData;borderStroke;
           borderColor;borderColorVariant;shadowEffect;shadowColor)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
width	num	0..10000	(image width)	
height	num	0..10000	(image height)	
position	int	1..4	left	*Relative to text.*
alignment	int	1..3	2	
gap	num	-10000..10000	2 %	
sourceType	int	1..4	1	
sourceData	str		""	
borderStroke	num[]	0..1000	0	
borderColor	rgba	0..255	none	
borderColorVariant	int	-1..128	0	
shadowEffect	num[]	-1000..1000	0	
shadowColor	rgba	0..255	#888a	

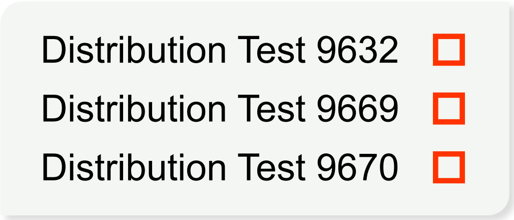
```
TagSymbol(width;height;position;alignment;gap;symbolType;symbolStroke;symbolColor;
          symbolColor;symbolColorVariant;symbolBackgroundColor;
          symbolBackgroundColorVariant;shadowEffect;shadowColor)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
width	num	0..10000	0	
height	num	0..10000	0	
position	num	1..4	left	*Relative to text.*
alignment	int	1..3	2	
gap	num	-10000..10000	2 %	
symbolType	int	0..126	3	
symbolStroke	num[]	0..1000	1	
symbolColor	rgba	0..255	black	
symbolColorVariant	int	-1..128	0	
symbolBackgroundColor	rgba	0..255	none	
symbolBackgroundColorVariant	int	-1..128	0	
shadowEffect	num[]	-1000..1000	0	
shadowColor	rgba	0..255	#888a	

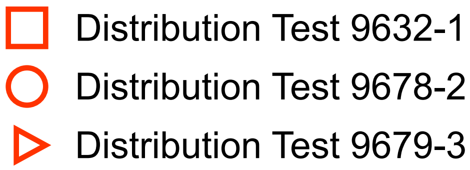
Arguments	*Type*	*Range*	*Default*	*Notes*
text	styt	0..10000	""	
fontName	str	0..1000	Arial	
textSize	num	0..1000	9	
textStyle	int	0..7	0	
textColor	rgba	0..255	black	
hAlignment	int	1..3	1	
textWidthMax	num[]	-1..10000	-1	
textHeightMax	num[]	-1..10000	-1	
ellipsisPosition	int	0..4	3	
shadowEffect	num[]	-1000..1000	0	
shadowColor	rgba	0..255	#888a	

Examples:

```
OpenDrawing(200;110)
  OpenGroup()
    OpenTag(20;20)
      TagSymbol(16;16;right;center;10;squareOutline;2;red)
      TagText("Distribution Test 9632";Arial;14)
    CloseTag()
    OpenTag(20;42)
      TagSymbol(16;16;right;center;10;squareOutline;2;red)
      TagText("Distribution Test 9669";Arial;14)
    CloseTag()
    OpenTag(20;64)
      TagSymbol(16;16;right;center;10;squareOutline;2;red)
      TagText("Distribution Test 9670";Arial;14)
    CloseTag()
    Background(#f3f6f2;0;0;;;2 2 2;lightGray)
    BackgroundOptions(8 0;10 15)
  CloseGroup()
CloseDrawing()
```



```
OpenDrawing(200;90)
  OpenTag(10;10)
    TagSymbol(20;20; /* Symbol width and height.          */
              left;  /* Symbol position relative to text. */
              top;   /* Symbol alignment.                 */
              8;     /* Gap between symbol and text.      */
              squareOutline;2;red)
    TagText("Distribution Test 9632-1";Arial;14)
  CloseTag()
  OpenTag(10;32)
    TagSymbol(20;20; /* Symbol width and height.          */
              left;  /* Symbol position relative to text. */
              top;   /* Symbol alignment.                 */
              8;     /* Gap between symbol and text.      */
              circle;2;red)
    TagText("Distribution Test 9678-2";Arial;14)
  CloseTag()
  OpenTag(10;54)
    TagSymbol(20;20; /* Symbol width and height.          */
              left;  /* Symbol position relative to text. */
              top;   /* Symbol alignment.                 */
              8;     /* Gap between symbol and text.      */
              triangleRightOutline;2;red)
    TagText("Distribution Test 9679-3";Arial;14)
  CloseTag()
  // Background() /* Uncomment while positioning the tag element. */
CloseDrawing()
```

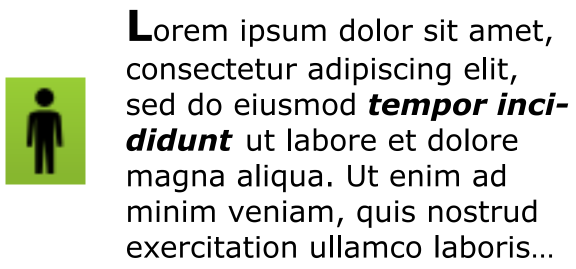


□ Distribution Test 9632-1

○ Distribution Test 9678-2

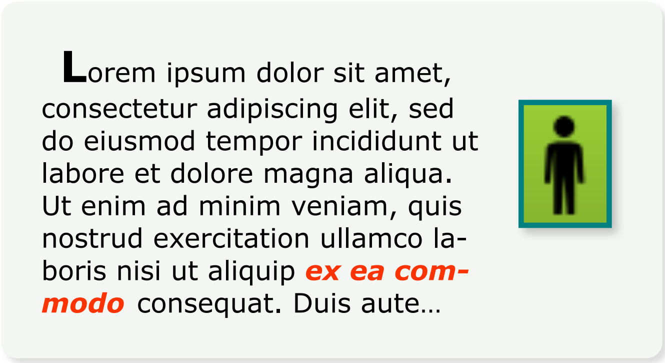
▷ Distribution Test 9679-3

```
OpenDrawing(250;150)
  OpenTag(20;20)
    TagPicture(30;40;left;center;15;stream;"data:image/png;base64,
iVBORw0KGgoAAAANSUhEUgAAAB4AAAAoCAIAAABmcd1FAAACdUlEQVRIx2OccdaUgTaAiYFmgOn/
fwYaISaG/ww0QozTTpsQ9NqzO99X9z16cPWrkDi7Y6SYTaAoMQHCOO2UCUFzO+Kv/fn9Hy7ikSjp
lylNRFgT8tnOhc+RzQWCvcte/vr5j6BGwmH99M53NOf8/vnvzdOfBDUSTiHCUuxoRjOzMvKLsBGR
QggB3wxpoFnIIm5xkpw8zMSkawJAUomjcIa6iiEvExMj0AeBeTJeKZL/iQCMk48Z0yg3Mk7Ca/Tr
Jz9vn/385Pa3f38RiURcjkNajUvNmJeQ0UexG/339/+ZpXdunP6ES6ewJHveVDVBcTbcxROpHP5
8Ac85gLB2+c/t815hjfx4ZD69O43xAgeARZMcyGCb5/9IifLfP/yF5KE+YRYMY2GhMOXD7/xuxp7
kocYLSDC+h9baPCLguz7/vUvnkyD09U/vv4FG4E9loC5EUj+AFpPRkaHuloMu9GCYiBX//z+78+v
/yRndKjRoqzYjYalue9gz5FWgUEChE8Yu9F8ImxwF5BcgX379BfscewBwsXDzM7JhD+4cacQiKuR
AgRiFgQAFfCLgKS+fvpDZgoRRDJaSBKp4P4PjWG8rsZhqbg8h4waFxc/CwMsYYvKsCOZzCCvxQ1U
wMTCSHIKKZiuDkTAMhouIiKNUt24J0gCFWhb8ZPfxEEkOAlElBLXxCEa8PCzcPEyk9QwIwRggc3J
y8zJy4JIIoQAEQ0zGODmY+HkZkbEI8HGAvEe5AK5mpmaLVU4AIYGPKyp11JlYGBlY2JmZuTmZ0HO
NYRaT8QFNTAo/oMdTnRQMxAbjTzgbMnBRUI0MrZv1x+CfZkhaTQAZhyc0O25ucYAAAAASUVORK5C
YII=")
    TagText("<span size=16><b>L</b></span>orem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod <b><i>tempor incididunt</i></b> ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum
dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim id est laborum.";Verdana;11;plain;black;left;170;100)
  CloseTag()
  // Background() /* Uncomment while positioning the tag element. */
CloseDrawing()
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod ***tempor incididunt*** ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris…

```
OpenDrawing(260;160)
  OpenTag(130;80;center;center)
    TagPicture(39;52;right;center;15;stream;"data:image/png;base64,
iVBORw0KGgoAAAANSUhEUgAAAB4AAAAoCAIAAABmcd1FAAACdUlEQVRIx2OccdaUgTaAiYFmgOn/
fwYaISaG/ww0QozTTpsQ9NqzO99X9z16cPWrkDi7Y6SYTaAoMQHCOO2UCUFzO+Kv/fn9Hy7ikSjp
lylNRFgT8tnOhc+RzQWCvcte/vr5j6BGwmH99M53NOf8/vnvzdOfBDUSTiHCUuxoRjOzMvKLsBGR
QggB3wxpoFnIIm5xkpw8zMSkawJAUomjcIa6iiEvExMj0AeBeTJeKZL/iQCMk48Z0yg3Mk7Ca/Tr
Jz9vn/385Pa3f38RiURcjkNajUvNmJeQ0UexG/339/+ZpXdunP6ES6ewJHveVDVBcTbcxROOpHP5
8Ac85gLB2+c/t815hjfx4ZD69O43xAgeARZMcyGCb5/9IifLfP/yF5KE+YRYMY2GhMOXD7/xuxp7
kocYLSDC+h9baPCLguz7/vUvnkyD09U/vv4FG4E9loC5EUj+AFpPRkaHuloMu9GCYiBX//z+78+v
/yRndKjRoqzYjYalue9gz5FWgUEChE8Yu9F8ImxwF5BcgX379BfscewBwsXDzM7JhD+4cacQiKuR
AgRiFgQAFfCLgKS+fvpDZgoRRDJaSBKp4P4PjWG8rsZhqbg8h4waFxc/CwMsYYvKsCOZzCCvxQ1U
wMTCSHIKKZiuDkTAMhouIiKNUt24J0gCFWhb8ZPfxEEkOAlElBLXxCEa8PCzcPEyk9QwIwRggc3J
y8zJy4JIIoQAEQ0zGODmY+HkZkbEI8HGAvEe5AK5mpmaLVU4AIYGPKyp11JlYGBlY2JmZuTmZ0HO
NYRaT8QFNTAo/oMdTnRQMxAbjTzgbMnBRUI0MrZv1x+CfZkhaTQAZhyc0O25ucYAAAAASUVORK5C
YII=";2;teal;0;2 2 3;#aaaa)
    TagText("  <span size=16><b>L</b></span>orem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip <span
color=red><i>ex ea commodo</i></span> consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.";Verdana;10;plain;black;left;165;120)
    Background(#f3f6f2;0;0;;;2 2 2;lightGray)
    BackgroundOptions(6;8 15 15 15)
  CloseTag()
CloseDrawing()
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ***ex ea commodo*** consequat. Duis aute...

Groups

A group make it possible to:

- add a background to a group of objects.
- rotate a group of objects, i.e. charts, graphic primitives, groups, tags, tables, views.

Nested groups, i.e. groups within a group, are also possible.

Group structure:

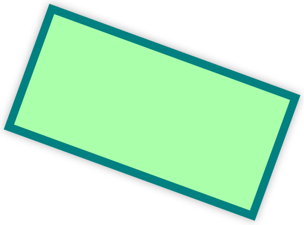
```
OpenGroup(rotation)
   ...
   list of objects (charts, graphics primitives, groups, tags, tables, views)

