

# Examples



*X2max*  
[www.x2max.com](http://www.x2max.com)

FileMaker Pro is a registered trademark of FileMaker, Inc.  
© 1997-2006 by X2max Software. All rights reserved.

# Table Of Contents

## Examples

<b>Introduction</b>	3
<b>Example 1</b>	5
<b>Example 2</b>	9
<b>Example 3</b>	16
<b>Example 4</b>	23
<b>Example 5</b>	32
<b>Example 6</b>	42

# Introduction

In the following the practical ways of using xmCHART are shown by way of selected examples. More extensive examples were purposely chosen to, on the one hand, show the versatility and power of xmCHART and, on the other hand, to illustrate the important interaction between FileMaker Pro scripts and xmCHART. By using FileMaker Pro scripts, it is possible to fully automate the creation of charts so that users can create expressive charts "at the touch of a button". All examples shown here are available as fully functional unlocked FileMaker Pro .FP7 files.

## Integrating xmCHART into FileMaker Pro databases:

### • FileMaker Pro fields:

In general, xmCHART requires three FileMaker Pro fields; advantageously, they are global fields which are denoted by the prefix "g" (global) in the following:

*gFunctions*: contains all function calls required for xmCHART.  
*gChart*: contains the chart created by xmCHART.  
*gError*: contains information on possible errors that may occur while creating the chart.

Additional so-called placeholders can be used to make the practical handling easier. To do this, another text global, for example with the name *gFunctionsWithPlaceholders*, is introduced. Just like the text variable *gFunctions*, it contains all function calls which are necessary for xmCHART, but instead of the "variable" data — i.e. the data which is normally taken from the database fields — placeholders are entered. Placeholders prove useful, e.g. for chart data, axis labels or legend texts. Placeholders are generally marked by pointed brackets "<>". For example:

```
ChartData(<gData>)  
AxisMajorTickLabelTexts(x;<gAxisLabels>)
```

A separate text global is set up for each placeholder, e.g. for <gData> a text global *gData*, for <gAxisLabels> a text global *gAxisLabels*, etc. The contents of these text globals are generally set up by

a FileMaker Pro script. Finally all placeholders are replaced by the actual values using another FileMaker Pro script. The working with placeholders is explained in the first three examples.

- *FileMaker Pro scripts:*

To draw a chart automatically at least one FileMaker Pro script is necessary which, for the most part, is divided into several subscripts to increase the clarity. The names of the subscripts always begin with an underscore, e.g. *\_SubstitutePlaceholders*.

*Instead of a text field and container field, a calculation field can also be used. The advantage of a calculation field is that no FileMaker Pro Script has to be called to create the chart, or, otherwise stated, the chart is automatically updated as soon as the values in the calculation field change which, in many cases, can prove to be extremely advantageous.*

To familiarize yourself with xmCHART quickly and easily, the field and script names used above have been kept the same in all examples.

In the example files an effort was made to represent the essential steps, starting with the collection of values to be depicted down to the finished chart, in a clear and comprehensible manner. However, in a professional database solution, neither the scripts necessary to create the function calls nor the function calls themselves are generally visible for the user so that a chart can mostly be created "at the touch of a button". It is up to the developer of a database solution as to which options for creating a chart would be made available to the user.

All data necessary for creating a chart has to be copied to xmCHART via the `ChartData()` function. For this reason, the script needed to set up the `ChartData()` function usually holds a central position. Possible methods are shown in the following examples. The other function calls necessary for xmCHART can mostly be added per script by concatenating text strings.

PLEASE NOTE: *When entering text strings or font names in the Script Editor, it should be noted that the required double quotes must always be entered twice. For example:*

```
gFunctions & "TitleText(" "Chart 1" ")"
```

# Example 1

## Problem:

In the first example a series of values is represented in the form of a pie chart (Fig. 8). The individual slices of the pie should be labeled with the name of the category, the actual value and the percentage. The corresponding FileMaker Pro database is xmEXAMPLE1.FP7.

## Solution:

### FileMaker Pro fields: (Fig. 1)

In addition to the global fields *gFunctions*, *gChart*, *gError* and *gFunctions-WithPlaceholders* described previously in the introduction, the following fields are also necessary:

- The text field *Name* which contains the name of the category.
- The number field *Value* which contains the value of the category.
- The text globals *gData*, *gLabels* and *gFormat*. They correspond to the matching placeholders *<gData>*, *<gLabels>* and *<gFormat>* and serve to store the actual values.
- Optional: the global fields *gLanguage* and *gVersion*. *gVersion* contains the exact name of the presently active xmCHART version, e.g. "xmCHART 3.1".

<u>Field Name</u>	<u>Type</u>	<u>Options</u>
Name	Text	
Value	Number	
gFunctionsWithPlaceholders	Text	Global
gFunctions	Text	Global
gChart	Container	Global
gError	Text	Global
gVersion	Text	Global
gLanguage	Text	Global
gData	Text	Global
gLabels	Text	Global
gFormat	Text	Global

Fig. 1

**FileMaker Pro scripts:**

- FileMaker Pro script *SetUpChart*: (Fig. 2)  
The script *SetUpChart* produces the desired chart and consists of several subscripts.

```

Freeze Window
Perform Script [ "_SetUpData" ]
Perform Script [ "_SetUpLabels" ]
Perform Script [ "_SubstitutePlaceholders" ]
Perform Script [ "_DrawChart" ]

```

Fig. 2

*(1) \_SetUpData*: (Fig. 3)

The subscript *\_SetUpData* takes the necessary data from the database, more precisely from the number field *Value*, and stores it in the form of a character string — the individual values are separated by spaces — in *gData*. This is done by using a loop over all records.

```

Set Field [ xmCHART::gData; "" ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [xmCHART::gData; xmCHART::gData &
        GetAsNumber(xmCHART::Value) & " " ]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop

```

Fig. 3

*(2) \_SetUpLabels*: (Fig. 4)

The subscript *\_SetUpLabels* takes the category names from the database and stores them in *gLabels*. The individual names are to be placed in double quotes and separated by semicolons. In addition, *gFormat* is added to every name. The text global *gFormat* defines the presentation of the numerical values: the first format specifier defines the output of the absolute values, the second format specifier the output of the percent values. Format specifiers are discussed in detail in *xmReference* and the various options for labeling pie charts in the *Styles* section of *xmTutorial*. Please note that when entering text or font names in the Script Editor the required double quotes must always be entered twice. For example:

```
gFunctions & "TitleText(""Chart 1"")"
```

```

Set Field [ xmCHART::gLabels; "" ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [xmCHART::gLabels; xmCHART::gLabels & "\" &
        xmCHART::Name & xmCHART::gFormat & "\";" ]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [xmCHART::gLabels;
    Left(xmCHART::gLabels; Length(xmCHART::gLabels)-1) ]

```

Fig. 4

(3) *\_SubstitutePlaceholders*: (Fig. 5)

The subscript *\_SubstitutePlaceholders* replaces the placeholders in *gFunctionsWithPlaceholders* (Fig. 6) with the actual values.

```

Set Field [xmCHART::gFunctions;
    Substitute(xmCHART::gFunctionsWithPlaceholders;
        "<gData>"; xmCHART::gData) ]
Set Field [xmCHART::gFunctions; Substitute(xmCHART::gFunctions;
    "<gLabels>"; xmCHART::gLabels) ]

```

Fig. 5

```

OpenDrawing(400;300)
    ChartData(<gData>)
    PieChart(label;10;60)
    BorderStyle(all;0) // hide borders
    PieChartAuxLines() // labels with auxiliary lines
    LabelTexts(1;<gLabels>)
    LabelStyle(all;"Verdana";9;;;center)// labels centered
    LabelBackground(all;lightYellow;;0;;;2)
CloseDrawing()

```

Fig. 6

(4) *\_DrawChart*: (Fig. 7)

In the subscript *\_DrawChart* all functions included in *gFunctions* are sent to xmCHART by the external function *xmCH\_DrawChart()*. If no error occurs the chart is then stored in the container field *gChart*.

```

Set Field [ xmCHART::gChart; xmCH_DrawChart( xmCHART::gFunctions ) ]
Set Field [ xmCHART::gError; xmCH_GetErrorMessage( "103" ) ]

```

Fig. 7

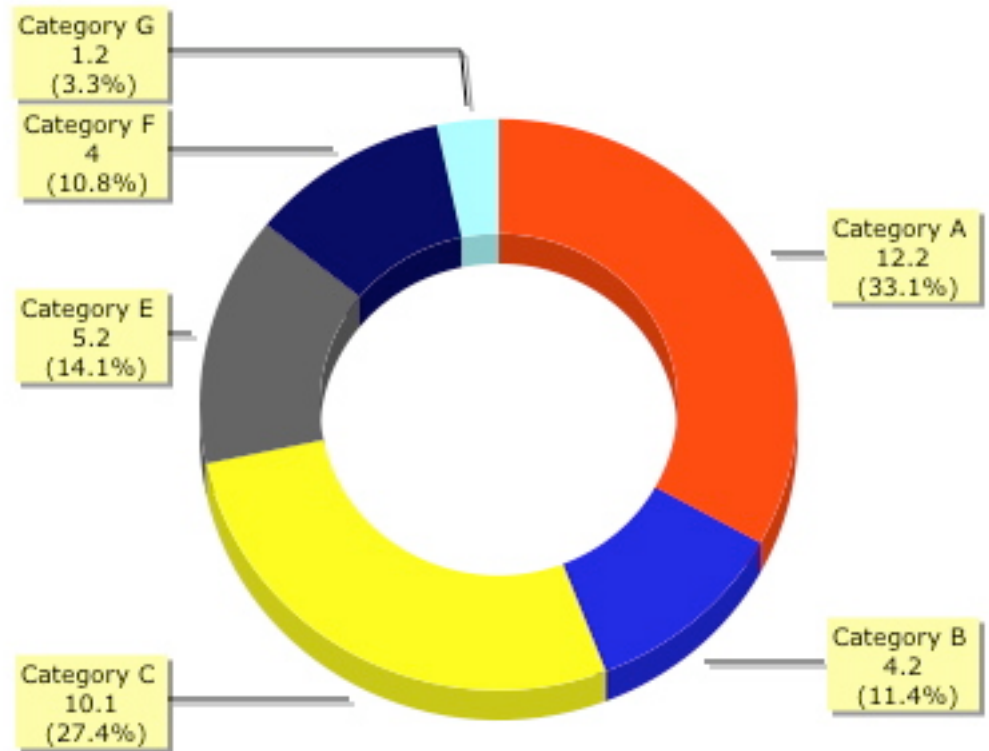
**Result:**

Fig. 8

**Variations:**

The labels can be positioned inside the chart or split, i.e. both inside and outside the chart. In addition, a text can also be placed in the center of the pie chart. Details with examples can be found in *xmTutorial*, section *Charts*.

