

LibreOffice
The Document Foundation

Base Handbook

Chapter 6
Reports

Copyright

This document is Copyright © 2013 by its contributors as listed below. You may distribute it and/or modify it under the terms of either the GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), version 3 or later, or the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), version 3.0 or later.

All trademarks within this guide belong to their legitimate owners.

Contributors

Robert Großkopf
Hazel Russman

Jost Lange
Jean Hollis Weber

Jochen Schiffers

Feedback

Please direct any comments or suggestions about this document to:
documentation@global.libreoffice.org.

Caution



Everything you send to a mailing list, including your email address and any other personal information that is in the mail, is publicly archived and can not be deleted.

Acknowledgments

This chapter is based on an original German document and was translated by Hazel Russman.

Publication date and software version

Published 22 April 2013. Based on LibreOffice 3.5.

Note for Mac users

Some keystrokes and menu items are different on a Mac from those used in Windows and Linux. The table below gives some common substitutions for the instructions in this chapter. For a more detailed list, see the application Help.

<i>Windows or Linux</i>	<i>Mac equivalent</i>	<i>Effect</i>
Tools > Options menu selection	LibreOffice > Preferences	Access setup options
<i>Right-click</i>	<i>Control+click</i>	Opens a context menu
<i>Ctrl (Control)</i>	⌘ (<i>Command</i>)	Used with other keys
<i>F5</i>	<i>Shift+⌘+F5</i>	Opens the Navigator
<i>F11</i>	<i>⌘+T</i>	Opens the Styles and Formatting window

Contents

Copyright	2
Contributors	2
Feedback	2
Acknowledgments	2
Publication date and software version	2
Note for Mac users	2
Creating reports using the Report Builder	4
The user interface of the Report Builder	4
General properties of fields	11
Special properties of graphical controls	13
Incorporating charts into the report	14
Data properties of fields	15
Functions in the Report Builder	16
Entering formulas	16
User-defined functions	22
Formula entry for a field	23
Conditional print	23
Conditional formatting	23

Creating reports using the Report Builder

Reports are used to present data in a way that makes it readily understood by people without knowledge of the database. Reports can:

- Present data in easy-to-read tables
- Create charts for displaying data
- Make it possible to use data for printing labels
- Produce form letters such as bills, recall notices, or notifications to people joining or leaving an association

To create a report requires careful preparatory work on the underlying database. Unlike a form, a report cannot include subreports and thus incorporate additional data sources. Neither can a report present different data elements than those that are available in the underlying data source, as a form can do using list boxes.

Reports are best prepared using queries. In this way all variables can be determined. In particular, if sorting within the report is required, always use a query that makes provision for sorting. This means that queries in direct SQL mode should be avoided under these conditions. If you must use a query of this type in your database, you can carry out the sort by first creating a view from the query. Such a view can always be sorted and filtered using the graphical user interface (GUI) of Base.

Caution



When using the Report Builder, you should frequently save your work during editing. In addition to saving within the Report Builder itself after each significant step, you should also save the whole database.

Depending on the version of LibreOffice that you are using, the Report Builder can sometimes crash during editing.

The functionality of completed reports is not affected even if they were created under another version, in which the problem does not occur.

The user interface of the Report Builder

To start the Report Builder from within Base, use **Reports > Create Report in Design View**.

The initial window of the Report Builder (Figure 1) shows three parts. On the left is the current division of the report into Page header, Detail, and Page footer; in the middle are the corresponding areas where the content will be entered; and, to the right, the properties of these regions are shown.

At the same time the Add fields dialog is displayed. This dialog corresponds to the one in form creation. It creates fields with their corresponding field labels.

Without content from the database, a report has no proper function. For this reason, the dialog opens at the Data tab. Here you can set the content of the report; in the example it is the *View_Report_Recall* table. As long as **Analyze SQL command** is set to Yes, the report can be subjected to sorting, grouping, and filtering. A view has been chosen for the basis of this report, so no filter will be applied; it has already been included in the query underlying the view.