   Hint: To precisely place a chart enclose the chart functions in
         OpenChart()...CloseChart().
   ...
   Background(...)        /* Optional. */
   BackgroundOptions(...) /* Optional. */
CloseGroup()
```

OpenGroup(rotation)

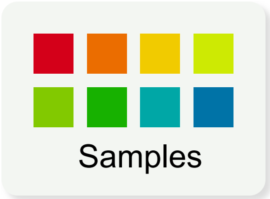
Arguments	*Type*	*Range*	*Default*	*Notes*
rotation	num	-360..360	0	

Examples:

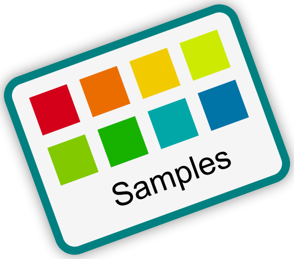
```
OpenDrawing(200;100)
  OpenGroup(20) /* Rotate clockwise by 20 degrees. */
    AddRect(50;25;100;50;#afa;solid;3;teal;solid;0 0 3)
  CloseGroup()
CloseDrawing()
```



```
OpenDrawing(180;100)
  OpenGroup()
    AddRect( 50;25;15;15;#d40019;solid;0)
    AddRect( 70;25;15;15;#eb6d00;solid;0)
    AddRect( 90;25;15;15;#f1cb00;solid;0)
    AddRect(110;25;15;15;#cdea03;solid;0)
    AddRect( 50;45;15;15;#82c900;solid;0)
    AddRect( 70;45;15;15;#17b100;solid;0)
    AddRect( 90;45;15;15;#00a7a6;solid;0)
    AddRect(110;45;15;15;#0073a7;solid;0)
    AddText(90;75;"Samples";Arial;12;plain;black;center)
    Background(#f3f6f2;0;0;;;2 2 2;lightGray)
    BackgroundOptions(6;12 12 8 12)
  CloseGroup()
CloseDrawing()
```



```
OpenDrawing(180;105)
  OpenGroup(-20) /* Rotate counter-clockwise by 20 degrees. */
    AddRect( 50;25;15;15;#d40019;solid;0)
    AddRect( 70;25;15;15;#eb6d00;solid;0)
    AddRect( 90;25;15;15;#f1cb00;solid;0)
    AddRect(110;25;15;15;#cdea03;solid;0)
    AddRect( 50;45;15;15;#82c900;solid;0)
    AddRect( 70;45;15;15;#17b100;solid;0)
    AddRect( 90;45;15;15;#00a7a6;solid;0)
    AddRect(110;45;15;15;#0073a7;solid;0)
    AddText(90;75;"Samples";Arial;12;plain;black;center)
    Background(whiteSmoke;0;3;teal;solid;0 0 5;gray)
    BackgroundOptions(6;6)
  CloseGroup()
CloseDrawing()
```



Color Scale Legends

The functions `LegendColorScale()` and `LegendColorTones()` make it possible to add a legend to heat maps, tree maps and density plots.

```
LegendColorScale(scaleValues;doReverseScale;scaleInset;tickLength;tickOffset;tickStroke;
                 tickColor;doMirrorTicks;numOfSubTicks;subTickLength;labelFormat;
                 borderStroke;borderColor)
```

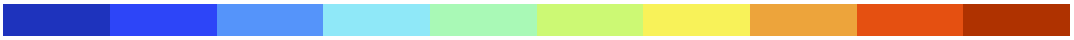
Arguments	*Type*	*Range*	*Default*	*Notes*
scaleValues	num[]	-inf..+inf	(required)	
doReverseScale	int	0..1	0	
scaleInset	num[]	0..1000	0	*Dimension: [pt]*
tickLength	num	0..1000	5	*Dimension: [pt]*
tickOffset	num	-1000..1000	0	*Dimension: [pt]*
tickStroke	num[]	0..1000	1	*Dimension: [pt]*
tickColor	rgba	0..255	black	
doMirrorTicks	int	0..1	0	
numOfSubTicks	int[]	0..1000	0	
subTickLength	num	0..1000	3	*Dimension: [pt]*
labelFormat	styt	0..1000	"|u|"	*See: Output Numbers*
borderStroke	num[]	0..1000	0	*Dimension: [pt]*
borderColor	rgba	0..255	black	

The scale values can be overridden by using function `LegendTexts()`.

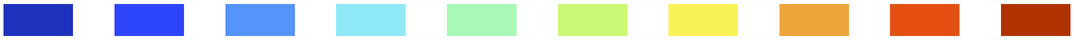
```
LegendColorTones(scaleValueMin;scaleValueMax;numOfColorTones;doReverseScale;colorToneGap;
                 labelFormat;borderStroke;borderColor)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
scaleValueMin	num	-inf..+inf	(required)	
scaleValueMax	num	-inf..+inf	(required)	
numOfColorTones	int	1..100	(required)	
doReverseScale	int	0..1	0	
colorToneGap	num	0..1	1	*Relative to color tone size.*
labelFormat	styt	0..1000	"|u|-|u|"	*See: Output Numbers*
borderStroke	num[]	0..1000	0	*Dimension: [pt]*
borderColor	rgba	0..255	black	

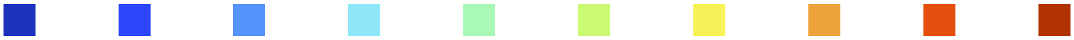
colorToneGap = 0:



colorToneGap = 0.5:



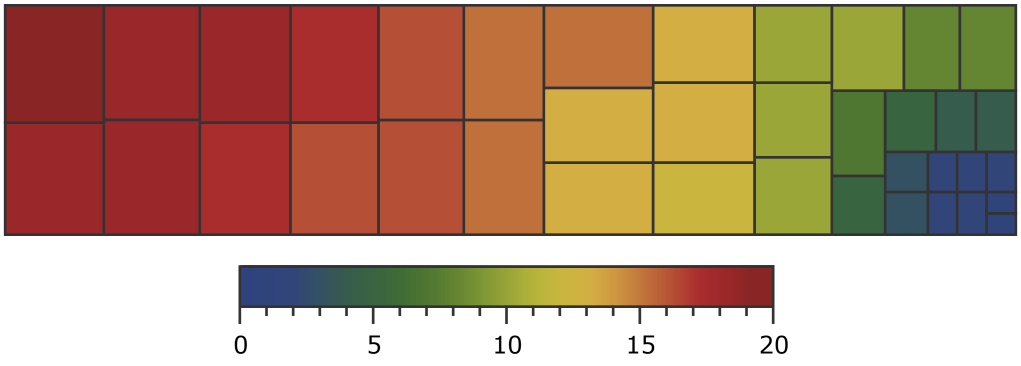
colorToneGap = 1:



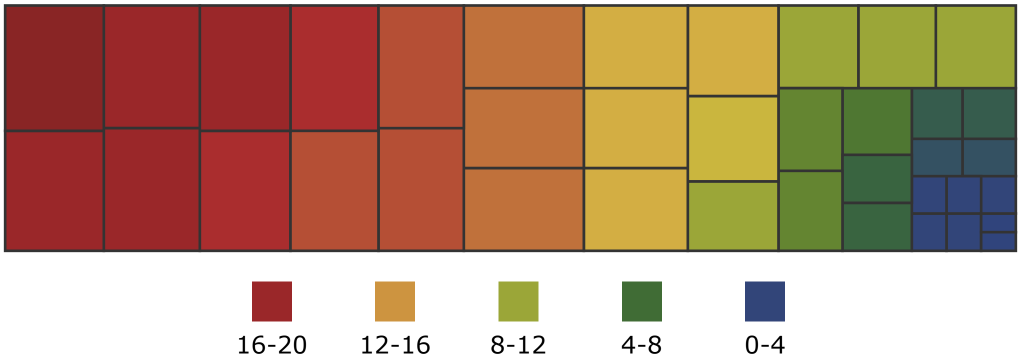
The scale values can be overridden by using function `LegendTexts()`.

Examples:

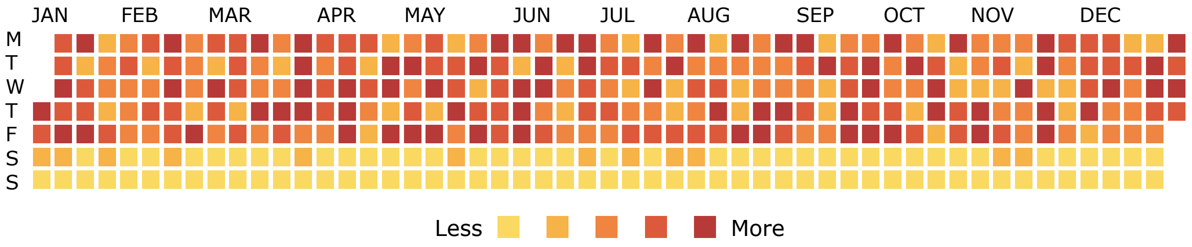
```
OpenDrawing(400;150)
  ChartData(4 2 1 0 1 2 4 7 10 13 16 18 19 18 18 17 15 13 10 8 5 3 2 2 2 3 5 8 10 13 15 17
18 16 16 15 13 12 10)
  TreeMap()
  FillColorScale(1;9)
  BorderStyle(all;;1;#333)
  LegendStyle(Verdana;9)
  LegendOptions(bottomCenter;
                0;  /* Place legend inside of plot area. */
                0;  /* Horizontal offset.                */
                3;  /* Vertical offset.                  */
                1;  /* Number of rows (ignored).         */
                0;  /* Legend item type (ignored).       */
              200;  /* Color scale length.               */
               15;  /* Color scale height.               */
                3;  /* Gap between text and ticks.       */
                0;  /* Row gap (ignored).                */
                0;  /* Column gap (ignored).             */
     bottomCenter)  /* Text location.                    */
  LegendColorScale(0 5 10 15 20; /* Scale values.     */
                   0;            /* Do reverse scale. */
                   0;            /* Scale inset.      */
                   6;            /* Tick length.      */
                   0;            /* Tick offset.      */
                   1;            /* Tick stroke.      */
                   #333;         /* Tick color.       */
                   0;            /* Do mirror ticks.  */
                   4;            /* Num of sub-ticks. */
                   3;            /* Sub-tick length.  */
                   "|u|";        /* Label format.     */
                   1;            /* Border stroke.    */
                   #333)         /* Border color.     */
CloseDrawing()
```