## Example 2

### Problem:

In the second example time-controlled data in the form of a so-called Gantt chart are to be represented (Fig. 17). To put it more precisely, the different periods of time for a series of tasks are to be plotted along a horizontal time axis. Depending on the period to be represented - whether it's days, weeks, months or years - an appropriate scaling should automatically be found. The different types of scaling can be tested by changing the periods of time (*from-to*) in the corresponding FileMaker Pro file xmEXAMPLE2.FP7.

### Solution:

#### FileMaker Pro fields: (Fig. 9)

In addition to the global fields *gFunctions*, *gChart*, *gError* and *gFunctions-WithPlaceholders* described previously in the introduction, the following fields are also necessary:

- The text field *Task* which contains the name of the task.
- The date fields *From1*, *To1*, *From2* and *To2*. A max. of two time periods can be defined for each task. In this concrete example the subdivision of each task was restricted to two time periods; however, in xmCHART each task can be divided into as many periods as desired.
- The global fields *gData*, *gXScaling*, *gXAxisMajorTickLabelTexts*, *gXAxis-MinorTickLabelTexts*, *gXAxisMajorTickLabelOrientation*, *gXAxis-MinorTickLabelOrientation*, *gXAxisMajorTickLabelVOffset* and *gYAxisLabelTexts* correspond to the matching placeholders *<gData>*, *<gXScaling>* etc. and contain the actual values.
- The remaining date and number globals are necessary for calculating the different scaling periods and labels.
- Optional: the global fields *gLanguage* and *gVersion*. *gVersion* contains the exact name of the presently active xmCHART version, e.g. "xmCHART 3.1".

<u>Field Name</u>	<u>Type</u>	<u>Options</u>
Task	Text	
From1	Date	
To1	Date	
From2	Date	
To2	Date	
gFunctionsWithPlaceholders	Text	Global
gFunctions	Text	Global
gChart	Container	Global
gError	Text	Global
gData	Text	Global
gXScaling	Text	Global
gXAxisMajorTickLabelTexts	Text	Global
gXAxisMinorTickLabelTexts	Text	Global
gXAxisMajorTickLabelOrientation	Text	Global
gXAxisMinorTickLabelOrientation	Text	Global
gXAxisMajorTickLabelVOffset	Text	Global
gYAxisLabelTexts	Text	Global
gMinDate	Date	Global
gMaxDate	Date	Global
gStartDate	Date	Global
gEndDate	Date	Global
gNumOfYears	Number	Global
gNumOfMonths	Number	Global
gNumOfWeeks	Number	Global
gNumOfDays	Number	Global
gLanguage	Text	Global
gVersion	Text	Global

Fig. 9

**FileMaker Pro scripts:**

- FileMaker Pro script *SetUpChart*: (Fig. 10)

The script *SetUpChart* produces the desired chart and consists of several subscripts.

**Freeze Window****Perform Script [ “\_SetUpXAxis” ]****Perform Script [ “\_SetUpYAxis” ]****Perform Script [ “\_SetUpData” ]****Perform Script [ “\_SubstitutePlaceholders” ]****Perform Script [ “\_DrawChart” ]**

Fig. 10

(1) *\_SetUpXAxis*: (Fig. 11)

The subscript *\_SetUpXAxis* calculates the appropriate scaling for the defined periods of time. First the subscript *\_\_SetUpMinMaxDate* is called in which all records are searched for the earliest and latest date. They are stored in *gMinDate* and *gMaxDate*. In subscript *\_\_CalculateRange* the number of years, months, weeks and days are calculated based on *gMinDate* and *gMaxDate*. In the third subscript *\_\_SetUpScalingParameters* the number of years, months, weeks and days between *gMinDate* and *gMaxDate* are calculated.

```

Set Field [ xmCHART::gXAxisMajorTickLabelTexts; "" ]
Set Field [ xmCHART::gXAxisMinorTickLabelTexts; "" ]
Set Field [ xmCHART::gXAxisMajorTickLabelOrientation; "0" ]
Set Field [ xmCHART::gXAxisMinorTickLabelOrientation; "0" ]
Set Field [ xmCHART::gXAxisMajorTickLabelVOffset; "0" ]
If [ GetAsNumber(xmCHART::gNumOfYears) > 12 ]
    Set Field [ xmCHART::gXScaling; ";;year" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelTexts; ""|YYYY|\ "" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelOrientation; "-90" ]
Else If [ GetAsNumber(xmCHART::gNumOfMonths) > 36 ]
    Set Field [ xmCHART::gXScaling; ";;year;quarter" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelTexts; ""|YYYY|\ "" ]
Else If [ GetAsNumber(xmCHART::gNumOfMonths) > 12 ]
    Set Field [ xmCHART::gXScaling; ";;year;month" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelTexts; ""|YYYY|\ "" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelVOffset; "15" ]
    Set Field [ xmCHART::gXAxisMinorTickLabelTexts; ""|Mo|\ "" ]
Else If [ GetAsNumber(xmCHART::gNumOfWeeks) > 12 ]
    Set Field [ xmCHART::gXScaling; ";;month;week" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelTexts; ""|Mon\nYYYY|\ "" ]
Else If [ GetAsNumber(xmCHART::gNumOfWeeks) > 4 ]
    Set Field [ xmCHART::gXScaling; ";;week;day" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelTexts; ""WK |WY\nM/D|\ "" ]
Else If [ GetAsNumber(xmCHART::gNumOfWeeks) > 1 ]
    Set Field [ xmCHART::gXScaling; ";;week;day" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelTexts; ""Week |WY|\ "" ]
    Set Field [ xmCHART::gXAxisMinorTickLabelTexts; ""|M/D|\ "" ]
    Set Field [ xmCHART::gXAxisMinorTickLabelOrientation; "-90" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelVOffset; "30" ]
Else
    Set Field [ xmCHART::gXScaling; ";;day" ]
    Set Field [ xmCHART::gXAxisMajorTickLabelTexts;
        ""|Weekday\nM/D/YYYY|\ "" ]
End If

```

Fig. 11

(2) *\_SetUpYAxis*: (Fig. 12)

The subscript *\_SetUpYAxis* takes the label texts for the y-axis from the database and stores them in *gYAxisLabelTexts*. The individual names are to be placed in double quotes and separated by semicolons.

```

Set Field [ xmCHART::gYAxisLabelTexts; "" ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [ xmCHART::gYAxisLabelTexts;
        xmCHART::gYAxisLabelTexts &
        "\"" & xmCHART::Task & "\";"]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gYAxisLabelTexts;
    Left(xmCHART::gYAxisLabelTexts;
        Length(xmCHART::gYAxisLabelTexts)-1) ]

```

Fig. 12

(3) *\_SetUpData*: (Fig. 13)

By using a loop over all records (one record corresponds to one task), the subscript *\_SetUpData* takes the dates *From1*, *To1*, *From2* and *To2* from the database and stores them in the form of a string in *gData*. Only the numbers of days since *gStartDate* are stored. Since each record corresponds to a data series, a semicolon must be added to the string after each record. In addition, the completeness of the data has to be checked. In other words, the period of time for each task must be clearly defined, otherwise it will be ignored.

```

Set Field [ xmCHART::gData; "" ]
Go to Record/Request/Page [ First ]
Loop
    If [ not IsEmpty(xmCHART::From1) and not IsEmpty(xmCHART::To1) ]
        Set Field [ xmCHART::gData; xmCHART::gData &
            Year ( xmCHART::From1 ) & "-" &
            Month ( xmCHART::From1 ) & "-" &
            Day ( xmCHART::From1 ) & " " &
            Year ( xmCHART::To1 ) & "-" &
            Month ( xmCHART::To1 ) & "-" &
            Day ( xmCHART::To1 ) & " " ]
        If [not IsEmpty(xmCHART::From2) and not IsEmpty(xmCHART::To2)]
            Set Field [ xmCHART::gData; xmCHART::gData &
                Year ( xmCHART::From2 ) & "-" &
                Month ( xmCHART::From2 ) & "-" &

```

```

Day ( xmCHART::From2 ) & " " &
Year ( xmCHART::To2 ) & "-" &
Month ( xmCHART::To2 ) & "-" &
Day ( xmCHART::To2 ) & ";" ]
Else
    Set Field [ xmCHART::gData; xmCHART::gData & ";" ]
End If
Else
    If [ not IsEmpty(xmCHART::From2) and not IsEmpty(xmCHART::To2)]
        Set Field [ xmCHART::gData; xmCHART::gData &
            Year ( xmCHART::From2 ) & "-" &
            Month ( xmCHART::From2 ) & "-" &
            Day ( xmCHART::From2 ) & " " &
            Year ( xmCHART::To2 ) & "-" &
            Month ( xmCHART::To2 ) & "-" &
            Day ( xmCHART::To2 ) & ";" ]
        Else
            Set Field [ xmCHART::gData; xmCHART::gData & "null null ;" ]
        End If
    End If
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [xmCHART::gData;
    Left(xmCHART::gData; Length(xmCHART::gData) - 1) ]

```

Fig. 13

(4) *\_SubstitutePlaceholders*: (Fig. 14)

The subscript *\_SubstitutePlaceholders* replaces the placeholders in *gFunctionsWithPlaceholders* (Fig. 15) with the actual values.

```

Set Field [ xmCHART::gFunctions;
    Substitute(xmCHART::gFunctionsWithPlaceholders;
        "<gData>"; xmCHART::gData) ]
Set Field [ xmCHART::gFunctions; Substitute(xmCHART::gFunctions;
    "<gXScaling>"; xmCHART::gXScaling) ]
Set Field [ xmCHART::gFunctions; Substitute(xmCHART::gFunctions;
    "<gXAxisMajorTickLabelTexts>";
    xmCHART:: gXAxisMajorTickLabelTexts) ]
Set Field [ xmCHART::gFunctions; Substitute(xmCHART::gFunctions;
    "<gXAxisMinorTickLabelTexts>";
    xmCHART:: gXAxisMinorTickLabelTexts) ]
Set Field [ xmCHART::gFunctions; Substitute(xmCHART::gFunctions;
    "<gXAxisMajorTickLabelOrientation>";

