



Last updated 2005-11-10

What's new in xmCHART 3.0

The fully redesigned FileMaker Pro 7/8 plug-in xmCHART 3.0 offers many new and updated features:

GRAPHICS ENGINE

- The new graphics engine supports GDI+ for Windows and Quartz for Mac OS X, thus making it possible to create high quality professional graphics with xmCHART 3.0.
- xmCHART 3.0 stores the drawing directly in a FileMaker container field. The workaround via the clipboard is no longer necessary. If desired, you can still copy a graph onto the clipboard via the `SendToClipboard()` function. It is now possible to create dynamic charts without launching a FileMaker script!
- Antialiasing: The 4th argument in `OpenDrawing()` controls antialiasing.
For example:
`OpenDrawing(400;300;;0)` // no antialiasing
`OpenDrawing(400;300;;1)` // antialiase geometric shapes
`OpenDrawing(400;300;;2)` // antialiase texts
`OpenDrawing(400;300;;3)` // antialiase geometric shapes and texts (default)
Note: The antialiasing flag works only for bitmaps in Mac OS X, i.e.
the 3rd argument must be set to 1, e.g.: `OpenDrawing(400;300;1;2)`
- Transparency (alpha channel) as optional 4th argument in R G B A
Range: 0 (invisible) ... 255 (opaque) default: 255
For example:
`FillStyle(1;100 0 88 255)` // same as `FillStyle(1;100 0 88)`
`FillStyle(1;255 0 255 200)`
`FillStyle(1;#FF8900DE)`
`FillStyle(1;darkRed 200)` // Error, not allowed!
`FillStyle(1;darkRed,200)` // Error, not allowed!
- Rotated texts and labels in any direction, for example:
`AddText(10;10;"Text";"Verdana";10;plain;blue;left;-90)`
Range: -360...+360 (+ clockwise, - counter clockwise)
- Dashed line styles. The dash pattern is attached to the line width.
The last "segment" is interpreted as dash offset in the event of an odd number of dash segments.
For example:
`LineStyle(1;poly;1 2 3 5 2;red)` // 2px dash 3px gap 5px dash 2px gap
`BorderStyle(1;4;1 8 4 2;green)` // 2px offset 8px dash 4px gap

- Variable smoothing factor for lines and borders
Range: 0.0 (polygonal)...2.0 (very smooth) (default: 1.0)
For example:
`LineStyle(all;4;1;darkBlue) // smoothing factor = 1.0 (default)`
`LineStyle(all;4 0;1;darkBlue) // smoothing factor = 0.0 (polygonal)`
`LineStyle(all;4 0.5;1;darkBlue) // smoothing factor = 0.5`
`LineStyle(all;smooth 0.5) // Error, not allowed!`
`LineStyle(all;smooth,0.5) // Error, not allowed!`
- Colors can now be entered as hexcodes, with or without alpha channel, #RRGGBB, #rrggbb, #RRGGBBAA, #rrggbbaa. For example:
`FillStyle(1;#FF8900DE)`
`BorderStyle(all;poly;2;#ff12c0)`
- Over 100 new color constants.
See list of constants at end of document.

INPUT/OUTPUT

- Entering chart values in scientific notation is now allowed.
For example: `ChartData(-1.2e04 0.2E04 .2e-3)`
- Support of "NULL" for missing charting values.
For example: `ChartData(12 98.3 null 23.1 Null NULL 7.23 -0.67)`
- Unicode support
Individual unicode characters can be entered in hexadecimal code.
For example:
`AxisLabelText(x;"Direction \u03b1") // "\u03b1" = Greek letter "alpha".`
- Save a drawing in a PDF file (only for Mac OS X)
For example: `SaveAsPDFFile("Macintosh HD:diagram.pdf";replace)`
- Support of C-style multi-line `/* ... */` comments.
- Function names and constants are no more case sensitive.
For example:
`BORDERSTYLE(all;POLY;2;darkblue) // allowed`
`BorderStyle(All;Poly;3;DARKBLUE) // no error`
- Improved error messages:
The corresponding script line is returned and the erroneous part is enclosed by `---> <---`. For example:
Function "AddRect": Argument #5: Value out of range.
`AddRect(10;10;100;100;255 255 --->256<---`)

DATE & TIME

- Entry of dates and times as charting values.

For example:

```
DateTimeOptions(mdy) // before ChartData()
ChartData(1/1/2005 1/4/2005 2/2/2005)
ChartData("2005-05-05" 2005-05-12 "2005-05-23")
ChartData(8:30 "9:15" 14:30:30 17:34 "22:34:12")
```

Time stamps are either to be placed in double quotes or the date and time components have to be concatenated using a "&".