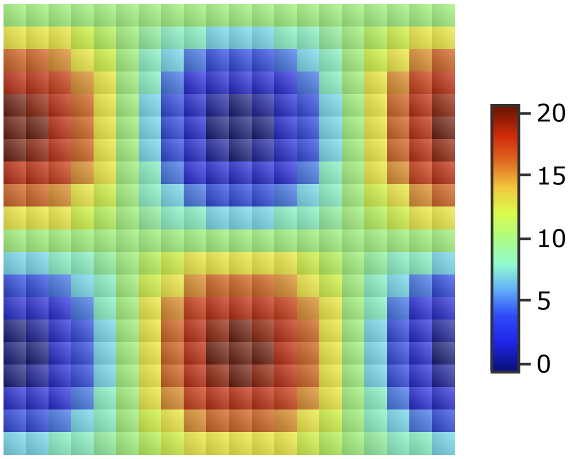
```
OpenDrawing(400;150)
  ChartData(4 2 1 0 1 2 4 7 10 13 16 18 19 18 18 17 15 13 10 8 5 3 2 2 2 3 5 8 10 13 15 17
18 16 16 15 13 12 10)
  TreeMap()
  FillColorScale(1; 9)
  BorderStyle(all;;1;#333)
  LegendStyle(Verdana;9)
  LegendOptions(bottomCenter;
                0;  /* Place legend inside of plot area. */
                0;  /* Horizontal offset.                */
                3;  /* Vertical offset.                  */
                1;  /* Number of rows (ignored).         */
                0;  /* Legend item type (ignored).       */
              200;  /* Color scale length.               */
               15;  /* Color scale height.               */
                3;  /* Gap between text and ticks.       */
                0;  /* Row gap (ignored).                */
                0;  /* Column gap (ignored).             */
     bottomCenter)  /* Text location.                    */
  LegendColorTones(0;        /* Scale value Min.         */
                  20;        /* Scale value Max.         */
                   5;        /* Number of color tones.   */
                   1;        /* Do reverse scale.        */
                   1;        /* Gap between color tones. */
                  "|u|-|u|"; /* Label format.            */
                   0;        /* Border stroke.           */
                   black)    /* Border color.            */
CloseDrawing()
```



```
OpenDrawing(560;120)
  OpenChart(20;20;530;72;on)
    ChartData(null null null 32 24 9 0 27 30 36 26 32 8 5 32 14 27 24 32 4 4 14 20 18 14 27 9 2 23 25 23
18 21 4 1 31 15 23 30 16 2 3 33 31 38 24 28 8 2 21 21 16 11 38 6 3 25 11 33 27 20 4 0 24 27 29 10 25 5 4
32 17 18 39 18 6 3 23 13 23 37 24 6 1 36 34 35 36 18 10 1 25 23 25 24 18 1 2 26 24 40 39 32 7 0 28 13 25
17 15 6 2 11 33 34 12 36 4 3 19 39 21 28 37 2 3 24 24 36 14 40 6 4 15 31 24 35 21 10 3 16 38 10 29 36 3
1 39 26 24 25 27 4 4 37 13 36 38 33 5 3 20 32 33 22 29 3 1 38 12 17 14 20 6 0 37 39 31 30 20 8 1 17 24
23 27 23 2 3 10 30 15 21 25 10 0 34 18 34 17 27 1 4 22 36 15 13 27 8 4 36 19 24 21 27 9 0 13 21 29 35 29
2 3 36 19 11 15 36 5 1 18 18 19 34 38 1 2 33 19 18 34 24 2 1 38 27 22 25 20 4 3 15 39 11 12 20 0 2 18 26
24 34 32 3 4 19 33 37 31 32 3 4 38 18 20 26 34 4 3 20 37 23 12 30 5 1 14 30 39 33 15 2 2 34 11 11 26 28
5 4 19 20 14 34 39 6 1 16 29 11 18 28 8 5 23 13 36 18 19 10 4 34 36 15 35 35 6 2 31 18 13 15 21 7 4 29
24 24 33 13 2 5 27 29 37 17 23 3 5 12 28 22 19 21 5 2 14 37 34 26 18 6 2 33 29 33 24)
    HeatMap(;7;53) /* 7 rows, 53 columns. */
    HeatMapOptions(0.4%;;;on)
    /* Set up styles. */
    FillColorScale(1;-18;solid;0.8;5)
    BorderStyle(1;none) /* Hide borders. */
    LegendTexts("Less";;;;"More")
    LegendStyle(Verdana;10)
    LegendOptions(bottomCenter;
                  0;  /* Place legend inside of plot area. */
                  0;  /* Horizontal offset.                */
                  3;  /* Vertical offset.                  */
                  1;  /* Number of rows (ignored).         */
                  0;  /* Legend item type (ignored).       */
                100;  /* Color scale length.               */
                 10;  /* Color scale height.               */
                  7;  /* Gap between text and ticks.       */
                  0;  /* Row gap (ignored).                */
                  0;  /* Column gap (ignored).             */
                  4)  /* Text location.                    */
    LegendColorTones(0;        /* Scale value Min.           */
                    40;        /* Scale value Max.           */
                     5;        /* Number of color tones.     */
                     0;        /* Do reverse scale.          */
                     1;        /* Gap between color tones.   */
                    "";        /* Label format.              */
                     0;        /* Border stroke.             */
                     black)    /* Border color.              */
  CloseChart()
  AddText(8;26;"M\nT\nW\nT\nF\nS\nS";Verdana;9)
  AddText( 20;15;"JAN";Verdana;9)
  AddText( 61;15;"FEB";Verdana;9)
  AddText(101;15;"MAR";Verdana;9)
  AddText(151;15;"APR";Verdana;9)
  AddText(191;15;"MAY";Verdana;9)
  AddText(241;15;"JUN";Verdana;9)
  AddText(281;15;"JUL";Verdana;9)
  AddText(321;15;"AUG";Verdana;9)
  AddText(371;15;"SEP";Verdana;9)
  AddText(411;15;"OCT";Verdana;9)
  AddText(451;15;"NOV";Verdana;9)
  AddText(501;15;"DEC";Verdana;9)
CloseDrawing()
```



```
OpenDrawing(230;190)
  OpenChart(10;10;170;170;on)
    ChartData(10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 13 13 13 12 11 10 9 8 8 7 7 7
8 8 9 10 11 12 13 13 16 16 15 13 12 10 8 7 5 4 4 4 5 7 8 10 12 13 15 16 18 18 17 15 13 10 8 5 3 2 2 2 3
5 8 10 13 15 17 18 20 19 18 16 13 10 7 4 2 1 0 1 2 4 7 10 13 16 18 19 20 20 18 16 13 10 7 4 2 0 0 0 2 4
7 10 13 16 18 20 20 19 18 16 13 10 7 4 2 1 0 1 2 4 7 10 13 16 18 19 18 18 17 15 13 10 8 5 3 2 2 2 3 5 8
10 13 15 17 18 16 16 15 13 12 10 8 7 5 4 4 4 5 7 8 10 12 13 15 16 13 13 13 12 11 10 9 8 8 7 7 7 8 8 9 10
11 12 13 13 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 7 7 8 8 9 10 11 12 13 13 13 13
13 12 11 10 9 8 8 7 4 4 5 7 8 10 12 13 15 16 16 16 15 13 12 10 8 7 5 4 2 2 3 5 8 10 13 15 17 18 18 18 17
15 13 10 8 5 3 2 0 1 2 4 7 10 13 16 18 19 20 19 18 16 13 10 7 4 2 1 0 0 2 4 7 10 13 16 18 20 20 20 18 16
13 10 7 4 2 0 0 1 2 4 7 10 13 16 18 19 20 19 18 16 13 10 7 4 2 1 2 2 3 5 8 10 13 15 17 18 18 18 17 15 13
10 8 5 3 2 4 4 5 7 8 10 12 13 15 16 16 16 15 13 12 10 8 7 5 4 7 7 8 8 9 10 11 12 13 13 13 13 13 12 11 10
9 8 8 7)
    HeatMap()
    /* Set up styles. */
    FillColorScale(1;25;shaded)
    BorderStyle(1;none)
    LegendStyle(Verdana;9)
    LegendOptions(centerRight;
              0;  /* Place legend inside of plot area. */
              5;  /* Horizontal offset.                */
              3;  /* Vertical offset.                  */
              1;  /* Number of rows (ignored).         */
              0;  /* Legend item type (ignored).       */
             10;  /* Color scale width.                */
            100;  /* Color scale height.               */
              3;  /* Gap between text and ticks.       */
              0;  /* Row gap (ignored).                */
              0;  /* Column gap (ignored).             */
              6)  /* Text location.                    */
    LegendColorScale(0 5 10 15 20; /* Scale values.    */
                 1;            /* Do reverse scale. */
                 3;            /* Scale inset.      */
                 4;            /* Tick length.      */
                 0;            /* Tick offset.      */
                 1;            /* Tick stroke.      */
                 #333;         /* Tick color.       */
                 0;            /* Do mirror ticks.  */
                 0;            /* Num of sub-ticks. */
                 3;            /* Sub-tick length.  */
                 "|u|";        /* Label format.     */
                 1;            /* Border stroke.    */
                 #333)         /* Border color.     */
  CloseChart()
CloseDrawing()
```



Improved Gantt Charts

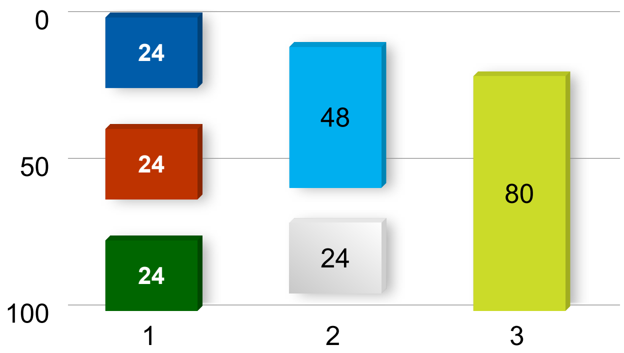
By defining more than two values per series, a task can be divided into several partial tasks. Starting with xmCHART 5 it is possible to assign an individual fill color to each partial task. For example:

```
FillStyle(2;red;solid;   /* Partial task 1 of series 2, fill color: red.         */
            green;solid; /* Partial task 2 of series 2, fill color: green.       */
            blue;shaded) /* Partial task 3 of series 2, fill color: shaded blue. */
```

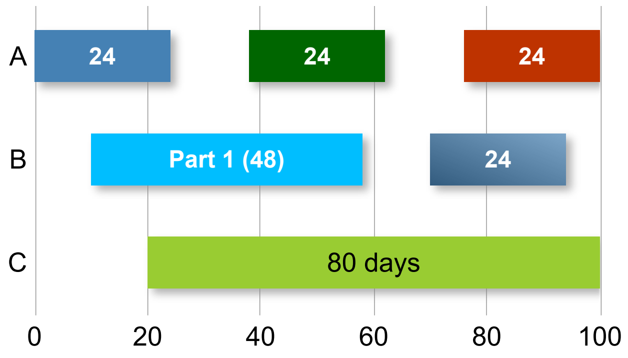
The additional *color;colorVariant* arguments in function `FillStyle()` are only available for Gantt charts and are ignored in combination with other chart types. All other style functions such as `BorderStyle()`, `LabelStyle()`, etc. have not been modified.

Examples:

```
OpenDrawing(250;150)
  ChartData( 0  24  38 62  76 100;
            10  58  70 94;
            20 100)
  GanttChart(label+horizontal+shadow;;15)
  /* Set up styles. */
  FillStyle(1;#005ca9;solid;darkRed;solid;darkGreen)
  FillStyle(2;#00afef;solid;white;shaded)
  FillStyle(3;#cbdb29)
  BorderStyle(all;none)
  ShadowStyle(all;2 2 5)
  LabelStyle(1;;;bold;white)
  /* Set up axes. */
  ScalingOptions(y;on) /* y-scale top to bottom. */
  AxisLine(all;0)
  AxisMajorTicks(all;0)
  /* Set up grid. */
  MajorGridLineWidths(all;all;0)
  MajorGridLineWidths(y;x;0)
  MajorGridLineWidths(x;y;0.25)
CloseDrawing()
```



```
OpenDrawing(250;150)
  ChartData( 0  24  38 62  76 100;
            10  58  70 94;
            20 100)
  GanttChart(label+shadow)
  /* Set up styles. */
  FillStyle(1;steelBlue;0;darkGreen;0;darkRed)
  FillStyle(2;deepSkyBlue;0;steelblue;shaded)
  FillStyle(3;yellowGreen)
  BorderStyle(all;none)
  LabelStyle(1;;;bold;white)
  LabelStyle(2;;;bold;white)
  LabelTexts(2;"Part 1 (|u|)";"|u|")
  LabelTexts(3;"|u| days")
  ShadowStyle(all;2 2 3)
  /* Set up axes. */
  ScalingOptions(y;on) /* y-scale top to bottom. */
  AxisLine(all;0)
  AxisMajorTicks(all;0)
  AxisMajorTickLabelTexts(y;"A";"B";"C")
  /* Set up grid. */
  MajorGridLineWidths(x;y;0)
  MajorGridLineWidths(y;x;0.25)
CloseDrawing()
```