```

```

xmCHART::gXAxisMajorTickLabelOrientation) ]
Set Field [ xmCHART::gFunctions; Substitute(xmCHART::gFunctions;
"<gXAxisMinorTickLabelOrientation>";
xmCHART::gXAxisMinorTickLabelOrientation) ]
Set Field [ xmCHART::gFunctions; Substitute(xmCHART::gFunctions;
"<gXAxisMajorTickLabelVOffset>";
xmCHART::gXAxisMajorTickLabelVOffset) ]
Set Field [ xmCHART::gFunctions; Substitute(xmCHART::gFunctions;
"<gYAxisLabelTexts>"; xmCHART::gYAxisLabelTexts) ]

```

Fig. 14

```

OpenDrawing(400;300;;2)
DateTimeOptions(ymd;1)
ChartData(<gData>)
GanttChart(shadow+label)
FillStyle(all;darkYellow)
ShadowStyle(all;1)
Scaling(x;linear;<gXScaling>)
ScalingOptions(y;on) // y-scaling top to bottom
// axes
AxisMajorTickLabelStyle(all;"Verdana";10)
AxisMajorTickLabelTexts(x;<gXAxisMajorTickLabelTexts>)
AxisMinorTickLabelTexts(x;<gXAxisMinorTickLabelTexts>)
AxisMajorTickLabelStyle(x;"Verdana";;;;
<gXAxisMajorTickLabelOrientation>)
AxisMinorTickLabelStyle(x;"Verdana";;;;
<gXAxisMinorTickLabelOrientation>)
AxisMajorTickLabelOptions(x;;;
<gXAxisMajorTickLabelVOffset>)
AxisMajorTickLabelTexts(y;<gYAxisLabelTexts>)
AxisLine(all;0)
AxisMajorTicks(all;0)
AxisMinorTicks(all;0)
// grid
MajorGridLineWidths(x;y;0) // hide horizontal grid
MinorGridLineColors(all;all;lightGray)
GridFrame(xy;1;gray)
CloseDrawing()

```

Fig. 15

(5) *\_DrawChart*: (Fig. 16)

In the subscript *\_DrawChart* all functions included in *gFunctions* are sent to xmCHART by the external function *xmCH\_DrawChart()*. If no error occurs, the chart is then stored in the container field *gChart*.

```
Set Field [ xmCHART::gChart; xmCH_DrawChart( xmCHART::gFunctions ) ]
Set Field [ xmCHART::gError; xmCH_GetErrorMessage( "103" ) ]
```

Fig. 16

### Result:

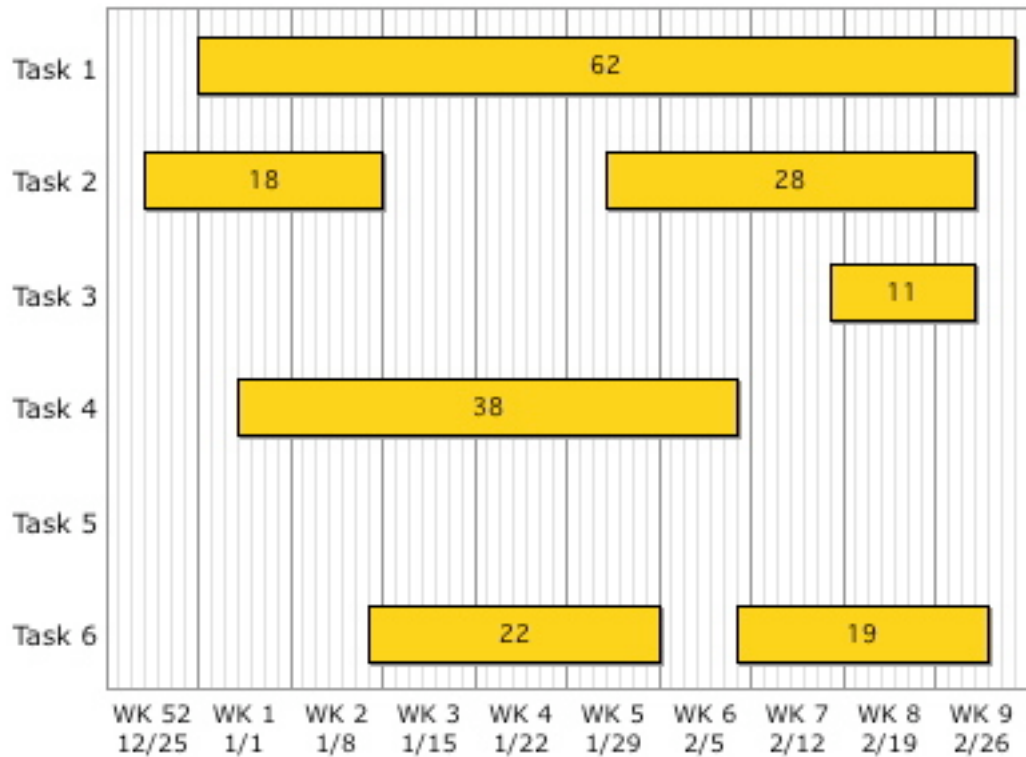


Fig. 17

### Variations:

The four scalings used in this example can be refined even further; the same goes for the scaling labels. In addition, the bars can be labeled or rotated 90° so that the time axis runs from top to bottom instead of from left to right. Examples of this can be found in the *Charts* section of *xmTutorial*.

## Example 3

**Problem:**

In the third example two charts, a bar chart and a line chart, are to be positioned over each other (Fig. 26). The corresponding FileMaker Pro file is xmEXAMPLE3.FP7. The following points are to be noted when overlaying charts:

- The charts, more precisely the areas enclosed by the grid, must cover each other exactly. This can be done by using the `OpenChart()` and `CloseChart()` functions. Please note that *isPlotArea=on* is set. The basic setup is:

```
OpenChart(left;top;width;height;isPlotArea)
//
// set up chart 1
//
CloseChart()

OpenChart(left;top;width;height;isPlotArea)
//
// set up chart 2
//
CloseChart()
```

- The scaling of both charts must be identical — exceptions are charts with two separately arranged y-axes. Since the individual overlayed charts are completely independent from each other and "do not know anything about each other", the scaling for both charts has to be set exactly the same either manually by using the `Scaling()` function or with a small trick: in this concrete example the minimum and maximum values are determined while copying the chart data from the database. Then these extreme values are added to both charts as a second data series. By doing so, it is guaranteed that both charts have the same y-scaling. The second data series is made invisible by using the appropriate style functions. For non-stacked bar charts the invisible second series can also be moved behind the first series by using *seriesGap=-100* so that the bars and labels under the x-axis match exactly.

- Furthermore, it is advantageous to display only the axes and grids of the first "bottom" chart and to suppress them on the second "top" chart.

### Solution:

#### FileMaker Pro fields: (Fig. 18)

In addition to the global fields *gFunctions*, *gChart*, *gError* and *gFunctions-WithPlaceholders* described previously in the introduction, the following fields are also necessary:

- The text field *Name* which contains the labels for the x-axis.
- The number fields *BarChartValue* and *LineChartValue* which contain the respective chart data.
- The global fields *gBarChartValues*, *gLineChartValues* and *gXAxisLabels*. They correspond to the matching placeholders *<gBarChartValues>*, *<gLineChartValues>* and *<gXAxisLabels>* and contain the actual values. There are also the global fields *gMinValue* and *gMaxValue*.
- Optional: the global fields *gLanguage* and *gVersion*. *gVersion* contains the exact name of the presently active xmCHART version, e.g. "xmCHART 3.1".

<u>Field Name</u>	<u>Type</u>	<u>Options</u>
Name	Text	
BarChartValue	Number	
LineChartValue	Number	
gFunctionsWithPlaceholders	Text	Global
gFunctions	Text	Global
gChart	Container	Global
gError	Text	Global
gVersion	Text	Global
gLanguage	Text	Global
gBarChartValues	Text	Global
gLineChartValues	Text	Global
gXAxisLabels	Text	Global
gMinValue	Number	Global
gMaxValue	Number	Global

Fig. 18

#### FileMaker Pro scripts:

- FileMaker Pro script *SetUpChart*: (Fig. 19)

The script *SetUpChart* produces the desired chart and consists of several subscripts.

```

Freeze Window
Perform Script [ “_SetUpMinMaxValues” ]
Perform Script [ “_SetUpData” ]
Perform Script [ “_SetUpXAxisLabels” ]
Perform Script [ “_SubstitutePlaceholders” ]
Perform Script [ “_DrawChart” ]

```

Fig. 19

(1) *\_SetUpMinMaxValues*: (Fig. 20)

The subscript *\_SetUpMinMaxValues* searches for the lowest and highest value in *BarChartValue* and *LineChartValue* and stores them in *gMinValue* and *gMaxValue*.

```

Set Field [ xmCHART::gMinValue; 0 ]
Set Field [ xmCHART::gMaxValue; 0 ]
Go to Record/Request/Page [ First ]
Loop
  If [ not IsEmpty(xmCHART::gBarChartValues) ]
    Set Field [ xmCHART::gMinValue;
      If(GetAsNumber(xmCHART::BarChartValue) <
        GetAsNumber(xmCHART::gMinValue);
        xmCHART::BarChartValue; xmCHART::gMinValue) ]
    Set Field [ xmCHART::gMaxValue;
      If(GetAsNumber(xmCHART::BarChartValue) >
        GetAsNumber(xmCHART::gMaxValue);
        xmCHART::BarChartValue; xmCHART::gMaxValue) ]
  End If
  If [ not IsEmpty(xmCHART::LineChartValue) ]
    Set Field [ xmCHART::gMinValue;
      If(GetAsNumber(xmCHART::LineChartValue) <
        GetAsNumber(xmCHART::gMinValue);
        xmCHART::LineChartValue; xmCHART::gMinValue) ]
    Set Field [ xmCHART::gMaxValue;
      If(GetAsNumber(xmCHART::LineChartValue) >
        GetAsNumber(xmCHART::gMaxValue);
        xmCHART::LineChartValue; xmCHART::gMaxValue) ]
  End If
  Go to Record/Request/Page [ Next; Exit after last ]
End Loop

```

Fig. 20

(2) *\_SetUpData*: (Fig. 21)