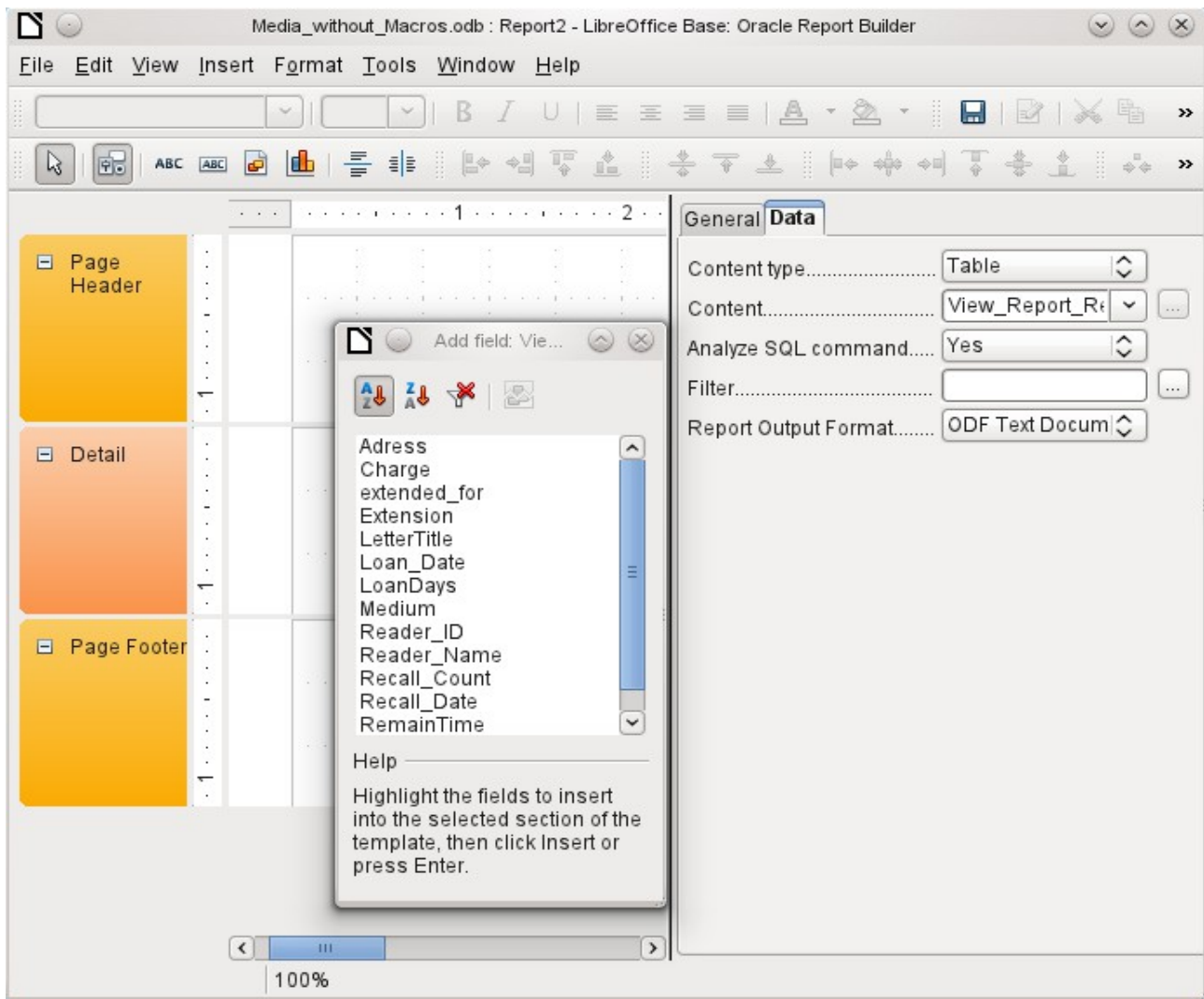










































Figure 1: Initial window of Report Builder

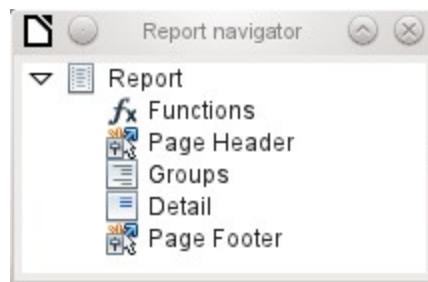
Two output formats for reports are available for selection: *ODF Text document* (a Writer document) or *ODF Spreadsheet* (a Calc document). If you just want a tabular view of your data, the Calc document should definitely be chosen for your report. It is significantly faster to create and is also easier to format subsequently, as there are fewer options to consider and columns can easily be dragged to the required width afterward.

By default, the Report Builder looks for its data source in the first table in the database. This ensures that at least a test of the functions is possible. A data source has to be chosen before the report can be provided with fields.

The Report Builder provides a lot of additional buttons, so the table on the next page shows the buttons with their descriptions. The buttons for aligning elements are not further described in this chapter. They are useful for quick adjustment of fields in a single area of the Report Builder, but in principle everything can be done by direct editing of field properties.

Buttons for editing content	Buttons for aligning elements
<ul style="list-style-type: none">  Save  Edit File  Cut  Copy  Paste  Change property 'Command'  Redo  Add Field  Report Navigator  Sorting and Grouping  Execute Report  LibreOffice Help <input type="checkbox"/> What's This? 	