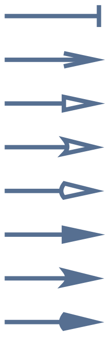
Extended Arrow Object

In addition to function AddArrow(), a new extended function AddArrowExt() has been added to xmCHART 5. Using the 1st argument *pathData*, the path of the arrow can be defined. See function AddPath().

```
AddArrowExt(pathData;stroke;color;colorVariant;headType;headSize;headGap;
            tailType;tailSize;tailGap;shadowEffect;shadowColor)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
pathData	num[]	-inf..+inf	(required)	*See function AddPath()*
stroke	num[]	0..1000	1	*Dimension: [pt]*
color	rgba	0..255	black	
colorVariant	int	-1..128	solid	
headType	int	-8..126	-7	
headSize	num	0..1000	10	*Dimension: [pt]*
headGap	num	-1000..1000	0	*Dimension: [pt]*
tailType	int	-8..126	0	
tailSize	num	0..1000	10	*Dimension: [pt]*
tailGap	num	-1000..1000	0	*Dimension: [pt]*
shadowEffect	num[]	-1000..1000	0	
shadowColor	rgba	0..255	#888a	

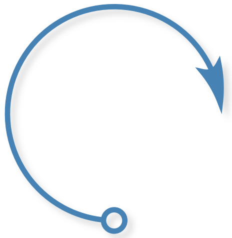
Arrowhead types:



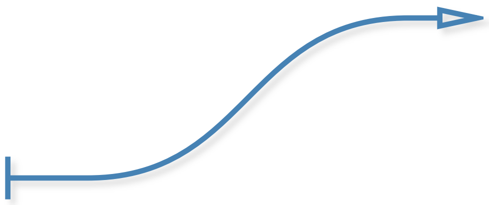
arrowheadBar (-1)
arrowheadPlain (-2)
arrowheadOutline (-3)
arrowheadOutlineIndent (-4)
arrowheadOutlineOutdent (-5)
arrowheadFilled (-6)
arrowheadFilledIndent (-7)
arrowheadFilledOutdent (-8)

Examples:

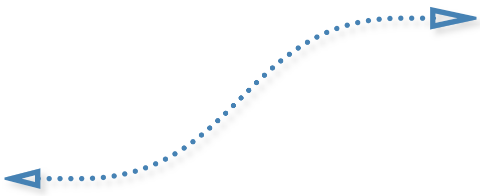
```
OpenDrawing(120;120)
  AddArrowExt(M 60 100 A 100 60 40 40 0 1 1;
              2;                     /* Stroke.        */
              steelBlue;             /* Color.         */
              solid;                 /* Color variant. */
              arrowHeadFilledIndent; /* Head type.     */
              20;                    /* Head size.     */
              0;                     /* Head gap.      */
              circle;                /* Tail type.     */
              8;                     /* Tail size.     */
              0;                     /* Tail gap.      */
              2 2 2;                 /* Shadow effect. */
              lightGray)             /* Shadow color.  */
CloseDrawing()
```



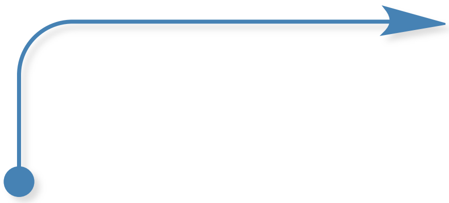
```
OpenDrawing(200;100)
  AddArrowExt(M 10 80 L 40 80 C 100 80 100 20 160 20 L 190 20;
              2;                /* Stroke.        */
              steelBlue;        /* Color.         */
              solid;            /* Color variant. */
              arrowHeadOutline; /* Head type.     */
              20;               /* Head size.     */
              0;                /* Head gap.      */
              arrowHeadBar;     /* Tail type.     */
              16;               /* Tail size.     */
              0;                /* Tail gap.      */
              2 2 2;            /* Shadow effect. */
              lightGray)        /* Shadow color.  */
CloseDrawing()
```



```
OpenDrawing(200;100)
  AddArrowExt(M 10 80 L 40 80 C 100 80 100 20 160 20 L 190 20;
              2 2 2;             /* Stroke.        */
              steelBlue;         /* Color.         */
              solid;             /* Color variant. */
              arrowHeadOutline;  /* Head type.     */
              20;                /* Head size.     */
              0;                 /* Head gap.      */
              arrowHeadOutline;  /* Tail type.     */
              16;                /* Tail size.     */
              0;                 /* Tail gap.      */
              2 2 2;             /* Shadow effect. */
              lightGray)         /* Shadow color.  */
CloseDrawing()
```



```
OpenDrawing(200;100)
  AddArrowExt(M 20 80 L 20 40 A 40 20 20 20 0 0 1 L 180 20;
              1.5;               /* Stroke.        */
              steelBlue;         /* Color.         */
              solid;             /* Color variant. */
              -7;                /* Head type.     */
              25;                /* Head size.     */
              0;                 /* Head gap.      */
              bullet;            /* Tail type.     */
              10;                /* Tail size.     */
              0;                 /* Tail gap.      */
              2 2 2;             /* Shadow effect. */
              lightGray)         /* Shadow color.  */
CloseDrawing()
```



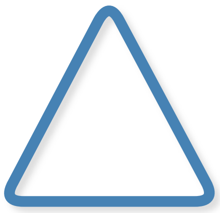
Polygon with Round Corners

```
AddRoundPolygon(scanFlag;coords;roundings;fillColor;fillColorVariant;
                borderStroke;borderColor;borderColorVariant;shadowEffect;shadowColor)
```

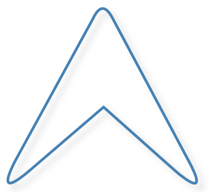
Arguments	*Type*	*Range*	*Default*	*Notes*
scanFlag	int	1..2	xxyy	*See function AddPolygon()*
coords	num[]	0..1000	(required)	*Dimension: [pt]*
roundings	num[]	0..1000	5 %	
fillColor	rgba	0..255	black	
fillColorVariant	int	-1..128	solid	
borderStroke	num[]	0..1000	0	*Dimension: [pt]*
borderColor	rgba	0..255	none	
borderColorVariant	int	-1..128	solid	
shadowEffect	num[]	-1000..1000	0	
shadowColor	rgba	0..255	#888a	

Examples:

```
OpenDrawing(100;100)
  AddRoundPolygon(xyxy;50 10  10 85  90 85  50 10;
             10%;              /* Corner rounding. */
             none;0;           /* Fill.            */
             4;steelBlue;solid; /* Border.         */
             2 2 2;            /* Shadow effect.   */
             lightGray)        /* Shadow color.    */
CloseDrawing()
```

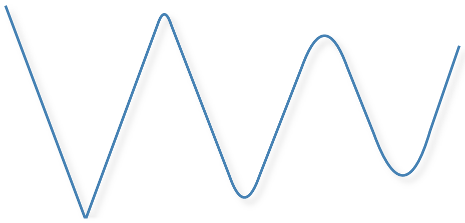


```
OpenDrawing(100;100)
  AddRoundPolygon(xyxy;50 10  10 85  50 50  90 85  50 10;
             10% 20% 0 20%;     /* Corner roundings. */
             none;0;            /* Fill.             */
             1;steelBlue;solid; /* Border.           */
             2 2 2;             /* Shadow effect.    */
             lightGray)         /* Shadow color.     */
CloseDrawing()
```



```
OpenDrawing(100;100)
  AddRoundPolygon(xyxy;
             50.0 10.0
             59.0 37.6
             88.0 37.6
             64.6 54.7
             73.5 82.4
             50.0 65.3
             26.5 82.4
             35.4 54.7
             12.0 37.6
             41.0 37.6
             50.0 10.0;
             5%;                /* Corner rounding. */
             206 222 236;solid; /* Fill.            */
             4;steelBlue;solid; /* Border.          */
             2 2 2;             /* Shadow effect.   */
             lightGray)         /* Shadow color.    */
CloseDrawing()
```

```
OpenDrawing(200;100)
  AddRoundPolygon(xyxy;
            15  20
            45 100
            75  20
           105 100
           135  20
           165 100
           185  35;
           0 5% 10% 15% 20%;  /* Corner roundings. */
           none;0;            /* Fill.             */
           1;steelBlue;solid; /* Border.           */
           2 2 2;             /* Shadow effect.    */
           lightGray)         /* Shadow color.     */
CloseDrawing()
```



Text along Path

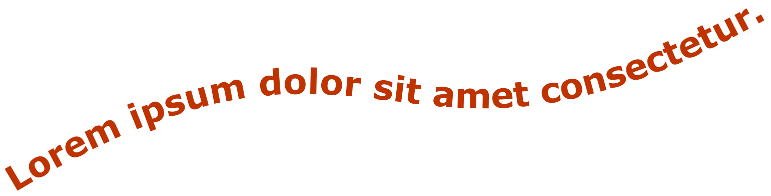
```
AddTextOnPath(pathData;text;fontName;textSize;textStyle;textColor;hAlignment;
              textWidthMax;textHeightMax;ellipsisPosition;shadowEffect;shadowColor)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
pathData	num[]	-inf..+inf	(required)	*See function AddPath()*
text	txt	0..10000	""	*max. 10000 chars.*
fontName	str	0..1000	ApplFont	
textSize	num	0..1000	9	*Dimension: [pt]*
textStyle	int	0..7	0	
textColor	rgba	0..255	black	
hAlignment	int	1..3	left	
textWidthMax	num[]	-1..10000	-1	*Dimension: [pt]*
textHeightMax	num[]	-1..10000	-1	*Dimension: [pt]*
ellipsisPosition	int	0..4	3	
shadowEffect	num[]	-1000..1000	0	
shadowColor	rgba	0..255	#888a	

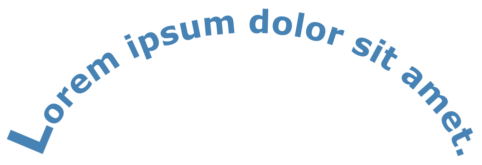
Underline style, gradient color and background color are not supported in combination with function `AddTextOnPath()`.

Examples:

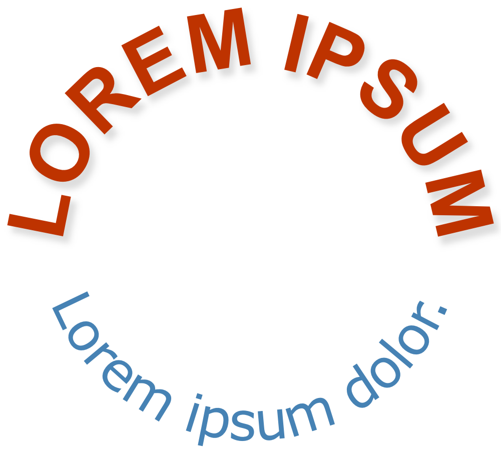
```
OpenDrawing(300;100)
  AddTextOnPath(M 10 85 C 150 0 150 100 292 20;
                "Lorem ipsum dolor sit amet consectetur.";
                Verdana;13;bold;darkRed)
CloseDrawing()
```



```
OpenDrawing(200;80)
  AddTextOnPath(M 20 80 A 180 80 80 60 0 0 1;
                "<span size=20>L</span>orem ipsum dolor sit amet.";
                Verdana;12;bold;steelBlue;center)
CloseDrawing()
```



```
OpenDrawing(200;200)
  AddTextOnPath(M 30 110 A 170 110 70 70 0 1 1;
                "LOREM IPSUM";
                Arial;30;bold;darkRed;center;-1;-1;;2 2 2;lightGray)
  AddTextOnPath(M 20 100 A 180 100 80 80 0 1 0;
                "Lorem ipsum dolor.";
                Verdana;20;plain;steelBlue;center)
CloseDrawing()
```



Symbols along Path

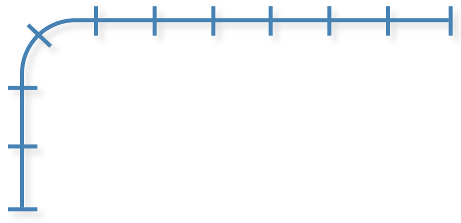
```
AddSymbolsOnPath(pathData;pathStroke;pathColor;pathColorVariant;symbolDistribution;
                symbolType;symbolSize;symbolStroke;symbolColor;symbolColorVariant;
                symbolBackgroundColor;symbolBackgroundColorVariant;shadowEffect;shadowColor)
```

Arguments	*Type*	*Range*	*Default*	*Notes*
pathData	num[]	-inf..+inf	(required)	*See function AddPath()*
pathStroke	num[]	0..1000	1	*Dimension: [pt]*
pathColor	rgba	0..255	black	
pathColorVariant	int	-1..128	solid	
symbolDistribution	num[]	0..1000	1	
symbolType	int	0..126	bullet	
symbolSize	num	0..1000	9	*Dimension: [pt]*
symbolStroke	num[]	0..1000	1	*Dimension: [pt]*
symbolColor	rgba	0..255	black	
symbolColorVariant	int	-1..128	solid	
symbolBackgroundColor	rgba	0..255	none	
symbolBackgroundColorVariant	int	-1..128	solid	
shadowEffect	num[]	-1000..1000	0	
shadowColor	rgba	0..255	#888a	

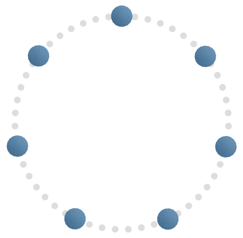
The argument *symbolDistribution* can have up to 4 values:
[1] Distribution unit [1..3]
  1…Total path length (default).
  2…Length between path vertices.
  3…Length between path elements.
[2] Interval length, absolute in pixels or relative in percent of the distribution unit. Default: 25%
[3] Indent length, absolute in pixels or relative in percent of the distribution unit. Default: 0
[4] Orientation, constant in degrees or, if missing, the symbol is rotated automatically to match the direction of the path (default).