The subscript *\_SetUpData* takes the values necessary for drawing the bar chart and line chart from the database. The values are stored in the form of a character string — the individual numbers are separated by spaces. This is done by using a loop over all records. Possible missing values are replaced by *null*. Finally, the extreme values *gMinValue* and *gMaxValue* are added as the second data series. By doing so, it is guaranteed that both charts have the same y-scaling.

```

Set Field [ xmCHART::gBarChartValues; "" ]
Set Field [ xmCHART::gLineChartValues; "" ]
Go to Record/Request/Page [ First ]
Loop
  If [ IsEmpty(xmCHART::BarChartValue) ]
    Set Field [ xmCHART::gBarChartValues;
      xmCHART::gBarChartValues & "null " ]
  Else
    Set Field [ xmCHART::gBarChartValues;
      xmCHART::gBarChartValues &
      GetAsNumber(xmCHART::BarChartValue) & " " ]
  End If
  If [ IsEmpty(xmCHART::LineChartValue) ]
    Set Field [ xmCHART::gLineChartValues;
      xmCHART::gLineChartValues & "null " ]
  Else
    Set Field [ xmCHART::gLineChartValues;
      xmCHART::gLineChartValues &
      GetAsNumber(xmCHART::LineChartValue) & " " ]
  End If
  Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gBarChartValues; xmCHART::gBarChartValues & ";" &
  GetAsNumber(xmCHART::gMinValue) & " " &
  GetAsNumber(xmCHART::gMaxValue) ]
Set Field [ xmCHART::gLineChartValues; xmCHART::gLineChartValues & ";" &
  GetAsNumber(xmCHART::gMinValue) & " " &
  GetAsNumber(xmCHART::gMaxValue) ]

```

Fig. 21

**(3) *\_SetUpXAxisLabels*:** (Fig. 22)

The subscript *\_SetUpXAxisLabels* takes the label texts for the x-axis from the database and stores them in the form of a character string in *gXAxisLabels*. The individual names are to be placed in double quotes and separated by semicolons. Please note that when entering texts in the Script Editor the required double quotes must always be entered twice.

```

Set Field [ xmCHART::gXAxisLabels; "" ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [ xmCHART::gXAxisLabels;
                xmCHART::gXAxisLabels & "\"" & xmCHART::Name & "\""; ]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gXAxisLabels;
            Left(xmCHART::gXAxisLabels; Length(xmCHART::gXAxisLabels)-1) ]

```

Fig. 22

**(4) *\_SubstitutePlaceholders*:** (Fig. 23)

The subscript *\_SubstitutePlaceholders* replaces the placeholders in *gFunctionsWithPlaceholders* (Fig. 24) with the actual values.

```

Set Field [ xmCHART::gFunctions;
            Substitute(xmCHART::gFunctionsWithPlaceholders;
                      "<gMinValue>"; GetAsNumber(xmCHART::gMinValue)) ]
Set Field [ xmCHART::gFunctions;
            Substitute(xmCHART::gFunctions;
                      "<gBarChartValues>"; xmCHART::gBarChartValues) ]
Set Field [ xmCHART::gFunctions;
            Substitute(xmCHART::gFunctions;
                      "<gLineChartValues>"; xmCHART::gLineChartValues) ]
Set Field [ xmCHART::gFunctions;
            Substitute(xmCHART::gFunctions;
                      "<gXAxisLabels>"; xmCHART::gXAxisLabels) ]

```

Fig. 23

```

OpenDrawing(400;300)
// BAR CHART
OpenChart(40;20;340;250;on)
    ChartData(<gBarChartValues>)
    BarChart(label;;-100) // -100 move 2nd series
    AxisMajorTicks(all;0) // hide tick marks
    AxisMajorTickLabelTexts(x;<gXAxisLabels>)
    AxisMajorTickLabelStyle(all;"Verdana";10)
    // grid
    MajorGridLineWidths(y;x;0) // hide vertical grid
    MajorGridLineColors(all;all;lightGray)
    MajorGridStripeColors(x;y;230 230 230;250 250 250)
    GridFrame()
    LabelStyle(1;"Verdana";10)
    // hide 2nd series
    BorderStyle(2;none)
    FillStyle(2;;transparent)
    LabelTexts(2;"")
CloseChart()

// LINE CHART
OpenChart(40;20;340;250;on)
    ChartData(<gLineChartValues>)
    LineChart(label;on)
    LineStyle(1;poly;2;250 200 0)
    LabelStyle(1;"Verdana";10)
    LabelBackground(1;lightYellow)
    LabelOptions(1;centerCenter)
    AxisOptions(all;none) // hide axes
    GridLocation(;none) // hide grid
    // hide 2nd series
    LineStyle(2;none)
    LabelTexts(2;"")
CloseChart()
CloseDrawing()

```

Fig. 24

(5) *\_DrawChart*: (Fig. 25)

In the subscript *\_DrawChart* all functions included in *gFunctions* are sent to xmCHART by the external function *xmCH\_DrawChart()*. If no error occurs, the chart is then stored in the container field *gChart*.

```
Set Field [ xmCHART::gChart; xmCH_DrawChart( xmCHART::gFunctions ) ]
Set Field [ xmCHART::gError; xmCH_GetErrorMessage( "103" ) ]
```

Fig. 25

### Result:

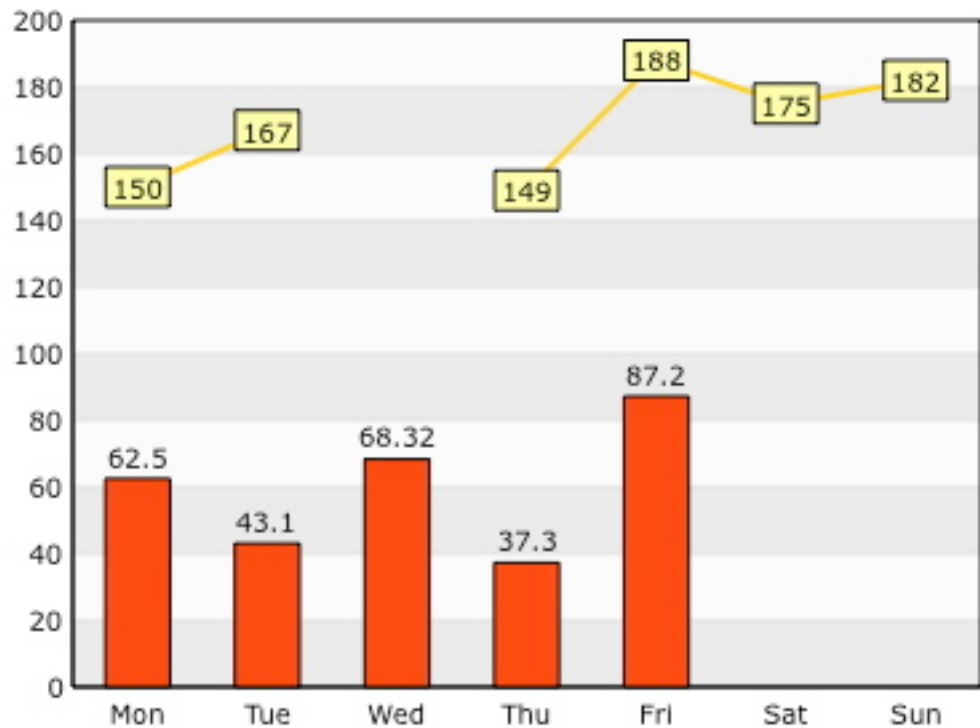


Fig. 26

### Variations:

As an option, a separate scaling can be used for bar and line charts. For example, the y-axis of the bar chart is drawn on the left (default) and the y-axis of the line chart on the right side - the y-axis is shifted to the right by using the *AxisOptions(y;;on)* function. An example of this can be found in *xmGALLERY.FP7*.

## Example 4

In the fourth example, the field contents and their sums are to be evaluated graphically. The corresponding FileMaker Pro file is xmEXAMPLE4.FP7.

### Problem:

Let's assume you want to graphically evaluate the turnover of your company. On the one hand, the turnover should be evaluated quarterly; on the other hand, it should also be categorized according to employees and represented in the form of a stacked bar chart.

For this purpose, you maintain an employee database in which the turnover of the individual employees is recorded per quarter. One record is set up for each employee with the fields *Employee\_Name*, *Turnover\_Q1*, *Turnover\_Q2*, *Turnover\_Q3*, *Turnover\_Q4*, as well as the summary fields *Total\_Turnover\_Q1*, *Total\_Turnover\_Q2*, *Total\_Turnover\_Q3*, *Total\_Turnover\_Q4* and the formula field *Employee\_Turnover*, which calculates the total turnover of an employee within one year, followed by the evaluation field *Total\_Turnover*, which contains the sum of all turnover within one year.

### Solution:

#### FileMaker Pro fields: (Fig. 27)

In addition to the data fields described previously, the following fields are also necessary:

- The three "xmCHART Globals" *gFunctions*, *gChart* and *gError*.
- Optional: the global fields *gLanguage* and *gVersion*. *gVersion* contains the exact name of the presently active xmCHART version, e.g. "xmCHART 3.1".

<u>Field Name</u>	<u>Type</u>	<u>Options</u>
Employee_Name	Text	
Turnover_Q1	Number	
Turnover_Q2	Number	
Turnover_Q3	Number	
Turnover_Q4	Number	
Total_Turnover_Q1	Summary	= Total of Turnover_Q1
Total_Turnover_Q2	Summary	= Total of Turnover_Q2
Total_Turnover_Q3	Summary	= Total of Turnover_Q3
Total_Turnover_Q4	Summary	= Total of Turnover_Q4
Employee_Turnover	Calculation	= Turnover_Q1+...+Turnover_Q4
Total_Turnover	Summary	= Total of Employee_Turnover
gFunctions	Text	Global
gChart	Container	Global
gError	Text	Global
gVersion	Text	Global
gLanguage	Text	Global

Fig. 27

**Turnover per quarter:**

- FileMaker Pro script *SetUpFunctions\_TurnoverPerQuarter*:

The script *SetUpFunctions\_TurnoverPerQuarter* is made up of three sub-scripts: (Fig. 28)

**Freeze Window**

```
Set Field [ xmCHART::gFunctions; "OpenDrawing(420;300;;2)¶" ]
Perform Script [ "_ChartData_TurnOverPerQuarter" ]
Perform Script [ "_Functions_TurnOverPerQuarter" ]
Perform Script [ "_Legendtexts_TurnOverPerQuarter" ]
```

Fig. 28

**(1) \_ChartData\_TurnoverPerQuarter:**

This subscript sets up the `ChartData()` function. In order to determine the turnover per quarter, the employee turnover is added to `ChartData()` by looping all records. (see Fig. 29)