<ul style="list-style-type: none">  Left Align on Section  Right Align on Section  Top Align on Section  Bottom Align on Section <hr/> <ul style="list-style-type: none">  Shrink  Shrink from top  Shrink from bottom <hr/> <ul style="list-style-type: none">  Fit to smallest width  Fit to greatest width  Fit to smallest height  Fit to greatest height
<ul style="list-style-type: none">  Select  Properties  Label Field  Text Box  Graphic  Chart  Horizontal Line  Vertical Line  Display Grid  Snap to Grid  Helplines While Moving 	<ul style="list-style-type: none">  Left  Centered  Right <hr/> <ul style="list-style-type: none">  Top  Center  Bottom

Just as with forms, it is helpful to use the appropriate navigator. So, for example, a careless click at the start of the Report Builder can make it difficult to find the properties of the data for the report. Such data may only be reachable through the report navigator. Left-click on *Report* and the properties of the report are once more accessible.



Initially the navigator shows, in addition to the visible sections of the document (Page header, Groups, Detail, and Page footer), the possibility of including functions. Groups can be used, for example, to assign all media being recalled to the person who has borrowed them, to avoid multiple recall notices. Detail areas show the records belonging to a group. Functions are used for calculations such as sums.

To obtain useful output in the above example, the content of the view must be reproduced with suitable grouping. Each reader should be linked to the recall notices for all of their loaned and overdue media.

View > Sorting and Grouping or the corresponding button starts the grouping function.