Examples:

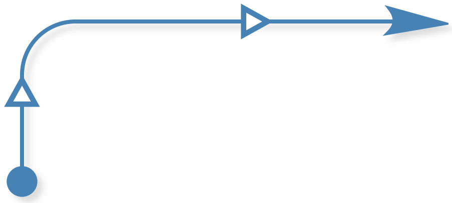
```
OpenDrawing(200;100)
  AddSymbolsOnPath(M 20 90 L 20 40 A 40 20 20 20 0 0 1 L 180 20;
              1.5;         /* Path stroke.                              */
              steelBlue;   /* Path color.                               */
              0;           /* Path color variant.                       */
              1 10% ;      /* 11 symbols evenly distributed along path. */
              barVertical; /* Symbol type.                              */
              11;          /* Symbol size.                              */
              1.5;         /* Symbol stroke.                            */
              steelBlue;   /* Symbol color.                             */
              0;           /* Symbol color variant.                     */
              none;        /* Symbol background color.                  */
              0;           /* Symbol background color variant.          */
              2 2 2;       /* Shadow effect.                            */
              lightGray)   /* Shadow color.                             */
CloseDrawing()
```



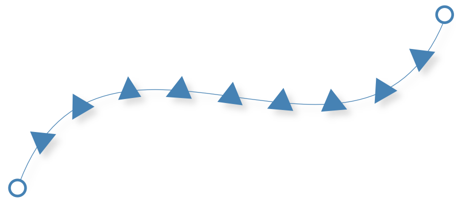
```
OpenDrawing(100;100)
  AddSymbolsOnPath(M 50 10 A 50 90 40 40 0 1 1 A 50 10 40 40 0 1 1;
              2.5 2.5 2.5; /* Path stroke.                             */
              lightGray;   /* Path color.                              */
              0;           /* Path color variant.                      */
              1 14.3%;     /* 7 symbols evenly distributed along path. */
              bullet;      /* Symbol type.                             */
              6;           /* Symbol size.                             */
              2;           /* Symbol stroke.                           */
              steelBlue;   /* Symbol color.                            */
              shaded)      /* Symbol color variant.                    */
CloseDrawing()
```



```
OpenDrawing(200;100)
  AddArrowExt(M 20 80 L 20 40 A 40 20 20 20 0 0 1 L 180 20;
              1.5;                  /* Stroke.        */
              steelBlue;            /* Color.         */
              0;                    /* Color variant. */
              -7;                   /* Head type.     */
              25;                   /* Head size.     */
              0;                    /* Head gap.      */
              bullet;               /* Tail type.     */
              10;                   /* Tail size.     */
              0;                    /* Tail gap.      */
              2 2 2;                /* Shadow effect. */
              lightGray)            /* Shadow color.  */
  AddSymbolsOnPath(M 20 80 L 20 40 A 40 20 20 20 0 0 1 L 180 20;
              0;none;0;             /* Hide path.                 */
              1 50% 15%;            /* 2 symbols with 15% indent. */
              triangleRightOutline; /* Symbol type.               */
              12;                   /* Symbol size.               */
              2;                    /* Symbol stroke.             */
              steelBlue;            /* Symbol color.              */
              0;                    /* Symbol color variant.      */
              white)                /* Symbol background color.   */
CloseDrawing()
```



```
OpenDrawing(200;100)
  AddSymbolsOnPath(M 20 85 C  50 0 150 100 180 20;
              0.25;              /* Path stroke.                           */
              steelBlue;         /* Path color.                            */
              0;                 /* Path color variant.                    */
              1 10% 10.5%;       /* Symbols evenly distributed along path. */
              triangleUp;        /* Symbol type.                           */
              9;                 /* Symbol size.                           */
              1;                 /* Symbol stroke.                         */
              steelBlue;         /* Symbol color.                          */
              0;                 /* Symbol color variant.                  */
              white;             /* Symbol background color.               */
              0;                 /* Symbol background color variant.       */
              2 2 2;             /* Shadow effect.                         */
              lightGray)         /* Shadow color.                          */
  AddSymbolsOnPath(M 20 85 C  50 0 150 100 180 20;
              0;none;0;          /* Hide path.                        */
              1 100% ;           /* Symbols on start and end of path. */
              circle;            /* Symbol type.                      */
              6;                 /* Symbol size.                      */
              1;                 /* Symbol stroke.                    */
              steelBlue;         /* Symbol color.                     */
              0;                 /* Symbol color variant.             */
              white)             /* Symbol background color.          */
CloseDrawing()
```



Backgrounds

For all elements with an optional background, e.g. labels, title, legend, groups, views, tags and tables a new *background options* function is available. The following functions make it possible to define round corners and paddings:

BackgroundOptions(*cornerRadius;padding*)
ChartBackgroundOptions(*cornerRadius;padding*)
AxisLabelBackgroundOptions(*axisIndex;cornerRadius;padding*)
AxisMajorTickLabelBackgroundOptions(*axisIndex;cornerRadius;padding*)
AxisMinorTickLabelBackgroundOptions(*axisIndex;cornerRadius;padding*)
LabelBackgroundOptions(*seriesIndex;cornerRadius;padding*)
LegendBackgroundOptions(*cornerRadius;padding*)
TitleBackgroundOptions(*cornerRadius;padding*)
PieChartCenterLabelBackgroundOptions(*cornerRadius;padding*)
PieChartInnerLabelBackgroundOptions(*cornerRadius;padding*)

The first argument *cornerRadius* can have up to four corner radii (topLeft, topRight, bottomRight and bottomLeft). As the 2nd argument *padding*, it is possible to define the padding for each side of an element (top, right, bottom and left). Refer to the mnemonic “TRouBLe” as an easy way to remember the order of paddings. The corner radii and paddings can be entered absolute in pixels or relative as percentage of the diagonal of the drawing area.

Notes:

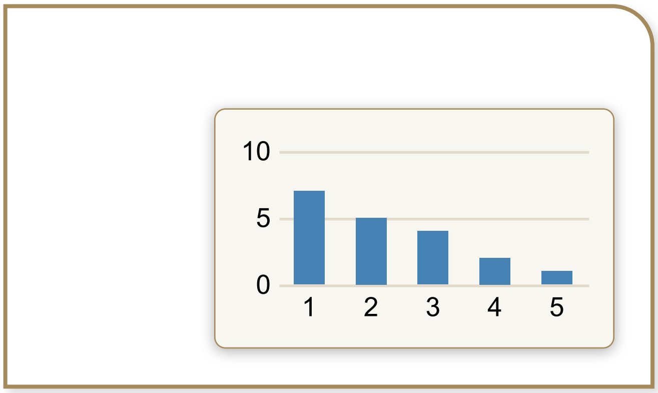
- Currently, the function ChartBackgroundOptions() is only available for circular bar charts, linear meters, linear gauges and radial gauges.
- Paddings are ignored in combination with views: *OpenView()…CloseView()*.
- Background option functions are only enabled in combination with the corresponding background function. For example:

  ```
  ...
  /* Chart legend. */
  LegendTitle("Legend")
  LegendTexts("Item 1";"Item 2";"Item 3")
  LegendStyle(Verdana;10;plain)
  LegendOptions(bottomCenter)
  LegendBackgroundOptions(10;10) /* No effect, function LegendBackground() is missed. */
  ...
  ```

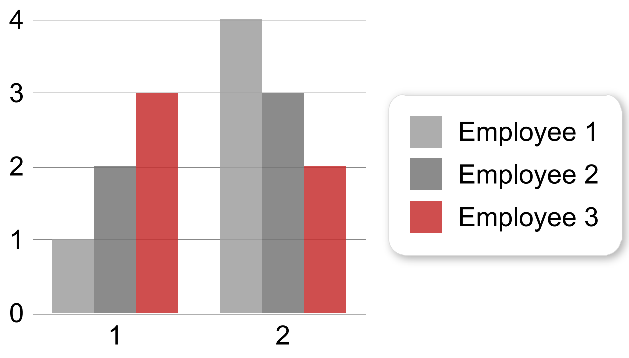
  ```
  ...
  /* Chart legend. */
  LegendTitle("Legend")
  LegendTexts("Item 1";"Item 2";"Item 3")
  LegendStyle(Verdana;10;plain)
  LegendOptions(bottomCenter)
  LegendBackground(#f3f6f2;0;0;;;2 2 2;lightGray)
  LegendBackgroundOptions(10;10) /* Correct, function LegendBackground() is called. */
  ...
  ```

Examples:

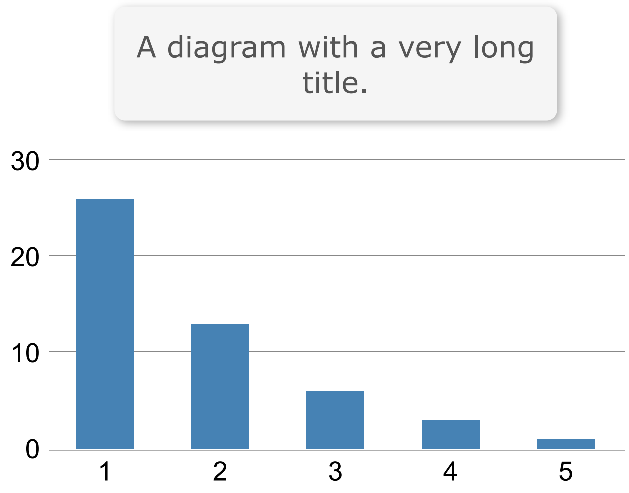
```
OpenDrawing(250;150)
  OpenView(80;40;150;90)
    ChartData(7 5 4 2 1)
    BarChart()
    /* Set up styles. */
    FillStyle(1;#4682b4)
    BorderStyle(all;none)
    /* Set up axes. */
    AxisLine(all;0)
    AxisMajorTicks(all;0)
    /* Set up grid and background. */
    MajorGridLineWidths(y;x;0)
    MajorGridLineColors(x;y;#e1d9c9)
    Background(#f7f6f1;solid;0.5;#a58b5c;solid;0 1 5)
    BackgroundOptions(4)
  CloseView()
  Background(white;solid;1.5;#a58b5c;;2 2 3;lightGray)
  BackgroundOptions(0 15 0 0) /* Border with a top right rounded corner. */
CloseDrawing()
```



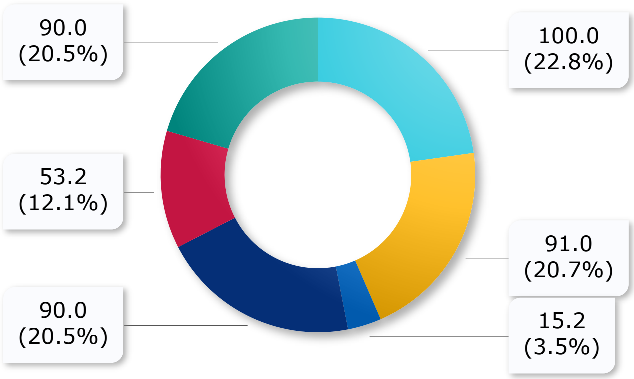
```
OpenDrawing(250;150)
  ChartData(1 4; 2 3; 3 2)
  BarChart()
  /* Set up styles. */
  FillStyle(1;160 160 160 220)
  FillStyle(2;120 120 120 220)
  FillStyle(3;200  50  50 220)
  BorderStyle(all;none)
  /* Set up axes. */
  AxisLine(all;0)
  AxisMajorTicks(all;0)
  /* Set up legend. */
  LegendTexts("Employee 1";"Employee 2";"Employee 3")
  LegendBackground(white;;0.25;lightGray;;1 1 3)
  LegendBackgroundOptions(6;8)
  /* Set up grid. */
  MajorGridLineWidths(x;y;0.25)
  MajorGridLineWidths(y;x;0)
CloseDrawing()
```



```
OpenDrawing(250;200)
  ChartData(26 13 6 3 1)
  BarChart()
  /* Set up styles. */
  FillStyle(1;#4682b4)
  BorderStyle(all;none)
  /* Set up axes. */
  AxisLine(all;0)
  AxisMajorTicks(all;0)
  /* Set up title. */
  TitleText("A diagram with a very long title.")
  TitleStyle(Verdana;11;plain;#555;;;150) /* Limit title width to 150 pixels. */
  TitleBackground(whiteSmoke;;0.25;none;;1 1 3)
  TitleBackgroundOptions(4;8)
  /* Set up grid. */
  MajorGridLineWidths(x;y;0.25)
  MajorGridLineWidths(y;x;0)
CloseDrawing()
```



```
OpenDrawing(260;230)
  OpenChart(60;50;140;120;on)
    ChartData(100 91 15.2 90 53.2 89.97)
    PieChart(label+shadow;;60)
    PieChartAuxLines(20;;center;0.25;#555) /* After PieChart() */
    /* Set up styles. */
    FillStyle(1;#00bed7;shaded)
    FillStyle(2;#ffb400;shaded)
    FillStyle(3;#0073dd;shaded)
    FillStyle(4;#063e9e;shaded)
    FillStyle(5;#f81a53;shaded)
    FillStyle(6;#00a79d;shaded)
    FillStyle(7;#37cebf;shaded)
    BorderStyle(all;none)
    ShadowStyle(all;2 2 3)
    LabelTexts(;"|f1|\n(|f1|%)")
    LabelStyle(;Verdana;8;plain;black;center)
    LabelBackground(1;#fafbfe;;1;#fafbfe;;0.5 0.5 2)
    LabelBackgroundOptions(all;4 0;3 5 5 5)
  CloseChart()
CloseDrawing()
```



Symbol Collection

A total of 126 symbols organized in 8 categories are available in xmCHART 5.