For example:

```
ChartData("2005-05-05 8:30" "2005-05-15 9:15" "2005-05-16 0:20:30")
ChartData(5/5/2005&8:30 5/15/2005&9:15 5/16/2005&0:20:30)
```

- New and enhanced functions:

```
Scaling(axisIndex;type;minValue;maxValue;numOfMajorIntervals;numOfMinorIntervals;...)
```

minValue and maxValue can be of type number or date/time. For example:

```
Scaling(x;linear;1/1/2000;12/31/2010)
Scaling(x;linear;0:00;24:00;hour;minute15)
Scaling(x;linear;0;1;hour;minute15) // same as 0:00..24:00
```

18 constants for defining the major and minor scaling.

```
year      ( -1)
quarter   ( -2)
month     ( -3)
week      ( -4)
day       ( -5)
hour      ( -6)
minute30  ( -7)
minute20  ( -8)
minute15  ( -9)
minute10  (-10)
minute5   (-11)
minute    (-12)
second30  (-13)
second20  (-14)
second15  (-15)
second10  (-16)
second5   (-17)
second    (-18)
```

Examples:

```
Scaling(x;linear;;year;quarter)
Scaling(x;linear;2005-01-01;2005-12-31;week;day)
Scaling(x;linear;"2005-01-01 6:00";"2005-01-02 6:00";hour)
```

```
DateTimeOptions(dateOrder;startingDay) // before ChartData()!
```

```
dateOrder constants: ymd (1)
                    mdy (2)
                    dmy (3)
```

```
startingDay: 1...Sunday (default)
              2...Monday, ... 7...Saturday
```

startingDay is used for calculating the week of the year.

Examples:

```
DateTimeOptions(ymd)
DateTimeOptions(mdy;1) // US date format
DateTimeOptions(;2)    // week of year according to ISO 8601
```

- Comprehensive set of date and time output formatting options:
 - Year specifiers
 - YY 05 // year without century
 - YYYY 2005 // year with 4 digits
 - Quarter specifier
 - Q 1 // quarter of year (1..4)
 - Month specifiers
 - M 1 // month of year (1..12)
 - MM 01 // month of year with leading 0 (01..12)
 - Mo J // 1st letter of localized month name, uppercase
 - Mon Jan // abbr. month name, localized
 - MON JAN // abbr. month name, localized, uppercase
 - Month January // month name, localized
 - MONTH JANUARY // month name, localized, uppercase
 - Week specifiers
 - W 1 // weekday index (Sunday=1, Monday=2,..., Saturday=7)
 - WY 8 // week of year (1..53) starting day is defined by DateTimeOptions()
 - WWY 08 // week of year with leading 0 (01..53)
 - WD1 M // 1st letter of localized weekday name, uppercase
 - Wd2 Mo // 2-letter abbr. of localized weekday name
 - WD2 MO // 2-letter abbr. of localized weekday name, uppercase
 - Wd3 Mon // abbr. weekday name, localized
 - WD3 MON // abbr. weekday name, localized, uppercase
 - Weekday Monday // weekday name, localized
 - WEEKDAY MONDAY // weekday name, localized, uppercase
 - Day specifiers
 - D 2 // day of month (1..31)
 - DD 02 // day of month with leading 0 (01..31)
 - DY 83 // day of year (1..366)
 - DDY 083 // day of year with leading 0 (001..366)
 - Dx 2nd // English ordinal suffix, not localized
 - DX 2ND // English ordinal suffix, not localized, uppercase
 - Time specifiers
 - h 1 // hour in 24-hour format (0..23)
 - hh 01 // hour in 24-hour format with leading 0 (00..23)
 - h12 1 // hour in 12-hour format (1..12)
 - hh12 01 // hour in 12-hour format with leading 0 (01..12)
 - m 1 // minute (0..59)
 - mm 01 // minute with leading 0 (00..59)
 - s 1 // second (0..59)
 - ss 01 // second with leading 0 (00..59)
 - z1 .1 // 10th of a second
 - z2 .12 // 100th of a second
 - z3 .123 // 1000th of a second (millisecond)
 - ampm am // am or pm (in combination with h12 or hh12)
 - AMPM PM // AM or PM (in combination with h12 or hh12)

Examples:

```
AxisMajorTickLabelTexts(x;"|M/D/YY|")
AxisMajorTickLabelTexts(x;"|YYYYMMDD\nhhmm|")
AxisMajorTickLabelTexts(y;"|h12:mm AMPM|")
AxisMajorTickLabelTexts(y;"Week #|WY|")
```