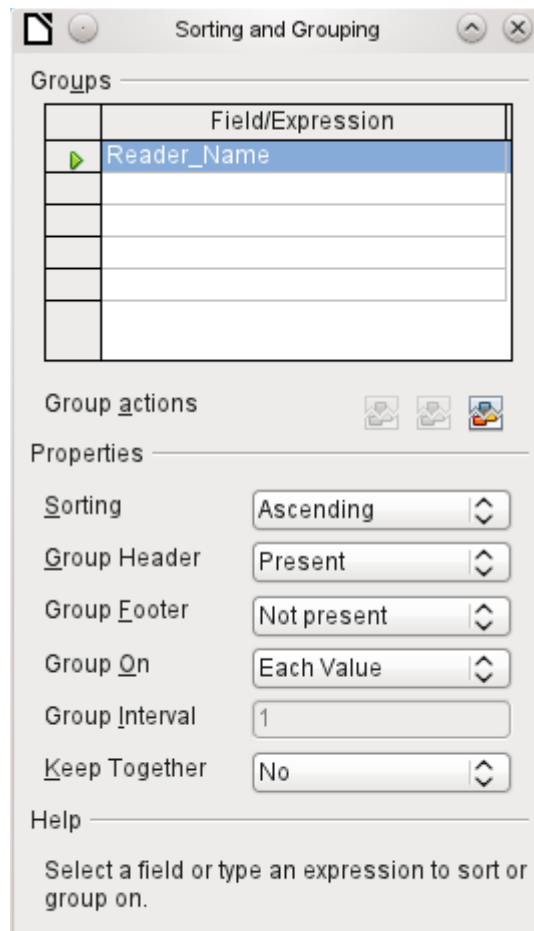


Figure 2: Sorting and Grouping

Here grouping and sorting are by the *Reader_Name* field. Additional fields could also be included in the table above. For example, if you also want to group and sort by the *Loan_Date* field, choose this as the second line.

Directly under the table, several grouping actions are available for selection. You can move a group up or down the list or completely remove it. As only one group is necessary for the planned report, Figure 2 shows only the *Delete* symbol at the extreme right of the group actions as available.

The *Sorting* property is self-explanatory.

When the entry was created, the left side of the Report Builder immediately showed a new division. Next to the field description *Reader_Name* you can now see *Header*. This section is for the group header in the report. The header might contain the name of the person who will receive the recall notice. In this case there is no group footer. Such a footer could contain the fine due, or the place and current date and a space for the signature of the person sending the notice.

By default there is a new group for each value. So if the *Reader_Name* changes, a new group is started. Alternatively you can group by initial letter. In the case of a recall notice, however, this would put all readers with the same initial together in one group. Schmidt, Schulze, and Schulte would receive a common recall notice, which would be quite pointless in this example.

When grouping by initial letter, you can additionally specify how many letters later the next group should begin. One can imagine for example a grouping for a small telephone directory. According to the size of the contact list, one might imagine a grouping on every second initial letter. So A and B would form the first group, then C and D, and so on.

A group can be set either to be kept together with the first details section, or, as far as possible, as a complete group. By default, this option is set to *No*. For recall notices, you would probably want the group to be arranged so that a separate page is printed for each person who is to receive a recall letter. In another menu, you can choose that each group (in this case, each reader name) be followed by a page break before dealing with the next value.

If you have chosen to have a group header and perhaps a group footer, these elements will appear as sections in the report navigator under the corresponding fieldname *Reader_Name*. Here too you have the possibility of using functions, which will then be limited to this group.

To add fields, use the *Add field* function, as with forms. However in this case, the label and the field contents are not tied together. Both can be independently moved, changed in size and dragged to different sections.