- Spokes
- Stars
- Triangles
- Bars
- Circles
- Squares
- Diamonds
- Arrowheads

NO SYMBOL

```
none (0)
```

SPOKES

```
✕ cross (1)
+ plus (7)
⅄ spokes3 (25)
+ spokes4 (26) /* Same as plus (7) */
✱ spokes5 (27)
✻ spokes6 (28)
✼ spokes7 (29)
✳ spokes8 (30)
```

STARS

```
▲ star3 (19)
✦ star4 (20)
★ star5 (21)
✶ star6 (22)
✹ star7 (23)
✸ star8 (24)
```

TRIANGLES

- ▼ `triangleDown (5)`
- ▽ `triangleDownOutline (11)`
- ▽ `triangleDownHalfLeft (31)`
- ▽ `triangleDownHalfRight (32)`
- ▲ `triangleUp (6)`
- △ `triangleUpOutline (12)`
- ◭ `triangleUpHalfLeft (33)`
- ◮ `triangleUpHalfRight (34)`
- ◀ `triangleLeft (35)`
- ◁ `triangleLeftOutline (36)`
- ◁ `triangleLeftHalfUp (37)`
- ◁ `triangleLeftHalfDown (38)`
- ▶ `triangleRight (39)`
- ▷ `triangleRightOutline (40)`
- ▷ `triangleRightHalfUp (41)`
- ▷ `triangleRightHalfDown (42)`

BARS

- — `barHorizontal (13)`
- | `barVertical (14)`
- ╴ `barLeft (15)`
- ╶ `barRight (16)`
- ╵ `barTop (17)`
- ╷ `barBottom (18)`

ARROWHEADS Only available for function `AddArrowExt()`

- ⊢—┤ `arrowheadBar (-1)`
- → `arrowheadOpen (-2)`
- ⇾ `arrowheadOutline (-3)`
- ⇾ `arrowheadOutlineIndent (-4)`
- ⇾ `arrowheadOutlineOutdent (-5)`
- ➔ `arrowheadFilled (-6)`
- ➤ `arrowheadFilledIndent (-7)`
- ➔ `arrowheadFilledOutdent (-8)`

CIRCLES

bullet (2)
circle (8)
circleHalfTop (43)
circleHalfRight (44)
circleHalfBottom (45)
circleHalfLeft (46)
circleHalfLeftTop (47)
circleHalfRightTop (48)
circleHalfRightBottom (49)
circleHalfLeftBottom (50)
circleQuarterTop (51)
circleQuarterRight (52)
circleQuarterBottom (53)
circleQuarterLeft (54)
circleQuarterLeftTop (55)
circleQuarterRightTop (56)
circleQuarterRightBottom (57)
circleQuarterLeftBottom (58)
circleTwoQuarterLeftRight (59)
circleTwoQuarterTopBottom (60)
circleTwoQuarterLeftTopRightBottom (61)
circleTwoQuarterRightTopLeftBottom (62)
circleThreeQuarterTop (63)
circleThreeQuarterRight (64)
circleThreeQuarterBottom (65)
circleThreeQuarterLeft (66)
circleThreeQuarterLeftTop (67)
circleThreeQuarterRightTop (68)
circleThreeQuarterRightBottom (69)
circleThreeQuarterLeftBottom (70)

SQUARES

- square (3)
- squareOutline (9)
- squareHalfTop (71)
- squareHalfRight (72)
- squareHalfBottom (73)
- squareHalfLeft (74)
- squareHalfLeftTop (75)
- squareHalfRightTop (76)
- squareHalfRightBottom (77)
- squareHalfLeftBottom (78)
- squareQuarterTop (79)
- squareQuarterRight (80)
- squareQuarterBottom (81)
- squareQuarterLeft (82)
- squareQuarterLeftTop (83)
- squareQuarterRightTop (84)
- squareQuarterRightBottom (85)
- squareQuarterLeftBottom (86)
- squareTwoQuarterLeftRight (87)
- squareTwoQuarterTopBottom (88)
- squareTwoQuarterLeftTopRightBottom (89)
- squareTwoQuarterRightTopLeftBottom (90)
- squareThreeQuarterTop (91)
- squareThreeQuarterRight (92)
- squareThreeQuarterBottom (93)
- squareThreeQuarterLeft (94)
- squareThreeQuarterLeftTop (95)
- squareThreeQuarterRightTop (96)
- squareThreeQuarterRightBottom (97)
- squareThreeQuarterLeftBottom (98)

DIAMONDS

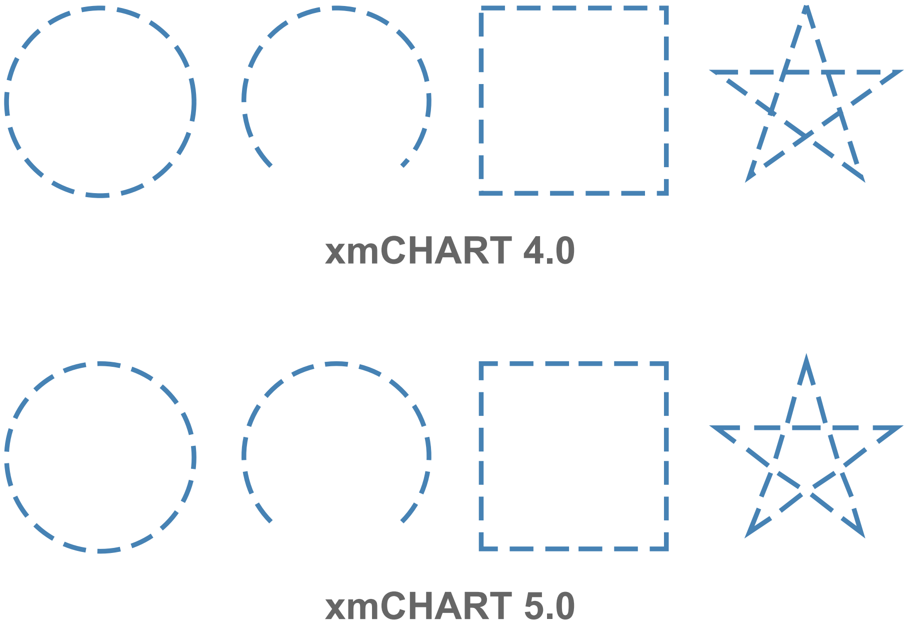
- ◆ `diamond (4)`
- ◇ `diamondOutline (10)`
- ⬘ `diamondHalfTop (99)`
- ⬗ `diamondHalfRight (100)`
- ⬙ `diamondHalfBottom (101)`
- ⬖ `diamondHalfLeft (102)`
- ◈ `diamondHalfLeftTop (103)`
- ◈ `diamondHalfRightTop (104)`
- ◈ `diamondHalfRightBottom (105)`
- ◈ `diamondHalfLeftBottom (106)`
- ◈ `diamondQuarterTop (107)`
- ◈ `diamondQuarterRight (108)`
- ◈ `diamondQuarterBottom (109)`
- ◈ `diamondQuarterLeft (110)`
- ◈ `diamondQuarterLeftTop (111)`
- ◈ `diamondQuarterRightTop (112)`
- ◈ `diamondQuarterRightBottom (113)`
- ◈ `diamondQuarterLeftBottom (114)`
- ◈ `diamondTwoQuarterLeftRight (115)`
- ◈ `diamondTwoQuarterTopBottom (116)`
- ◈ `diamondTwoQuarterLeftTopRightBottom (117)`
- ◈ `diamondTwoQuarterRightTopLeftBottom (118)`
- ◈ `diamondThreeQuarterTop (119)`
- ◈ `diamondThreeQuarterRight (120)`
- ◈ `diamondThreeQuarterBottom (121)`
- ◈ `diamondThreeQuarterLeft (122)`
- ◈ `diamondThreeQuarterLeftTop (123)`
- ◈ `diamondThreeQuarterRightTop (124)`
- ◈ `diamondThreeQuarterRightBottom (125)`
- ◈ `diamondThreeQuarterLeftBottom (126)`

Improved Line Styles

Dashed lines have been optimized in xmCHART 5, so they now fit perfectly in corner points and at line ends.

Example:

```
OpenDrawing(400;130)
  AddOval(12;12;81;81;none;0;2 10 5;steelBlue)
  AddArc(112;12;80;80;-135;270;2 10 5;steelBlue)
  AddRect(212;12;80;80;none;0;2 10 5;steelBlue)
  AddPolygon(xyxy;
             350.0 12.0
             373.5 84.4
             312.0 39.6
             388.0 39.6
             326.5 84.4
             350.0 12.0;
             none;0;2 10 5;steelBlue)
  AddText(200;115;"xmCHART 5.0";Arial;16;bold;#666;center;center;0;400)
CloseDrawing()
```

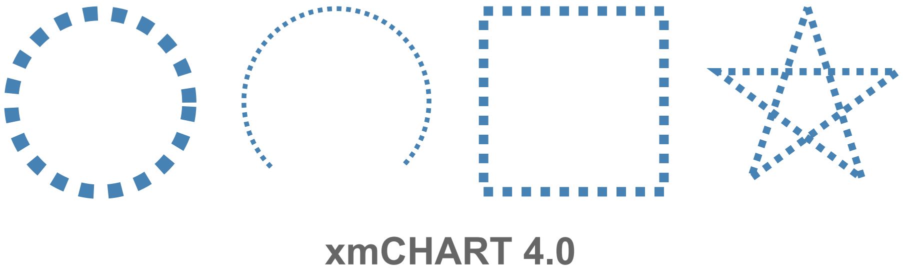


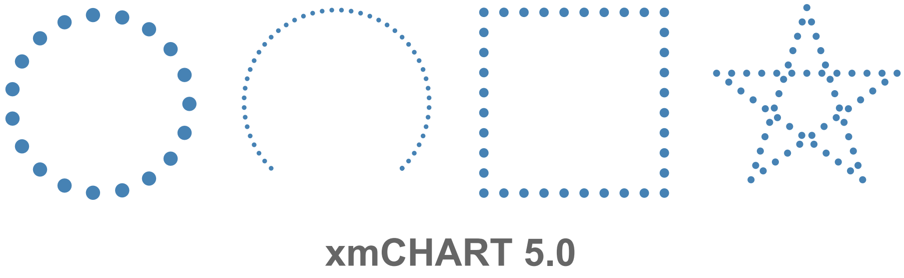
Dotted Lines

To create a dotted line with perfect round dots the line thickness and the dash and gap lengths must be identical. For example: *LineStyle(1;poly;2 2 2;red)*

Example:

```
OpenDrawing(400;130)
  AddOval(12;12;81;81;none;0;6 6 6;steelBlue)
  AddArc(112;12;80;80;-135;270;2 2 2;steelBlue)
  AddRect(212;12;80;80;none;0;4 4 4;steelBlue)
  AddPolygon(xyxy;
             350.0 12.0
             373.5 84.4
             312.0 39.6
             388.0 39.6
             326.5 84.4
             350.0 12.0;
             none;0;3 3 3;steelBlue)
  AddText(200;115;"xmCHART 5.0";Arial;16;bold;#666;center;center;0;400)
CloseDrawing()
```





Please note, dotted line styles are not supported in combination with contour plots.