NEW AND IMPROVED CHART TYPES

- BarChart(): 3D-bars implemented
BarChart(appearance;
 categoryGap; // [%] of bar width
 seriesGap; // [%] of bar width
 barDepth) // [%] of bar width
- GanttChart(): 3D-bars implemented
GanttChart(appearance;
 categoryGap; // [%] of bar width
 barDepth) // [%] of bar width
- PieChart(): 3D-pies implemented
PieChart(appearance;
 pieDepth; // [%] of pie radius
 innerRadius; // [%] of pie radius
 startAngle; // [deg]
 arcAngle) // [deg]
- PieChartCenterLabelStyle(): 4 new arguments added
PieChartCenterLabelStyle(font;
 size;
 style;
 color;
 alignment;
 orientation;
 maxWidth;
 maxHeight;
 ellipsisPosition)
- PieChartInnerLabelStyle(): 4 new arguments added
PieChartInnerLabelStyle(font;
 size;
 style;
 color;
 alignment;
 orientation;
 maxWidth;
 maxHeight;
 ellipsisPosition)

The arguments *orientation*, *maxWidth*, *maxHeight* and *ellipsisPosition* are explained in combination with the LabelStyle() function.

- AreaChart2D(): new chart type implemented
AreaChart2D(appearance) works like LineChart2D() with fills
- RadarChart(): 1 new argument added
RadarChart(appearance;
 startAngle; // [deg]
 arcAngle; // [deg]
 doShiftValues)

AXES

- AxisLabelStyle(): 4 new arguments added
AxisLabelStyle(axisIndex;
font;
size;
style;
color;
alignment;
orientation;
maxWidth;
maxHeight;
ellipsisPosition)
- AxisMajorTickLabelStyle(): 4 new arguments added
AxisMajorTickLabelStyle(axisIndex;
font;
size;
style;
color;
alignment;
orientation;
maxWidth;
maxHeight;
ellipsisPosition)
- AxisMinorTickLabelStyle(): 4 new arguments added
AxisMinorTickLabelStyle(axisIndex;
font;
size;
style;
color;
alignment;
orientation;
maxWidth;
maxHeight;
ellipsisPosition)

The arguments *orientation*, *maxWidth*, *maxHeight* and *ellipsisPosition* are explained in combination with the LabelStyle() function.

LEGEND

- more flexible layout options, legend texts layout: row by row, column by column, reverse direction
- legend markers can be arranged left/right/top/bottom of the legend texts
- LegendOptions(): enhanced 5th argument and 1 new argument added

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

<i>distribution flags</i>	<i>range</i>	<i>default</i>
numOfRows	-1..1000	10
numOfColumns	-1..1000	-1
doArrangeColumnByColumn	0..1	0
doArrangeReversed	0..1	0
useEquidistantColumnWidths	0..1	0
useEquidistantRowHeights	0..1	0

-1...variable

Examples:

```
LegendOptions(;;;;2)           // 2 rows (variable num of columns)  
LegendOptions(;;;;2 -1)        // 2 rows (same as above)  
LegendOptions(;;;-1 2)         // 2 columns (variable num of rows)  
LegendOptions(;;;;3 -1 1)      // 3 rows with layout column by column  
LegendOptions(;;;-1 1 0 1)     // 1 column, direction bottom to top
```

textLocation

1..9: topLeft..bottomRight (default: centerRight)

MOVING AVERAGES

All moving average functions have as 3rd argument: *calculationMethod*. Now you can display and compare moving average curves with different calculation methods simultaneously.

- MovingAverage(): 1 new argument (not compatible with xmCHART 2)
MovingAverage(seriesIndex;
 numOfIntervals;
 calculationMethod;
 weightList)
- MovingAverageLineStyle(): 1 new argument (not compatible with xmCHART 2)
MovingAverageLineStyle(seriesIndex;
 numOfIntervals;
 calculationMethod;
 lineShape;
 lineStroke;
 lineColor;
 linePattern)
- xmCHART 3.0 supports moving averages in combination with bar charts.

GRIDS

- Major and minor grid stripes (oval and polygonal) for radar and polar charts implemented.

GRAPHICS PRIMITIVES

- AddText(): 5 new arguments added

```
AddText(left;  
        top;  
        text;  
        font;  
        size;  
        style;  
        color;  
        horizontalAlignment;  
        verticalAlignment;  
        orientation;  
        maxWidth;  
        maxHeight;  
        ellipsisPosition)
```

The arguments *orientation*, *maxWidth*, *maxHeight* and *ellipsisPosition* are explained in combination with the LabelStyle() function.