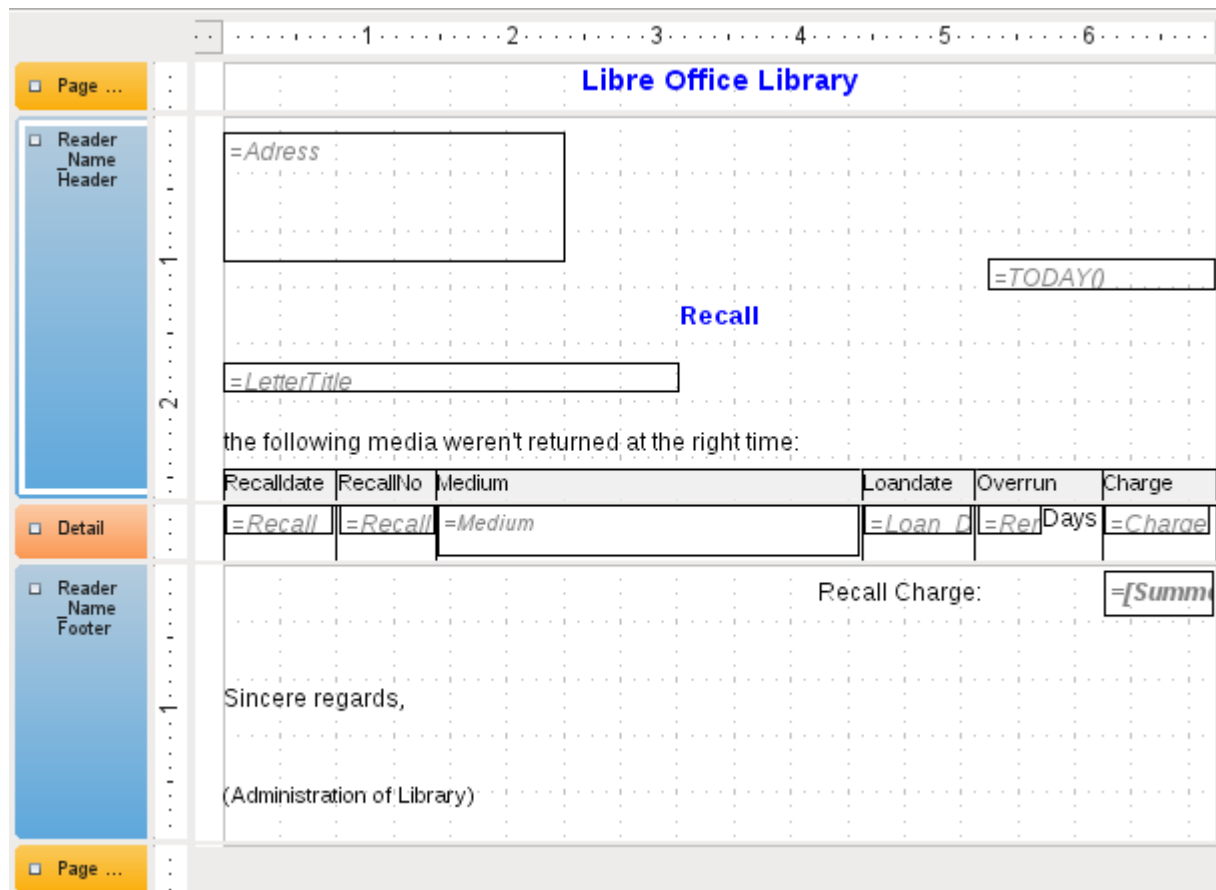


Figure 3: Report design for example recall notice

Figure 3 shows the report design for the recall notice. In the page header is the heading Libre Office Library, inserted as a label field. Here you could also have a letterhead with a logo, since the incorporation of graphics is possible. The fact that this level is called *Page header* does not imply that there is no space above it. That depends on the page settings; if an upper margin has been set, it lies above the page header.

Reader_Name header is the header for the grouped and sorted data. In the fields that are to contain data, the names of the corresponding data fields are shown in light gray. So, for example, the view underlying the report has a field named Address, containing the complete address of the recipient with street and town. To put this into a single field requires line breaks in the query. You can use **CHAR(13)** to create them.

Example:

```
SELECT "FirstName"||' '||"LastName"||CHAR(13)||"Street"||' '||"No"||
CHAR13||"Postcode"||' '||"Town" FROM "FirstName"
```

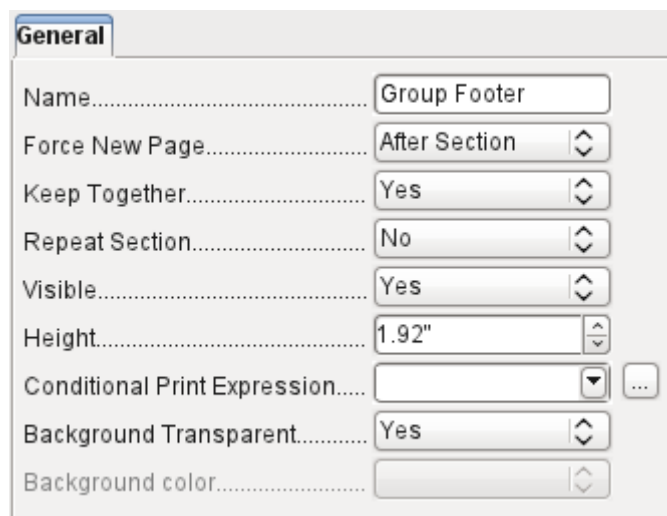
The `=TODAY()` field represents a built-in function, which inserts the current date into this position.