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "ChartData(" ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [xmCHART::gFunctions; xmCHART::gFunctions &
        GetAsNumber(xmCHART::Turnover_Q1) & " " &
        GetAsNumber(xmCHART::Turnover_Q2) & " " &
        GetAsNumber(xmCHART::Turnover_Q3) & " " &
        GetAsNumber(xmCHART::Turnover_Q4) & "; " ]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gFunctions;
    Left(xmCHART::gFunctions;Length(xmCHART::gFunctions)-1)& ")" ]

```

Fig. 29

(2) *\_Functions\_TurnoverPerQuarter:*

Other necessary function calls in addition to `ChartData( )` are added to *gFunctions* as represented in Fig. 30.

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
    "BarChart(stacked;;;30)¶" &
    "BorderStyle(all;none)¶" &
    "AxisMajorTickLabelTexts(x;\Q1\";\Q2\";\Q3\";\Q4\")¶" &
    "TitleText(\"Turnover per Quarter\")¶" &
    "AddText(320;250;\Total Turnover:\n" &
    GetAsNumber(xmCHART::Total_Turnover) & "\";;;bold)¶" &
    "SetThousandsSep(\",\")¶" ]

```

Fig. 30

(3) *\_Legndtexts\_TurnoverPerQuarter:*

Since the legend should contain both the names of the employees and their turnover, the `LegendTexts( )` function is set up dynamically. That means the names of the employees and all of their turnover are added to the `LegendTexts( )` function by looping all records. (Fig. 31)

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "LegendTexts(" ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [xmCHART::gFunctions; xmCHART::gFunctions & "\" &
        xmCHART::Employee_Name & " (" &
        GetAsNumber (xmCHART::Employee_Turnover) & ")\";" ]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gFunctions;
    Left(xmCHART::gFunctions;Length(xmCHART::gFunctions)-1)& ")"]

```

Fig. 31

- FileMaker Pro field *gFunctions*:  
After running the script *SetUpFunctions\_TurnoverPerQuarter*, the global field *gFunctions* contains the following function calls for xmCHART. (see Fig. 32)

```

OpenDrawing(420;300;;2)
    ChartData(32500 62400 49700 42100; 12300 32500 53200;
        23900 52300 53400 45200;30100 58200 56100 0;
        0 0 22000 0)
    BarChart(stacked;;;30)
    AxisMajorTickLabelTexts(x;"Q1";"Q2";"Q3";"Q4")
    TitleText("Turnover per Quarter")
    AddText(320;250;"Total Turnover:\n625900";;;bold)
    SetThousandsSep(",")
    LegendTexts("Empl. 1 (186700)";"Empl. 2 (98000)";
        "Empl. 3 (174800)";"Empl. 4 (144400)";
        "Empl. 5 (22000)")

```

Fig. 32

- FileMaker Pro field *gChart*:  
The chart created by xmCHART and then stored in *gChart* is represented in Fig. 33.

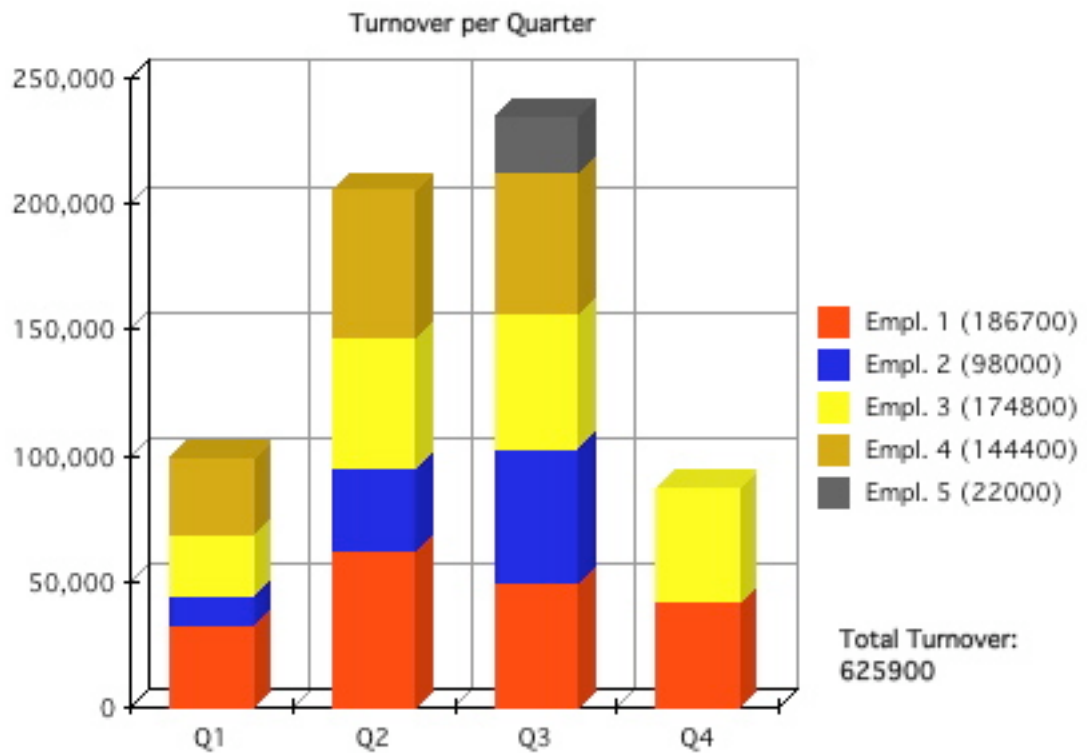


Fig. 33

**Turnover per employee:**

- FileMaker Pro script *SetUpFunctions\_TurnoverPerEmployee*:  
The script *SetUpFunctions\_TurnoverPerEmployee* is made up of four sub-scripts: (Fig. 34)

**Freeze Window**

```
Set Field [ xmCHART::gFunctions; "OpenDrawing(420;300;;2)¶" ]
```

```
Perform Script [ "_ChartData_TurnOverPerEmployee" ]
```

```
Perform Script [ "_Functions_TurnOverPerEmployee" ]
```

```
Perform Script [ "_Legendtexts_TurnOverPerEmployee" ]
```

```
Perform Script [ "_AxisTickLabels_TurnOverPerQuarter" ]
```

Fig. 34

*(1) \_ChartData\_TurnoverPerEmployee:*

The `ChartData ( )` function is set up by this subscript. In order to determine the turnover per employee, the turnover is added to `ChartData ( )` quarterly by looping all employees. That means a total of four data series (one series per quarter) are produced. (see Fig. 35)

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "ChartData(" ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [xmCHART::gFunctions; xmCHART::gFunctions &
        GetAsNumber(xmCHART::Turnover_Q1) & " " ]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "," ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [xmCHART::gFunctions; xmCHART::gFunctions &
        GetAsNumber(xmCHART::Turnover_Q2) & " " ]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "," ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [xmCHART::gFunctions; xmCHART::gFunctions &
        GetAsNumber(xmCHART::Turnover_Q3) & " " ]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "," ]
Go to Record/Request/Page [ First ]
Loop
    Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
        GetAsNumber(xmCHART::Turnover_Q4) & " " ]
    Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gFunctions;
    Left(xmCHART::gFunctions;Length(xmCHART::gFunctions)-1)& ")"]

```

Fig. 35

*(2) \_Functions\_TurnoverPerEmployee:*

Other necessary function calls in addition to `ChartData ( )` are added to `gFunctions` as represented in Fig. 36.

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
  "BarChart(stacked;;;50)¶" &
  "BorderStyle(all;none)¶" &
  "TitleText(\"Turnover per Employee\")¶" &
  "AddText(320;250;\n\"Total Turnover:\n\" &
  GetAsNumber(xmCHART::Total_Turnover) & "\";;;bold)¶" &
  "SetThousandsSep(\",\")¶" ]

```

Fig. 36

*(3) \_Legendtexts\_TurnoverPerEmployee:*

The LegendTexts ( ) function is put together by concatenating text constants and field contents like in Fig. 37.

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "LegendTexts(
  \"1.Quarter (\" & GetAsNumber(xmCHART::Total_Turnover_Q1)&")\"";
  \"2.Quarter (\" & GetAsNumber(xmCHART::Total_Turnover_Q2)&")\"";
  \"3.Quarter (\" & GetAsNumber(xmCHART::Total_Turnover_Q3)&")\"";
  \"4.Quarter (\" & GetAsNumber(xmCHART::Total_Turnover_Q4)&")\"
)¶"]

```

Fig. 37

*(4) \_AxisTickLabels\_TurnoverPerEmployee:*

Since the x-axis tick mark labels should contain the names of the employees, the AxisMajorTickLabelTexts ( ) function is set up dynamically. That means the names of the employees are added to the AxisMajorTickLabelTexts ( ) function by looping all employees. (Fig. 38)

```

Set Field [xmCHART::gFunctions; xmCHART::gFunctions &
  "AxisMajorTickLabelTexts(x;" ]
Go to Record/Request/Page [ First ]
Loop
  Set Field [xmCHART::gFunctions; xmCHART::gFunctions & "\"\" &
    xmCHART::Employee_Name & "\";"]
  Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [xmCHART::gFunctions;
  Left(xmCHART::gFunctions;Length(xmCHART::gFunctions)-1) & ")¶"]

```

Fig. 38

- FileMaker Pro field *gFunctions*:

After running the script *SetUpFunctions\_TurnoverPerEmployee*, the field *gFunctions* contains the following function calls for xmCHART. (Fig. 39)

```
OpenDrawing(420;300;1;2)
  ChartData(32500 23900 30100 0 ;
            62400 12300 52300 58200 0 ;
            49700 32500 53400 56100 22000 ;
            42100 53200 45200 0 0)
  BarChart(stacked;;;50)
  BorderStyle(all;none)
  TitleText("Turnover per Employee")
  AddText(320;250;"Total Turnover:\n625900";;;bold)
  SetThousandsSep(",")
  LegendTexts("1.Quarter (86500)";
              "2.Quarter (185200)";
              "3.Quarter (213700)";
              "4.Quarter (140500)" )
  AxisMajorTickLabelTexts(x;"Empl. 1";"Empl. 2";
                           "Empl. 3";"Empl. 4";
                           "Empl. 5")
```

Fig. 39

- FileMaker Pro field *gChart*:

The chart created by xmCHART and then stored in *gChart* is represented in Fig. 40.