Custom Color Schemes and Color Scales

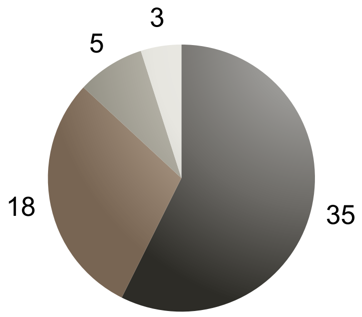
A custom color scheme can be defined by adding a list of colors as the first argument in one of the following color scheme functions:

```
BorderColorScheme()
FillColorScheme()
LineColorScheme()
SymbolColorScheme()
```

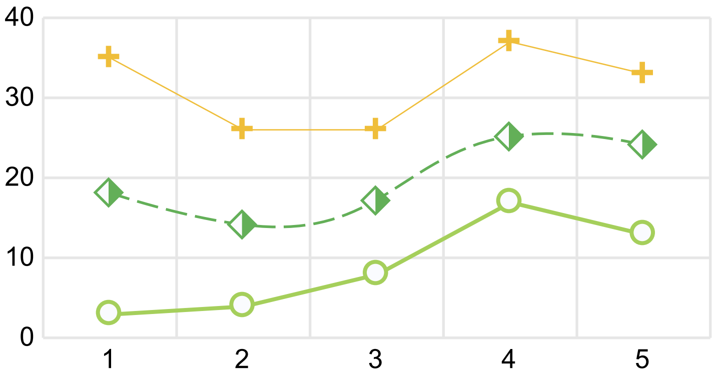
Colors can be entered as red, green and blue components between 0 and 255 or as hexcodes – always without an alpha value.

Examples:

```
OpenDrawing(150;140)
  OpenChart(25;25;100;100;on)
    ChartData(35 18 5 3)
    PieChart(label;0; 0)
    FillColorScheme(#3b3933 #967e67 #a3a092 #e2e1da #5a6a75 #607e65;shaded)
    BorderStyle(all;none)
  CloseChart()
CloseDrawing()
```



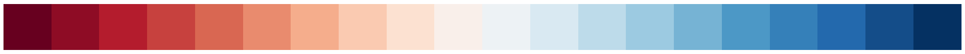
```
OpenDrawing(320;180)
  OpenChart(40;25;250;120;on)
    ChartData( 3  4 8  17 13;
              18 14 17 25 24;
              35 26 26 37 33)
    LineChart(symbol;on)
    LineColorScheme(165 207 91
                     99 175 88
                    239 190 58) /* Before LineStyle() function. */
    SymbolColorScheme(165 207 91
                       99 175 88
                      239 190 58) /* Before SymbolStyle() function. */
    LineStyle(1;poly;1.5)
    LineStyle(2;smooth;1 8 3)
    LineStyle(3;poly;0.5)
    SymbolStyle(1;circle;8;1.5;;;#fafbfe)
    SymbolStyle(2;diamondHalfRight;9;;;;#fafbfe)
    SymbolStyle(3;spokes4;8;2)
    AxisLine(all;0)       /* Hide axes. */
    AxisMajorTicks(all;0) /* Hide tick marks. */
    MajorGridLineColors(all;all;230 230 230)
  CloseChart()
CloseDrawing()
```



A custom color scale can be defined by adding a list of color stops as the second argument in function `FillColorScale()`. A color stop is represented by 4 elements: offset, red, green and blue. Offsets are between 0 and 1, the 3 RGB components between 0 and 255 or, as an option, as hexcodes. Colors must be entered without an alpha value.

Example:

```
OpenDrawing(360;18)
  OpenChart(0;0;360;18;on)
    ChartData(1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
    Heatmap(;1) /* 1 row. */
    FillColorScale(1; /*  Series index. */
                   0   #67001f
                   0.1 #b2182b
                   0.2 #d6604d
                   0.3 #f4a582
                   0.4 #fddbc7
                   0.5 #f7f7f7
                   0.6 #d1e5f0
                   0.7 #92c5de
                   0.8 #4393c3
                   0.9 #2166ac
                   1   #053061)
    BorderStyle(1;none) /* Hide borders. */
  CloseChart()
CloseDrawing()
```



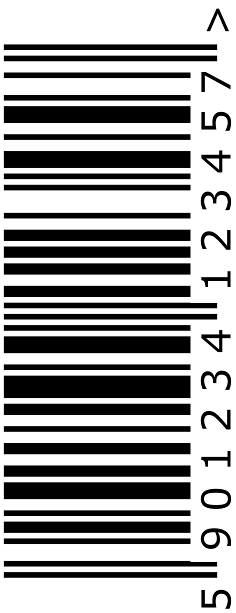
Rotated Views

As an option, a new argument *rotation* has been added to function `OpenView()`.

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

Example:

```
OpenDrawing(140;270;print)
  OpenView(-50;80;240;110;-90)
    BarcodeEAN(20;10;200;80;"5901234123457")
    AddText(7;95;"5";Verdana;15)
    AddRect(26;80;90;20;white)
    AddText(30;95;"9 0 1 2 3 4";Verdana;15)
    AddRect(123;80;90;20;white)
    AddText(127;95;"1 2 3 4 5 7";Verdana;15)
    AddText(223;95;">";Verdana;15)
    // Background() /* Uncomment while positioning the barcode. */
  CloseView()
  // Background() /* Uncomment while positioning the barcode. */
CloseDrawing()
```



Styled Texts

In xmCHART 5 the style of any portion of text inside a text element can be changed by embedding that portion inside a *<span>* tag or other simple HTML tag from the following list:

"...<b>bold text</b>..."
"...<i>italic text</i>..."
"...<u>underline text</u>..."

<span> Tag Attributes:
The following five <span> tag attributes are available:

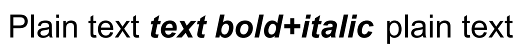
font, size, shiftY, color, backgroundColor

Please note that attribute values containing white space must be placed in single quotes.
For example:

```
AddText(10;10;"...<span font='Times New Roman'>Some text</span>...")
AddText(10;10;"...<span font=Verdana>Some text</span>...")
AddText(10;10;"...<span color='rgb255 (100,150,0)'>Some text</span>...")
AddText(10;10;"...<span color=rgba255(100,150,0,200)>Some text</span>...")
```

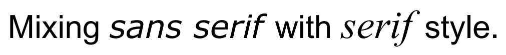
Examples:

```
OpenDrawing(300;30)
  AddText(10;20;"Plain text <b><i>text bold+italic</i></b> plain text";Arial;12)
CloseDrawing()
```



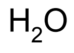
Plain text ***text bold+italic*** plain text

```
OpenDrawing(200;30)
  AddText(10;20;"Mixing <span font=Verdana><i>sans serif</i></span> with <span
font='Times New Roman' size=15><i>serif</i></span> style.";Arial;12)
CloseDrawing()
```



Mixing *sans serif* with *serif* style.

```
OpenDrawing(300;30)
  AddText(10;20;"H<span size=8 shiftY=4>2</span>O";Arial;12) /* H₂O */
CloseDrawing()
```

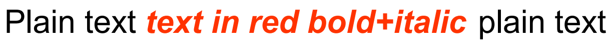


H_2O

```
OpenDrawing(300;30)
  AddText(10;20;"x<span size=8 shiftY=-4>3</span>";Arial;12) /* x³ */
CloseDrawing()
```

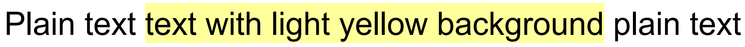
x^3

```
OpenDrawing(300;30)
  AddText(10;20;"Plain text <span color=red><b><i>text in red bold+italic</i></b></span>
plain text";Arial;12)
CloseDrawing()
```



Plain text ***text in red bold+italic*** plain text

```
OpenDrawing(300;30)
  AddText(10;20;"Plain text <span backgroundColor=rgba255(255,255,0,100)>text with light
yellow background</span> plain text";Arial;12)
CloseDrawing()
```



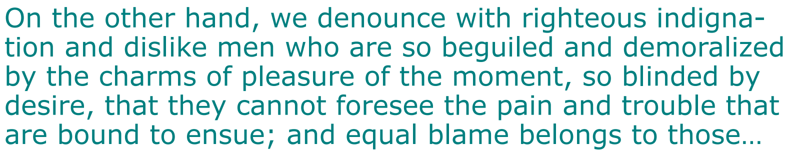
Plain text text with light yellow background plain text

Custom Line Spacing

The line height of a text can be controlled by adding a line increment factor to argument *textHeightMax*. The argument *textHeightMax* is available in all text-related functions such as AddText(), LabelStyle(), TitleStyle(), LegendStyle(), AxisLabelStyle(), TagText() etc. A factor < 0 squeezes the text together, a factor > 0 stretches the lines. The factor can be entered absolute in pixels or relative in % of the line height.

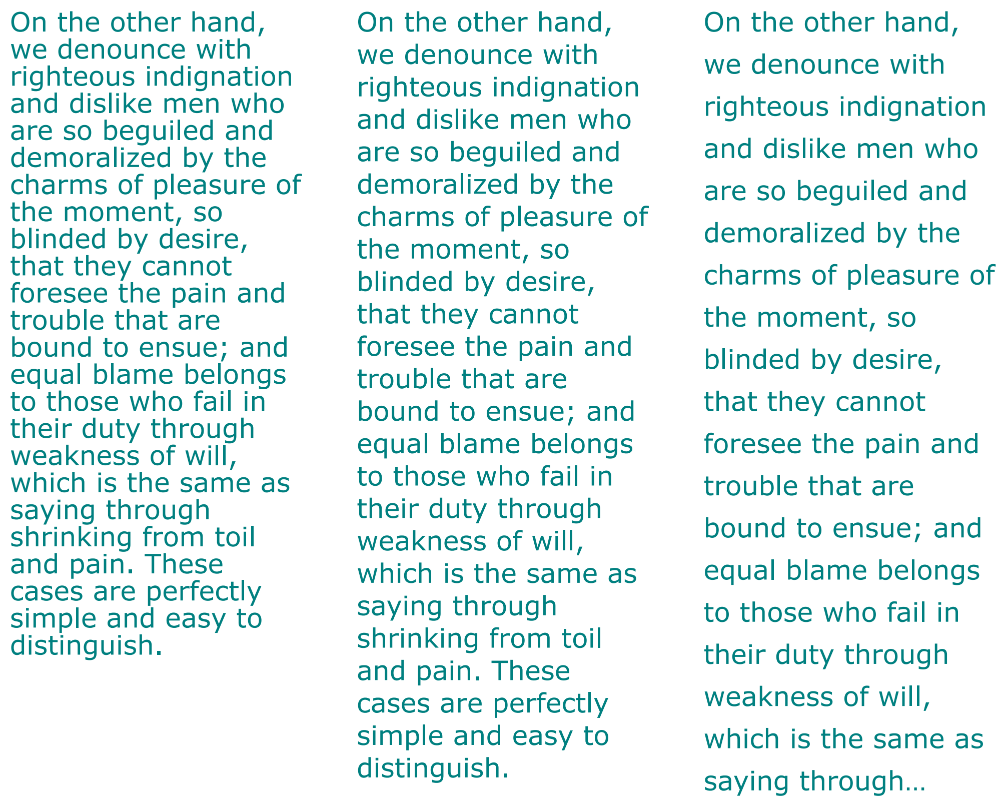
Examples:

```
OpenDrawing(320;60)
  AddText(10;12;"On the other hand, we denounce with righteous indignation and dislike men
who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by
desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal
blame belongs to those who fail in their duty through weakness of will, which is the same as
saying through shrinking from toil and pain. These cases are perfectly simple and easy to
distinguish.";
  Verdana;10;plain;teal;;;0;300;60 -10%) /* Reduce line height by 10%. */
CloseDrawing()
```