In *Reader_Name header*, in addition to the salutation, we see the column headings for the following table view. These elements should appear only once, even if several media are listed.

In the background of these column headers is a gray rectangle, which also serves as a frame for the data.

The details area is repeated as often as there are separate records with the same *Reader_Name* data. Here are listed all media that have not been returned on time. There is another rectangle in the background to frame the contents. This rectangle is filled with white rather than gray.

Note	<p>In principle LibreOffice provides for the possibility of adding horizontal and vertical lines. These are displayed in design mode. At present they do not appear when the report is complete.</p> <p>These lines have the further disadvantage that they are interpreted only as hairlines. They can be reproduced better if rectangles are used. Set the background of the rectangle to black and the size to, for example, 17cm wide and 0.03cm high. This will create a horizontal line with a thickness of 0.03cm and a length of 17cm.</p>
-------------	--



The *Reader_Name footer* closes the letter with a greeting formula and an area for the signature. The footer is so defined that an additional page break will occur after this area. Also, contrary to the default setup, it is specified that this area should be kept together in all cases. After all, it would look rather odd if many recall notices had the signature on a separate page.

Keep together refers here to the page break. If you want the content of a record to be kept together independently of the break, this is only possible at present if the record is not read in as Details but is used as the basis for a grouping. You can choose *Keep together* = Yes, but it does not work; the Details area becomes separated. You have to put the content of Details in a separate group to keep it together.

A built-in function is used for calculating the total fines.

Below is what an actual recall notice would look like. The details area contains 5 media that the reader has taken out on loan. The group footer contains the total fine due.

Libre Office Library

Mr.
Heinrich Müller
Nowhereroad 14 b
GB 3P67Q Downtown

11/24/12

Recall

Dear Mr. Müller,

the following media weren't returned at the right time:

Recalldate	RecallNo	Medium	Loandate	Overrun	Charge
24.11.12	1	5 - I hear you knocking - by Edmunds, Dave	29.04.12	202Days	\$7.00
24.11.12	1	8 - Im Augenblick - by van Veen, Herman	22.04.12	209Days	\$7.25
24.11.12	1	2 - Eine kurze Geschichte der Zeit - by Hawking, Steven W.	04.04.12	213Days	\$7.50

Recall Charge: **\$21.75**

Sincere regards,

(Administration of Library)

Page 1 / 2 | Default | STD | Group_Header:A1 | 75%

Note

Reports for single records can also extend over more than one page. The size of the report is quite separate from the page size. However, stretching the details area over more than one page can lead to faulty breaks. Here the Report Builder still has problems in calculating the spacing correctly. If both grouping areas and graphical elements are included, this may result in unpredictable sizes for certain areas.

So far individual elements can be moved to positions outside the size of a single page only with the mouse and cursor keys. The properties of the elements always provide the same maximum distance from the upper corner of any area that lies on the first page.

General properties of fields

There are only three types of field for the presentation of data. In addition to text fields (which, contrary to their name, can also contain numbers and formatting), there is also a field type that can contain images from the database. The chart field displays a summary of data.

Property	Value
Name	Charge
Position X	6.12"
Visible	Yes
Position Y	0.00"
Width	0.73"
Height	0.20"
Print repeated values	Yes
Conditional Print Expression	
Mouse wheel scroll	When focused
Print When Group Change	No
Background Transparent	Yes
Background color	
Font	Liberation Sans, Reg
Horz. Alignment	Left
Vert. Alignment	Top
Formatting	\$1,234.57

As with forms, fields are given names. By default, the name is that of the underlying database field.

A field can be set to be invisible. This may seem a bit pointless in the case of fields but is useful for group headers and footers, which may be required to carry out other functions of the grouping without containing anything that needs to be displayed.