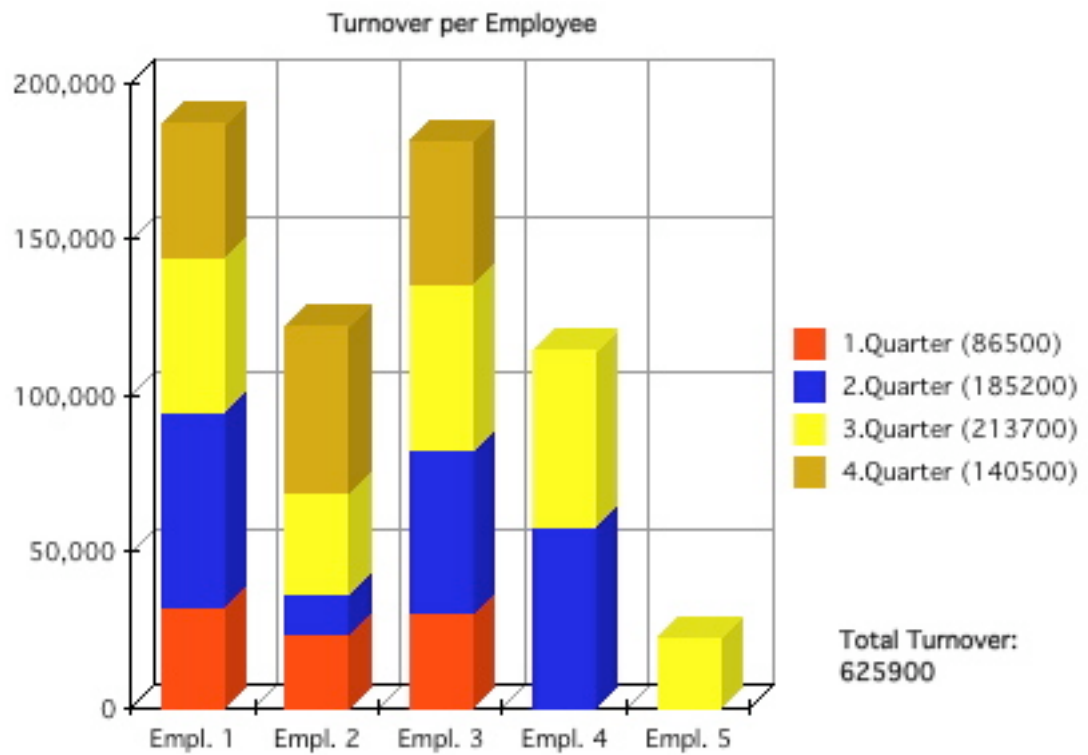


Fig. 40

## Example 5

In the fifth example the field contents are to be evaluated graphically according to the frequency. The corresponding FileMaker Pro database is xmEXAMPLE5.FP7.

### Problem:

Let's assume you have a website and you would like an overview of the distribution of visitor frequency within one day and within one week. In addition, you want to categorize the visitors to your website according to their native countries.

For this purpose, you have set up a FileMaker Pro file with three fields, *Hour*, *Weekday* and *Country*. You have taken the corresponding data from the access protocol file of your website. The number field *Hour* contains a value between 0 and 23 – more detailed information on minutes and seconds do not appear. The number field *Weekday* contains a number between 1 (Sunday) and 7 (Saturday) and the text field *Country* contains a country abbreviation consisting of two characters.

### Solution:

#### FileMaker Pro fields: (Fig. 41)

In addition to the three previously described data fields, *Hour*, *Weekday* and *Country*, the following other fields are also necessary:

- The global fields *gCountry*, *gLabel* and *gCounter*.  
These globals are necessary to set up the functions `ChartData()` and `AxisMajorTickLabelTexts()`.
- The three "xmCHART Globals" *gFunctions*, *gChart* and *gError*.
- Optional: the global fields *gLanguage* and *gVersion*. *gVersion* contains the exact name of the presently active xmCHART version, e.g. "xmCHART 3.1".

<u>Field Name</u>	<u>Type</u>	<u>Options</u>
Hour	Number	Indexed
Weekday	Number	Indexed
Country	Text	
gCountry	Text	Global
gLabel	Text	Global
gCounter	Number	Global
gFunctions	Text	Global
gChart	Container	Global
gError	Text	Global
gVersion	Text	Global
gLanguage	Text	Global

Fig. 41

**Frequency distribution over 24 hours:**

- FileMaker Pro script *SetUpFunctions\_24Hours*:

The script *SetUpFunctions\_24Hours* is made up of two subscripts: (see Fig. 42)

**Freeze Window**

**Set Field [ xmCHART::gFunctions; "OpenDrawing(420;300;;2)¶" ]**

**Perform Script [ "\_ChartData\_24Hours" ]**

**Perform Script [ "\_Functions\_24Hours" ]**

Fig. 42

**(1) \_ChartData\_24Hours:**

The *ChartData()* function is set up by this subscript. In order to determine the number of visitors per hour, records with *Hour=0*, *Hour=1*, *Hour=2* up to *Hour=23* are searched for via a loop and the number of records found is added to *ChartData()*. If no visitors are found within a certain hour interval, it should be noted that the resulting error (error number 401 – no records match this request) must be handled by means of the FileMaker Pro script step *Set Error Capture [On]* and by adding 0 to *ChartData()*. (see Fig. 43)

```

Set Error Capture [ On ]
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "ChartData(" ]
Set Field [ xmCHART::gCounter; 0 ]
Loop
    Enter Find Mode [ ]
    Set Field [ xmCHART::Hour; xmCHART::gCounter ]
    Perform Find [ ]
    If [ Get(LastError) = 401 ]
        Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "0 " ]
    Else
        Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
            Get(FoundCount) & " " ]
    End If
    Exit Loop If [ GetAsNumber(xmCHART::gCounter) = 23 ]
    Set Field [ xmCHART::gCounter; xmCHART::gCounter + 1 ]
End Loop
Set Field [ xmCHART::gFunctions;
    Left(xmCHART::gFunctions;Length(xmCHART::gFunctions)-1)& ")"]
Set Error Capture [ Off ]

```

Fig. 43

(2) *\_Functions\_24Hours:*

Other necessary function calls in addition to `ChartData()` are added to *gFunctions* as represented in Fig. 44.

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
    "BarChart()]" &
    "PictureStyle(1;resource;\\"8\\")]" &
    "TitleText(\"Frequency distribution over 24 hours\\")]" &
    "TitleBackground()]" &
    "GridLocation(;none)]" &
    "AxisMajorTicks(y;3;;;out)]" &
    "AxisMajorTicks(x;0)]" ]

```

Fig. 44

- FileMaker Pro field *gFunctions*:  
After running the script *SetUpFunctions\_24Hours*, the global field *gFunctions* contains the following function calls: (see Fig. 45)

```

OpenDrawing(420;300;;2)
  ChartData(65 46 37 33 24 33 36 47 59 42 41 47
            55 61 62 76 63 68 70 57 62 73 65 66)
  BarChart()
  PictureStyle(1;resource;"8")
  TitleText("Frequency distribution over 24 hours")
  TitleBackground()
  GridLocation(;none)
  AxisMajorTicks(y;3;;;out)
  AxisMajorTicks(x;0)

```

Fig. 45

- FileMaker Pro field *gChart*:

The chart created by xmCHART and then stored in *gChart* is represented in Fig. 46.

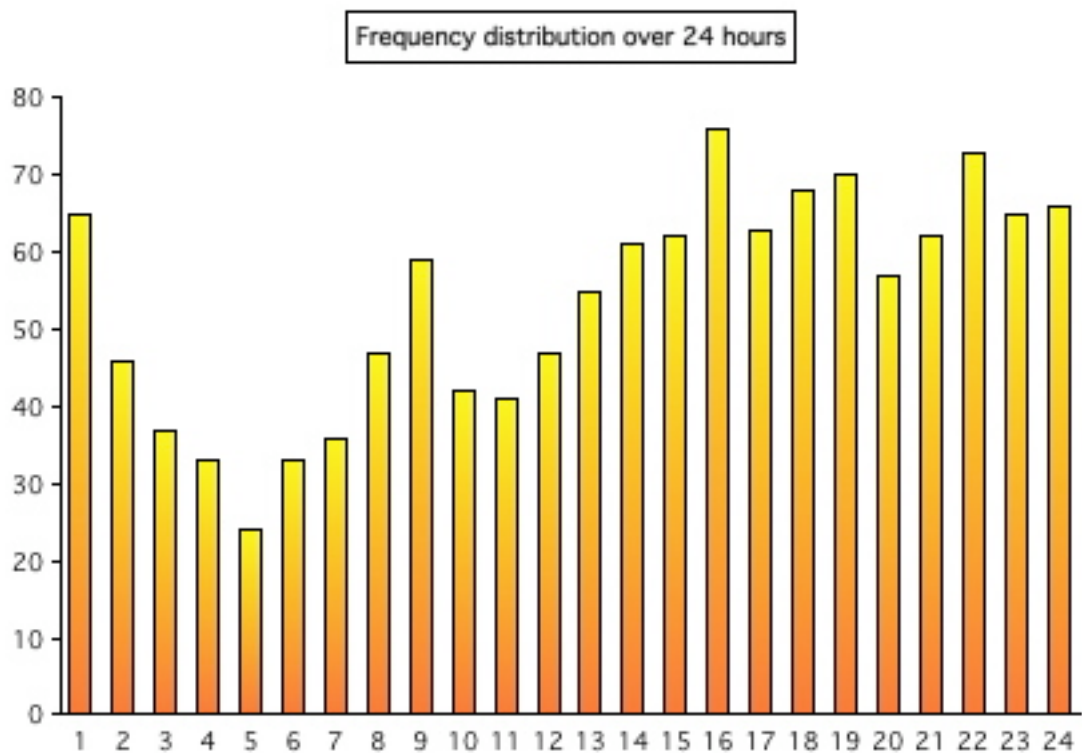


Fig. 46

**Frequency distribution over weekdays:**

- FileMaker Pro script *SetUpFunctions\_Weekdays*:

The script *SetUpFunctions\_Weekdays* is also made up of two subscripts: (see Fig. 47)

**Freeze Window**

```
Set Field [ xmCHART::gFunctions; "OpenDrawing(420;300;;2)¶" ]
Perform Script [ "_ChartData_Weekdays" ]
Perform Script [ "_Functions_Weekdays" ]
```

Fig. 47

*(1) \_ChartData\_Weekdays:*

The *ChartData ( )* function is set up by this subscript. In order to determine the number of visitors per weekday, records with *Weekday=1*, *Weekday=2* up to *Weekday=7* are searched for via a loop and the number of records found is added to *ChartData ( )*. If no visitors are found within a day, it should be noted that the resulting error (error number 401 – no records match this request) must be handled by means of the FileMaker Pro script step *Set Error Capture [On]* and by adding 0 to *ChartData ( )*. (Fig. 48)

```
Set Error Capture [ On ]
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "ChartData(" ]
Set Field [ xmCHART::gCounter; 1 ]
Loop
  Enter Find Mode [ ]
  Set Field [ xmCHART::Weekday; xmCHART::gCounter ]
  Perform Find [ ]
  If [ Get(LastError) = 401 ]
    Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "0 " ]
  Else
    Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
      Get(FoundCount) & " " ]
  End If
  Exit Loop If [ GetAsNumber(xmCHART::gCounter) = 7 ]
  Set Field [ xmCHART::gCounter; xmCHART::gCounter + 1 ]
End Loop
Set Field [ xmCHART::gFunctions;
  Left(xmCHART::gFunctions;Length(xmCHART::gFunctions)-1)& "¶"]
Set Error Capture [ Off ]
```

Fig. 48

*(2) \_Functions\_Weekdays:*

Other necessary function calls in addition to `ChartData()` are added to *gFunctions* as represented in Fig. 49.