- AddPicture(): 3 new arguments added

```
AddPicture(left;  
           top;  
           width;  
           height;  
           pictSourceType;  
           pictSourceName;  
           location;  
           adjustment;  
           isProportional)
```

- AddSlice(): 1 new argument (not compatible with xmCHART 2)

```
AddSlice(left;  
         top;  
         width;  
         height;  
         startAngle; // [deg]  
         arcAngle;   // [deg]  
         innerRadius; // [%] of slice radius  
         fillColor;  
         fillPattern)
```

- AddClipSlice(): 1 new argument added

```
AddClipSlice(type;  
             left;  
             top;  
             width;  
             height;  
             startAngle; // [deg]  
             arcAngle;   // [deg]  
             innerRadius) // [%] of slice radius
```

STYLES

- LabelStyle(): 4 new arguments added

```
LabelStyle(seriesIndex;  
            font;  
            size;  
            style;  
            color;  
            alignment;  
            orientation;  
            maxWidth;  
            maxHeight;  
            ellipsisPosition)
```

orientation:

-360..360 [degree] (+ clockwise, - counter clockwise)

maxWidth:

-1: no predefined max. width (default)

-2: use automatic width (only available for bar chart and Gantt chart labels)

maxHeight:

-1: no predefined max. height (default)

-2: use automatic height (only available for bar chart and Gantt chart labels)

ellipsisPosition

0: text trimmed at the end without ellipsis

1: text trimmed at the begin (...text)

2: text trimmed in the middle (text...text)

3: text trimmed at the end (text...)

4: Text trimmed at the begin and end: (...text...)

Examples:

```
LabelStyle(all;"Verdana";10;;;-45)
```

```
LabelStyle(all;"Verdana";10:bold:black;2;-90;-2;-2) // Bar & Gantt charts
```

NEW EXTERNAL FUNCTIONS

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

Examples:

```
xmCH_GetVersion(1) // returns, for example: "xmCHART 3.0.4"
xmCH_GetVersion(2) // returns, for example: "3.0.4"
```

```
xmCH_GetErrorMessage(flags) // flags is of type string!
```

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

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

Examples:

```
xmCH_GetErrorMessage("103") // default
xmCH_GetErrorMessage("110") // play sound and show only error code
xmCH_GetErrorMessage("110$$") // play sound and show error code, e.g. $$2058
```

COMPATIBILITY with xmCHART 2.2

- Since xmCHART 3.0 supports the new FileMaker 7 Plug-in API, the "External" in front of all your external function calls must be removed, e.g.:
in v2.2: External(xmCH-DrawChart(source))
in v3.0: xmCH_DrawChart(source) // underscore instead of hyphen!
- xmCHART 3.0 stores the created chart directly in a container field. The workaround via the clipboard is no longer required. Possible error messages are sent automatically to the container field and can also be requested via the external function xmCH_GetErrorMessage().
in v2.2: errorMessage = External(xmCH-DrawChart(source))
in v3.0: containerField = xmCH_DrawChart(source)
errorMessage = xmCH_GetErrorMessage("103")

If the source field is a calculation field, the chart is updated automatically without launching a FileMaker script!
- All functions from v2.2 are compatible with v3.0, except the following 3 calls: MovingAverage(), MovingAverageLineStyle(), AddSlice().

KNOWN ISSUES:

- Windows: The scaling bug regarding the EMF vector format already known from previous FileMaker Pro versions still exists in FileMaker Pro 7. Therefore, the workaround from v2.2 is still necessary:

Workaround:

The following two points are to be noted in order to display a metafile drawing, free from distortion and in the proper size in a FileMaker Pro container field:

- (1) The size of the container field and the drawing must be identical or more precisely, the container field should be 2 pixel larger than the drawing, as already explained above. The width and height of the drawing are defined (in pixels) by the function OpenDrawing(width;height;...). The width and height of the container field can be controlled in the Layout Mode under menu item Object Size... (By clicking the mouse onto one of the dimension labels [cm, px, in] on the right margin of the Object Size window it is possible to switch back and forth between centimeters, pixels and inches.)
 - (2) In the Layout Mode, set the Graphic Format of the chart container field under menu item Graphic... to "Reduce or Enlarge" and switch off the check box "Maintain original proportions". Switching off the "Proportional" check box is important, otherwise pictures will appear mirrored.
- Mac OS X: Due to a nasty bug in FileMaker Pro 7 antialiasing for drawings in vector format is switched on/off inadvertently by clicking inside or outside of a FileMaker field. Very strange and ugly, especially when displaying round shapes, like in pie charts!