On the other hand, we denounce with righteous indigna-
tion and dislike men who are so beguiled and demoralized
by the charms of pleasure of the moment, so blinded by
desire, that they cannot foresee the pain and trouble that
are bound to ensue; and equal blame belongs to those…

```
OpenDrawing(390;310)
  AddText(10;20;"On the other hand, we denounce with righteous indignation and dislike men
who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by
desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal
blame belongs to those who fail in their duty through weakness of will, which is the same as
saying through shrinking from toil and pain. These cases are perfectly simple and easy to
distinguish.";
  Verdana;10;plain;teal;;;0;110;-1 -2) /* -2...Line height reduced by 2 pixels. */
  AddText(140;20;"On the other hand, we denounce with righteous indignation and dislike men
who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by
desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal
blame belongs to those who fail in their duty through weakness of will, which is the same as
saying through shrinking from toil and pain. These cases are perfectly simple and easy to
distinguish.";
  Verdana;10;plain;teal;;;0;110;-1 0) /* 0...Regular line height (default). */
  AddText(270;20;"On the other hand, we denounce with righteous indignation and dislike men
who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by
desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal
blame belongs to those who fail in their duty through weakness of will, which is the same as
saying through shrinking from toil and pain. These cases are perfectly simple and easy to
distinguish.";
  Verdana;10;plain;teal;left;;0;110;300 30%) /* 30%...Line height increased by 30%. */
CloseDrawing()
```



```
On the other hand,
we denounce with
righteous indignation
and dislike men who
are so beguiled and
demoralized by the
charms of pleasure of
the moment, so
blinded by desire,
that they cannot
foresee the pain and
trouble that are
bound to ensue; and
equal blame belongs
to those who fail in
their duty through
weakness of will,
which is the same as
saying through
shrinking from toil
and pain. These
cases are perfectly
simple and easy to
distinguish.
```

```
On the other hand,
we denounce with
righteous indignation
and dislike men who
are so beguiled and
demoralized by the
charms of pleasure of
the moment, so
blinded by desire,
that they cannot
foresee the pain and
trouble that are
bound to ensue; and
equal blame belongs
to those who fail in
their duty through
weakness of will,
which is the same as
saying through
shrinking from toil
and pain. These
cases are perfectly
simple and easy to
distinguish.
```

```
On the other hand,
we denounce with
righteous indignation
and dislike men who
are so beguiled and
demoralized by the
charms of pleasure of
the moment, so
blinded by desire,
that they cannot
foresee the pain and
trouble that are
bound to ensue; and
equal blame belongs
to those who fail in
their duty through
weakness of will,
which is the same as
saying through…
```

Text Hyphenation

The hyphenation of text can be controlled by the function `HyphenationOptions()`.

HyphenationOptions(lineBreakFlag;hyphenationExceptions)

Arguments	*Type*	*Range*	*Default*	*Notes*
lineBreakFlag	int	-1..13	1	
hyphenationExceptions	str	0..100000	""	

The first argument *LineBreakFlag* can be used to define where to insert line breaks. The following flags are available:

-1	Character based line breaks. No hyphenation.
0	Word based line breaks. No hyphenation.
1	English-US hyphen rules (default).
2	English-GB hyphen rules.
3	German hyphen rules.
4	Spanish hyphen rules.
5	French hyphen rules.
6	Italian hyphen rules.
7	Greek hyphen rules.
8	Dutch hyphen rules.
9	Danish hyphen rules.
10	Swedish hyphen rules.
11	Norwegian hyphen rules.
12	Finnish hyphen rules.
13	Portuguese hyphen rules.

A list of hyphenation exceptions can be added as a second argument. The words are separated by spaces, tabs or line feeds. For example:

HyphenationOptions(3;"derby-ter-min wach-stu-be microsoft")

The new xmCHART 5 text engine also supports soft hyphens (Unicode: U+00AD, UFT-8 encoded: \xC2\xAD). A soft hyphen indicates a hyphenation point, where a line break is preferred when a word is to be hyphenated. For example:

AddText(10;20;"Lorem ipsum dolor sit am\xC2\xADDet, conse\xC2\xADctetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua";Verdana;12;plain;black;;;0;80)

The function HyphenationOptions() works globally and affects all texts in a drawing. However, from time to time, it can be advantageous to override the global hyphenation settings for individual texts. For instance, you may want to disable automatic hyphenation for a diagram title or you have texts in different languages.
The *lineBreakFlag* can be set individually for each text function by adding the line break flag to argument *textWidthMax*. The argument *textWidthMax* is available in all text-related functions such as AddText(), LabelStyle(), TitleStyle(), LegendStyle(), AxisLabelStyle(), TagText() etc. For example:

```
OpenDrawing(800;600)
   HyphenationOptions(3) /* Use German hyphen rules. */
   ...
   further xmCHART functions
   ...
   TitleStyle(Verdana;14;bold;black;center;0;200 0) /* 0...Turn off hyphenation. */
   ...
   AddText(100;100;"Uno dei fatti piu notevoli al principio del decimosesto secolo e
senza dubbio l'apparire della cortigiana";
   Verdana;10;plain;black;left;baseline;0;100 6) /* 6...Italian hyphen rules. */
   ...
CloseDrawing()
```

Miscellaneous

- Improved `AddArrow()` function:
  - Pixel-perfect arrow tips.
  - Hollow arrowheads are now drawn correctly without line through.
  - Arrows with dashed line style are drawn correctly with solid (not dashed) arrow tips.

- Optional border stroke and shadow effect added to functions:

```
AddPicture(left;top;width;height;sourceType;sourceData;
           borderStroke;borderColor;borderColorVariant;
           shadowEffect;shadowColor)
BackgroundPict(sourceType;sourceData;
               borderStroke;borderColor;borderColorVariant;
               shadowEffect;shadowColor)
ChartBackgroundPict(planeIndex;sourceType;sourceData;
                    borderStroke;borderColor;borderColorVariant;
                    shadowEffect;shadowColor)
```

- Five new color scales (21 to 25) have been added in xmCHART 5.
- Color scales can be reversed by negating the argument *colorScaleID* to *-colorScaleID*. For example: *FillColorScale(1;-17;shaded)*
- Three new arguments *referenceValueMin, referenceValueMax* and *referenceValueMid* have been added to function `FillColorScale()`. The arguments *referenceValueMin* and *referenceValueMax* are useful for comparing diagrams. By defining a common reference min and max value the color scale is identical for all diagrams and does not depend on the min/max value range of each individual diagram. The middle of the color scale can be defined by using the argument *referenceValueMid*. By default the middle is the arithmetic average of the min and max value.

- `TreeMap()`: Improved labels. Labels now can show absolute and relative (percent) values, e.g. `LabelTexts("|u|\n(|f1|%)")`

- Optimized script parser with:
  - Faster parsing of xmCHART scripts.
  - Precise location of an erroneous value within a value list.
  - In case of an "Out of range" error the bounds of the valid range are returned, e.g. *[0..100]*

- Better support of xmCHART within FileMaker Pro:
  - Support of Script Steps added.
  - Descriptions and online help (by clicking the tiny “?” button) added.

Incompatibilities

- `BackgroundPict()`, `ChartBackgroundPict()`, `AddPicture()`:
  The arguments *location*, *adjustment* and *isProportional* available in the obsolete xmCHART versions 2 and 3 have been replaced by the attributes *borderStroke*, *borderColor* and *borderColorVariant*.

- The timestamp entry format option "date time", i.e timestamps are enclosed by quotes and separated by whitespace, have been deprecated in xmCHART 4 and removed entirely in xmCHART 5. Instead, the date and time are joined together by an ampersand (&) – and not enclosed by quotes. For example:

```
ChartData("2024-01-01 11:22" "2024-03-22 18:33:12") /* Error in xmCHART 5+ */
ChartData(2024-01-01&11:22 2024-03-22&18:33:12)     /* Correct in xmCHART 5+ */
```

- All symbol names available in xmCHART 4 are compatible with xmCHART 5, however the symbol number IDs in xmCHART 4 [0..58] are only compatible up to number ID 34 [0..34].

- In xmCHART 5 the default fill rule for paths changed from *even-odd* to the *non-zero* winding rule.

Deprecations

Deprecations in xmCHART 5

- Numbers as path commands are deprecated, instead use command letters, see `AddPath()`.

- The fileFlag constant *throwError* has been deprecated in xmCHART 4 and removed entirely in xmCHART 5.

- `xmCH_GetVersion()`: A number as function argument is deprecated.
  `xmCH_GetVersion( 1 )`, use `xmCH_GetVersion("long")` instead.
  `xmCH_GetVersion( 2 )`, use `xmCH_GetVersion("short")` instead.
  `xmCH_GetVersion( 3 )`, use `xmCH_GetVersion("norm")` instead.
  `xmCH_GetVersion( 4 )` or `xmCH_GetVersion("platform")` has been deprecated, instead use FileMaker's built-in function `Get(SystemPlatform)`.
  `xmCH_GetVersion("autoupdate")` has been deprecated, instead use argument `"norm"`.
  `xmCH_GetVersion("norm")` returns the normalized version info as an 8-character string in the format:Major version (2 characters)
  Minor version (2 characters)
  Patch version (2 characters)
  Build number (2 characters)
  For example: For v5.0.11 `xmCH_GetVersion("norm")` returns “05001100”.

- xmCH_SetLanguage() has been deprecated.

- The external function `xmCH_AutoUpdate()` has been removed (obsolete).

Deprecations in xmCHART 4

- `AddEllipse()`, use `AddOval()` instead.
- `AddFrame()`, use `AddRect()` instead.
- `AddPolyline()`, use `AddPolygon()` instead.
- `AddRoundFrame()`, use `AddRoundRect()` instead.
- `AddSmoothPolyline()`, use `AddSmoothPolygon()` instead.

- `OpenDrawing(`*width;height;format;antialiasing;targetName*`)`
  The arguments *antialiasing* and *targetName* are ignored in xmCHART 4 or higher.
  targetName has been replaced by the 2nd parameter *fileName* of the external function `xmCH_DrawChart()`.

- The last 3 arguments *location*, *adjustment* and *isProportional* are ignored:
  `BackgroundPict(`*sourceType;sourceName;location;adjustment;isProportional*`)`
  `ChartBackgroundPict(`*planeIndex;sourceType;sourceName;location;adjustment;isProp.*`)`
  `AddPicture(`*left;top;width;height;sourceType;sourceName;location;adjustment;isProp.*`)`

- The obsolete macOS PICT format has been removed in xmCHART 4 and higher.
  As a consequence, the output Function `SaveAsPICTFile()` is ignored.

- The 3rd argument *creatorType* is ignored in all output functions:
  SaveAsBMPFile(*fileName;fileFlag;creatorType;resolution*)
  SaveAsGIFFile(*fileName;fileFlag;creatorType;resolution*)
  SaveAsJPGFile(*fileName;fileFlag;creatorType;compression;resolution*)
  SaveAsPDFFile(*fileName;fileFlag;creatorType;title;subject;author*)
  SaveAsPNGFile(*fileName;fileFlag;creatorType;resolution*)
  SaveAsSVGFile(*fileName;fileFlag;creatorType;title;description;comment*)
  SaveAsTIFFile(*fileName;fileFlag;creatorType;resolution*)

- SaveAsPDFFile(*fileName;fileFlag;creatorType;title;subject;author;creator;keywords*)
  The last 2 arguments *creator* and *keywords* are ignored.

- The fileFlag constant *throwError* has been deprecated in all output functions.

- The argument *shadowPattern* has been removed in function ShadowStyle() and in all background functions:
  AxisLabelBackground()
  AxisMajorTickLabelBackground()
  AxisMinorTickLabelBackground()
  Background()
  ChartBackground()
  LabelBackground()
  LegendBackground()
  PieChartCenterLabelBackground()
  PieChartInnerLabelBackground()
  TitleBackground()

- The schemeIndex constant *none* has been deprecated in all color scheme functions:
  BorderColorScheme(*schemeIndex;colorVariant*)
  FillColorScheme(*schemeIndex;colorVariant*)
  LineColorScheme(*schemeIndex;colorVariant*)
  SymbolColorScheme(*schemeIndex;colorVariant*)

  Instead, use the new constant *classic* or simply set the argument schemeIndex to 0 for future projects. For example:

  ```
  FillColorScheme(0)       /* Okay. */
  FillColorScheme(classic) /* Okay. */
  FillColorScheme(none)    /* Not recommended, constant "none" is deprecated. */
  ```

- The clipboard (sourceType constant *clipboard*) and the export function SendToClipboard(*resolution*) have been deprecated.

- The built-in gradients (sourceType constant *resource*) have been deprecated. Instead, you should use the much more flexible color gradients.