```
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
  "BarChart(label;;;50)" &
  "TitleText(\"Frequency distribution over weekdays\")" &
  "TitleBackground(;transparent)" &
  "Background(;119;0)" &
  "AxisMajorTicks(y;3;;;out)" &
  "AxisMajorTicks(x;0)" &
  "AxisMajorTickLabelTexts(x;\\"Sun\\";\\"Mon\\";\\"Tue\\";\\"Wed\\";
    \\"Thu\\";\\"Fri\\";\\"Sat\\")" ]
```

Fig. 49

- FileMaker Pro field *gFunctions*:

After running the script *SetUpFunctions\_Weekdays*, the global field *gFunctions* contains the following function calls: (see Fig. 50)

```
OpenDrawing(420;300;;2)
  ChartData(118 108 222 236 209 214 181)
  BarChart(label;;;50)
  TitleText("Frequency distribution over weekdays")
  TitleBackground(;transparent)
  Background(;119;0)
  AxisMajorTicks(y;3;;;out)
  AxisMajorTicks(x;0)
  AxisMajorTickLabelTexts(x;"Sun";"Mon";"Tue";
    "Wed";"Thu";"Fri";"Sat")
```

Fig. 50

- FileMaker Pro field *gChart*:

The chart created by xmCHART and then stored in *gChart* is represented in Fig. 51.

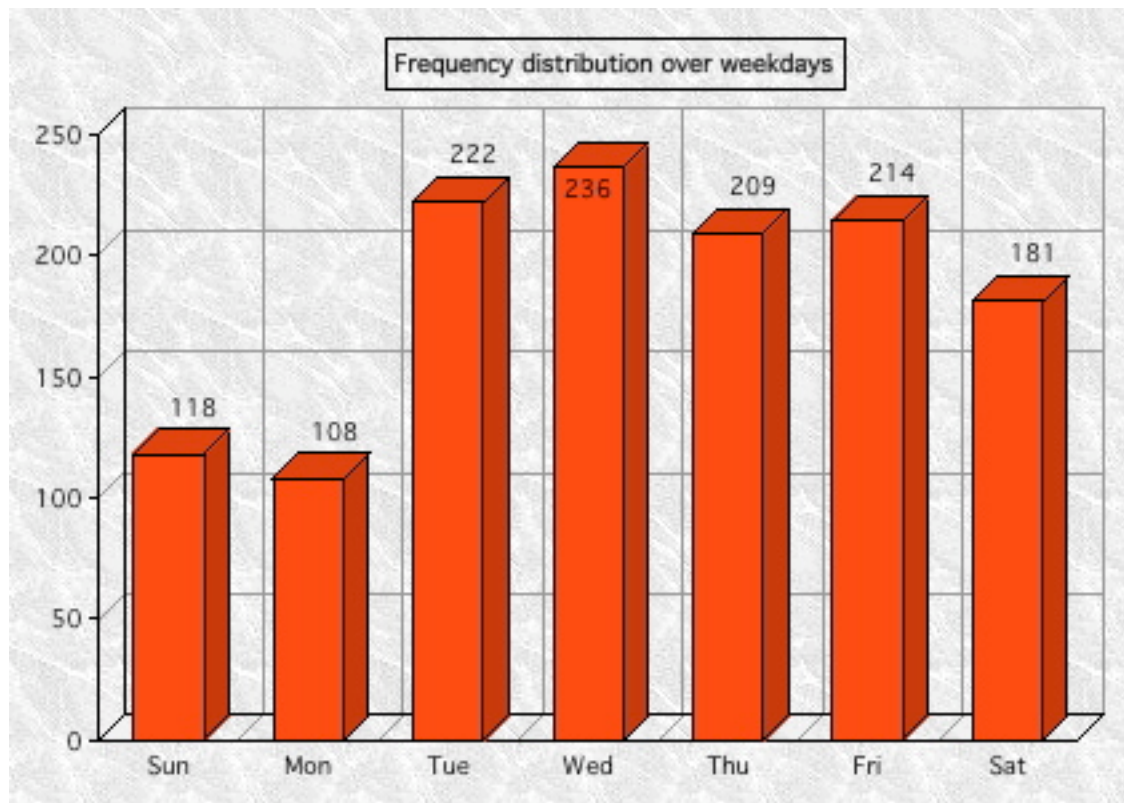


Fig. 51

#### Division according to countries:

- FileMaker Pro script *SetUpFunctions\_Countries*:

The script *SetUpFunctions\_Countries* is also made up of two subscripts: (Fig. 52)

##### Freeze Window

Set Field [ xmCHART::gFunctions; "OpenDrawing(420;300;;2)¶" ]

Perform Script [ "\_ChartData\_Countries" ]

Perform Script [ "\_Functions\_Countries" ]

Fig. 52

##### (1) *\_ChartData\_Countries*:

The functions *ChartData()* and *AxisMajorTickLabelTexts()* are set up by this subscript. In order to classify the visitors according to their native countries, the records are first sorted according to country abbreviations and then the frequencies of the individual country abbreviations are totaled and added to *ChartData()* by looping all records. In addition,

the country abbreviations and their frequencies are stored in the global field *gLabel*. (Fig. 53).

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "ChartData(" ]
Show All Records
Sort Records [ Specified Sort Order: xmCHART::Country; ascending ]
    [ Restore; No dialog ]
Go to Record/Request/Page [ First ]
Set Field [ xmCHART::gLabel; "AxisMajorTickLabelTexts(y;" ]
Set Field [ xmCHART::gCountry; xmCHART::Country ]
Set Field [ xmCHART::gCounter; 1 ]
Loop
    Go to Record/Request/Page [ Next; Exit after last ]
    If [ xmCHART::Country = xmCHART::gCountry ]
        Set Field [ xmCHART::gCounter; xmCHART::gCounter + 1 ]
    Else
        Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
            GetAsNumber(xmCHART::gCounter) & " " ]
        Set Field [ xmCHART::gLabel; xmCHART::gLabel & "\"(" &
            GetAsNumber(xmCHART::gCounter) & ") " &
            xmCHART::gCountry & "\";" ]
        Set Field [ xmCHART::gCountry; xmCHART::Country ]
        Set Field [ xmCHART::gCounter; 1 ]
    End If
End Loop
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
    GetAsNumber(xmCHART::gCounter) & " " ]
Set Field [ xmCHART::gLabel; xmCHART::gLabel & "\"(" &
    GetAsNumber(xmCHART::gCounter) & ") " &
    xmCHART::gCountry & "\""] ]
Set Field [ xmCHART::gCountry; xmCHART::Country ]
Set Field [ xmCHART::gCounter; 1 ]
Set Field [ xmCHART::gFunctions; Left(xmCHART::gFunctions;
    Length(xmCHART::gFunctions) - 1) & ")¶" ]
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
    xmCHART::gLabel & ")¶" ]

```

Fig. 53

(2) *\_Functions\_Countries*:

Other necessary function calls in addition to *ChartData()* and *AxisMajorTickLabelTexts()* are added to *gFunctions* as represented in Fig. 54.

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
    "BarChart(horizontal)¶" &
    "TitleText(\"Division according to countries\")¶" &
    "TitleBackground(;transparent)¶" &
    "BackgroundPict(resource;\"29\")¶" &
    "AxisMajorTicks(x;3;;;out)¶" &
    "AxisMajorTicks(y;0)¶" &
    "GridLocation(;none)¶" ]

```

Fig. 54

- FileMaker Pro field *gFunctions*:

After running the script *SetUpFunctions\_Countries*, the global field *gFunctions* contains the following function calls: (see Fig. 55)

```

OpenDrawing(420;300;1;2)
  ChartData(6 34 76 23 117 54 148 16 23 6 46 64 75 44
            8 48 8 7 47 438)
  AxisMajorTickLabelTexts(y;"(6) AR";"(34) AT";
                          "(76) AU";"(23) BE";
                          "(117) CA";"(54) CH";
                          "(148) DE";"(16) DK";
                          "(23) ES";"(6) FI";
                          "(46) FR";"(64) GB";
                          "(75) IT";"(44) JP";
                          "(8) MX";"(48) NL";
                          "(8) NO";"(7) NZ";
                          "(47) SE";"(438) US")