Workaround:

For the screen display create an antialiased bitmap, e.g.:

OpenDrawing(400;300;1;3)

For printing draw the chart a 2nd time in vector format, e.g.:

OpenDrawing(400;300;0)

This bug has been fixed in FileMaker Pro 8.0

NEW COLOR CONSTANTS

aliceblue	240	248	255
antiquewhite	250	235	215
aqua	0	255	255
aquamarine	127	255	212
azure	240	255	255
beige	245	245	220
bisque	255	228	196
black	0	0	0
blanchedalmond	255	235	205
blue	0	0	221
blueviolet	138	43	226
brown	165	42	42
burlywood	222	184	135
cadetblue	95	158	160
chartreuse	127	255	0
chocolate	210	105	30
coral	255	127	80
cornflowerblue	100	149	237
cornsilk	255	248	220
crimson	220	20	60
cyan	0	255	255
darkblue	0	0	119
darkcyan	0	139	139
darkgoldenrod	184	134	11
darkgray	102	102	102
darkgreen	0	102	0
darkkhaki	189	183	107
darkmagenta	139	0	139
darkolivegreen	85	107	47
darkorange	255	140	0
darkorchid	153	50	204
darkpurple	204	0	153
darkred	204	51	0
darksalmon	233	150	122
darkseagreen	143	188	143
darkslateblue	72	61	139
darkslategray	47	79	79
darkturquoise	0	206	209
darkviolet	148	0	211
darkyellow	255	204	0
deeppink	255	20	147
deepskyblue	0	191	255
dimgray	105	105	105
dodgerblue	30	144	255
firebrick	178	34	34
floralwhite	255	250	240
forestgreen	34	139	34
fuchsia	255	0	255
gainsboro	220	220	220
ghostwhite	248	248	255
gold	255	215	0
goldenrod	218	165	32
gray	136	136	136
green	0	136	0
greenyellow	173	255	47
honeydew	240	255	240
hotpink	255	105	180
indianred	205	92	92
indigo	75	0	130

ivory	255	255	240
khaki	240	230	140
lavender	230	230	250
lavenderblush	255	240	245
lawngreen	124	252	0
lemonchiffon	255	250	205
lightblue	153	204	255
lightcoral	240	128	128
lightcyan	224	255	255
lightgoldenrodyellow	250	250	210
lightgray	221	221	221
lightgreen	153	255	153
lightpink	255	182	193
lightpurple	255	153	255
lightred	255	153	153
lightsalmon	255	160	122
lightseagreen	32	178	170
lightskyblue	135	206	250
lightslategray	119	136	153
lightsteelblue	176	196	222
lightyellow	255	255	153
lime	0	255	0
limegreen	50	205	50
linen	250	240	230
magenta	255	0	255
maroon	128	0	0
mediumaquamarine	102	205	170
mediumblue	0	0	205
mediumorchid	186	85	211
mediumpurple	147	112	219
mediumseagreen	60	179	113
mediumslateblue	123	104	238
mediumspringgreen	0	250	154
mediumturquoise	72	209	204
mediumvioletred	199	21	133
midnightblue	25	25	112
mintcream	245	255	250
mistyrose	255	228	225
moccasin	255	228	181
navajowhite	255	222	173
navy	0	0	128
oldlace	253	245	230
olive	128	128	0
olivedrab	107	142	35
orange	255	165	0
orangered	255	69	0
orchid	218	112	214
palegoldenrod	238	232	170
palegreen	152	251	152
paletturquoise	175	238	238
palevioletred	219	112	147
papayawhip	255	239	213
peachpuff	255	218	185
peru	205	133	63
pink	255	192	203
plum	221	160	221
powderblue	176	224	230
purple	255	0	153
red	255	51	0
rosybrown	188	143	143
royalblue	65	105	225
saddlebrown	139	69	19
salmon	250	128	114
sandybrown	244	164	96

seagreen	46	139	87
seashell	255	245	238
sienna	160	82	45
silver	192	192	192
skyblue	135	206	235
slateblue	106	90	205
slategray	112	128	144
snow	255	250	250
springgreen	0	255	127
steelblue	70	130	180
tan	210	180	140
teal	0	128	128
thistle	216	191	216
tomato	255	99	71
turquoise	64	224	208
violet	238	130	238
wheat	245	222	179
white	255	255	255
whitesmoke	245	245	245
yellow	255	255	0
yellowgreen	154	205	50