  BarChart(horizontal)
  TitleText("Division according to countries")
  TitleBackground(;transparent)
  BackgroundPict(resource;"29")
  AxisMajorTicks(x;3;;;out)
  AxisMajorTicks(y;0)
  GridLocation(;none)

```

Fig. 55

- FileMaker Pro field *gChart*:

The chart created by xmCHART and then stored in *gChart* is represented in Fig. 56.

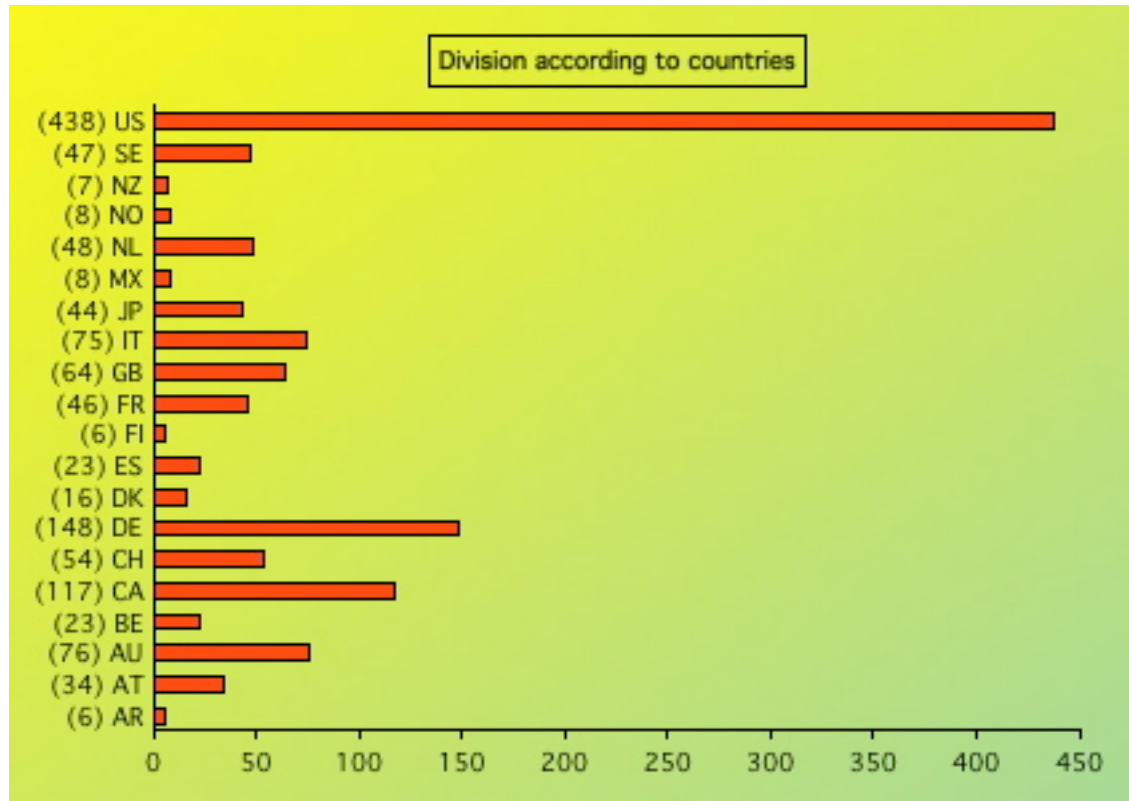


Fig. 56

## Example 6

In the sixth example time-dependent data is evaluated graphically. The corresponding FileMaker Pro file is xmEXAMPLE6.FP7.

### Problem:

Let's assume you record three readings in a 14-day interval which are to be represented graphically by a line chart. Furthermore, the series of readings are to begin at different times.

For this purpose, you set up a FileMaker Pro database with four fields, *Date*, *Series1*, *Series2* and *Series3*. A new record is set up for each day of reading. In this example a 2-dimensional line chart is used which is quite versatile. Thus, for example, data series can begin at different times or, furthermore, it would also be possible to use reading intervals of varying lengths.

### FileMaker Pro fields: (Fig. 57)

In addition to the data fields described previously, the following fields are also necessary:

- The three "xmCHART Globals" *gFunctions*, *gChart* and *gError*.
- Optional: the global fields *gLanguage* and *gVersion*. *gVersion* contains the exact name of the presently active xmCHART version, e.g. "xmCHART 3.1".

<u>Field Name</u>	<u>Type</u>	<u>Options</u>
Date	Date	
Series1	Number	
Series2	Number	
Series3	Number	
gFunctions	Text	Global
gChart	Container	Global
gError	Text	Global
gVersion	Text	Global
gLanguage	Text	Global

Fig. 57

**FileMaker Pro scripts:**

- FileMaker Pro script *SetUpFunctions*:

The script *SetUpFunctions* is made up of two subscripts: (Fig. 58)

**Freeze Window**

```
Set Field [ xmCHART::gFunctions; "OpenDrawing(420;300)¶" ]
Perform Script [ "_ChartData" ]
Perform Script [ "_Functions" ]
```

Fig. 58

**(1) \_ChartData:**

In order to make this process more clear, the script *\_ChartData* calls up three other subscripts in which the *ChartData()* function is set up. (see Fig. 59)

```
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "ChartData(" ]
Perform Script [ "__Series1" ]
Perform Script [ "__Series2" ]
Perform Script [ "__Series3" ]
Set Field [ xmCHART::gFunctions; Left(xmCHART::gFunctions;
Length(xmCHART::gFunctions) - 2) & "¶" ]
```

Fig. 59

The necessary values for representing the first series of readings are extracted by the script *\_\_Series1* shown in Fig. 60. First, the reading dates or, more precisely, the days of the year are added to function *ChartData()* by looping all records. Then the corresponding readings are attached to *ChartData()* by looping all records again.

```
Go to Record/Request/Page [ First ]
Loop
  If [ not IsEmpty(xmCHART::Series1) ]
    Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
DayOfYear(xmCHART::Date) & " " ]
  End If
  Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & "¶" ]
Go to Record/Request/Page [ First ]
Loop
  If [ not IsEmpty(xmCHART::Series1) ]
```

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
GetAsNumber(xmCHART::Series1) & " " ]
End If
Go to Record/Request/Page [ Next; Exit after last ]
End Loop
Set Field [ xmCHART::gFunctions; xmCHART::gFunctions & ";" ]

```

Fig. 60

(2) *\_Functions:*

Other necessary function calls in addition to `ChartData()` are added to *gFunctions* as represented in Fig. 61.

```

Set Field [ xmCHART::gFunctions; xmCHART::gFunctions &
"LineChart2D()" &
"LineStyle(all;;2)" &
"LineStyle(3;poly;2;0 150 150)" &
"LegendTexts(\"Series A\";\"Series B\";\"Series C\")" &
"Background (240 240 255;;0)" &
"Scaling(x;linear;1 ;365;1 2;1 )" &
"Scaling(y;linear;400;600;4;5)" &
"AxisMajorTicks(all;0)" &
"AxisMinorTicks(all;0)" &
"AxisMajorTickLabelTexts(x;\"1 Jan\";\"n1 Feb\";
\"1 Mar\";\"n1 Apr\";\"1 May\";\"n1 Jun\";
\"1 Jul\";\"n1 Aug\";\"1 Sep\";\"n1 Oct\";
\"1 Nov\";\"n1 Dec\";\"31 Dec\")" ]

```

Fig. 61

- FileMaker Pro field *gFunctions*:  
After running the script *SetUpFunctions*, the global field *gFunctions* contains the following function calls: (Fig. 62)

```

OpenDrawing(420;300)
  ChartData( 1 15 29 43 57 71 85 99 113 127 141
            155 169 183 197 211 225 239 253 267 281 295;
            428 531 493 473 472 473 442 443 467 454 442
            475 485 472 489 499 492 512 510 501 519 532;
            71 85 99 113 127 141 155 169 183 197 211
            225 239 253 267 281 295;
            540 572 567 557 554 563 561 571 572 573 564
            542 555 567 572 575 577;
            113 127 141 155 169 183 197 211 225 239 253
            267 281 295;
            500 505 510 520 525 521 510 512 496 490 489
            483 489 479)
  LineChart2D()
  LineStyle(all;poly;2)
  LineStyle(3;poly;2;0 150 150)
  LegendTexts("Series A";"Series B";"Series C")
  Background(240 240 255;;0)
  Scaling(x;linear;1;365;12;1)
  Scaling(y;linear;400;600;4;5)
  AxisMajorTicks(all;0)
  AxisMinorTicks(all;0)
  AxisMajorTickLabelTexts(x;"1 Jan";"\n1 Feb"; "1 Mar";
                          "\n1 Apr"; "1 May";"\n1 Jun";
                          "1 Jul";"\n1 Aug"; "1 Sep";
                          "\n1 Oct"; "1 Nov";"\n1 Dec";
                          "31 Dec")

```

Fig. 62

- FileMaker Pro field *gChart*:

The chart created by xmCHART and then stored in *gChart* is represented in Fig. 63.

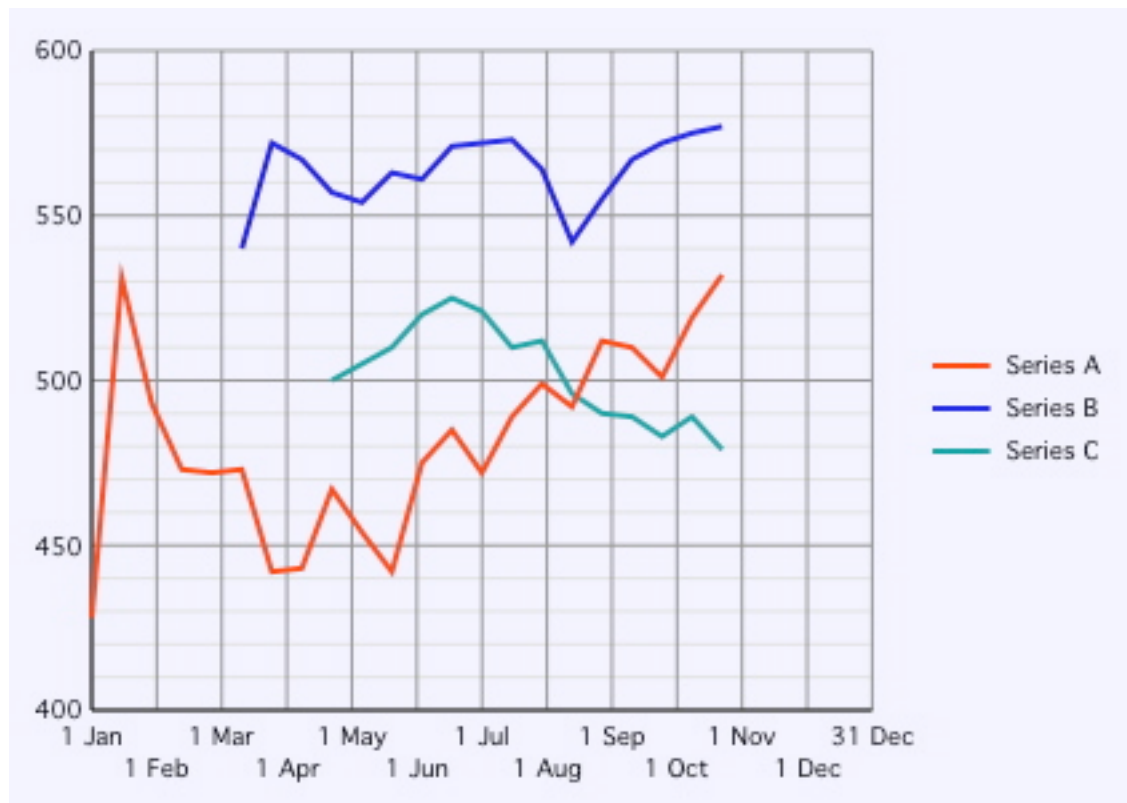


Fig. 63