If *Print repeated values* is deactivated, display of the field is inhibited when a field with the same content is loaded directly before. This functions correctly only for data fields that contain text. Numeric fields or date fields ignore the deactivation instruction, Label fields are completely faded out when deactivated, even if they occur only once.

In the Report Builder the display of certain content can be inhibited by using *Conditional Print Expression* or the value of the field can be used as a base for formatting text and background. More on conditional expressions is given in "Conditional print" on page 23.

The setting for the mouse wheel has no effect because report fields are not editable. It seems to be a leftover from the form definition dialog.

The *Print When Group Change* function could not be reproduced in reports either.

If the background is not defined as transparent, a background color can be defined for each field.

The other entries deal with the internal content of the field in question. This covers the font (for font color, font weight, and so on, see Figure 4), the alignment of the text within the field, and formatting with the corresponding Character dialog (see Figure 5).

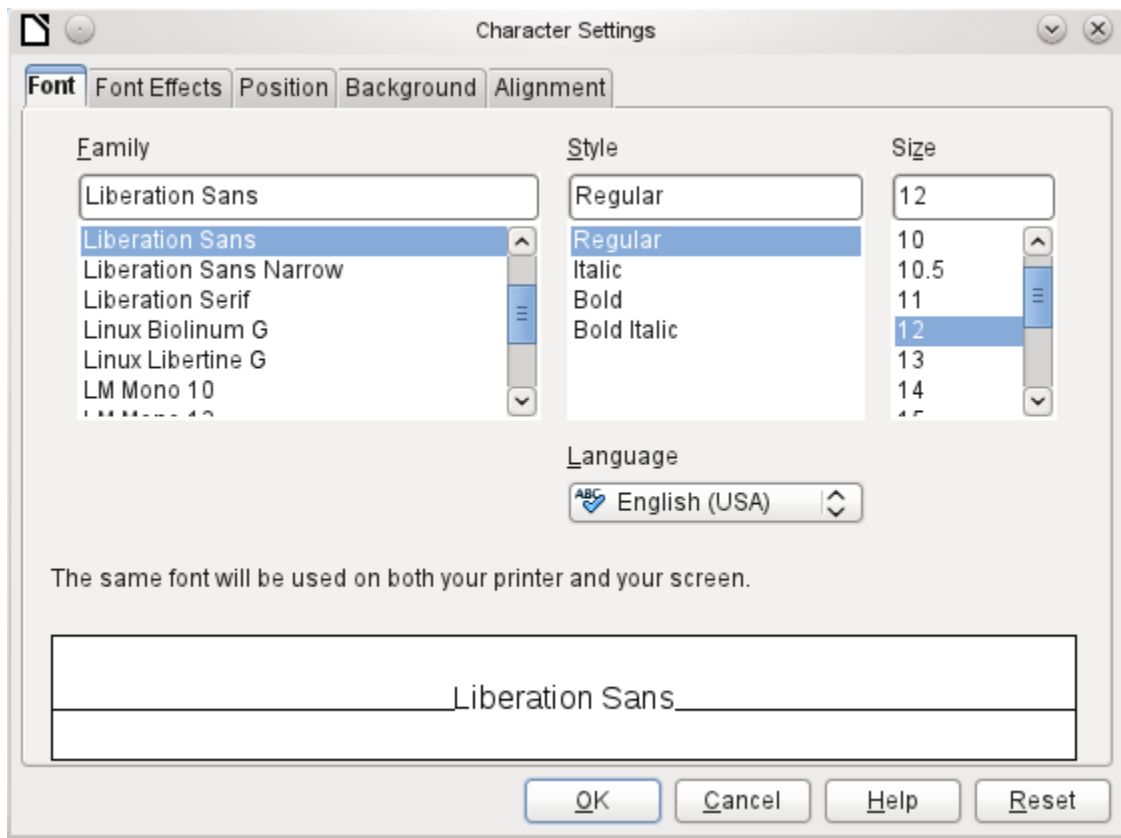


Figure 4: Fonts: Character Settings

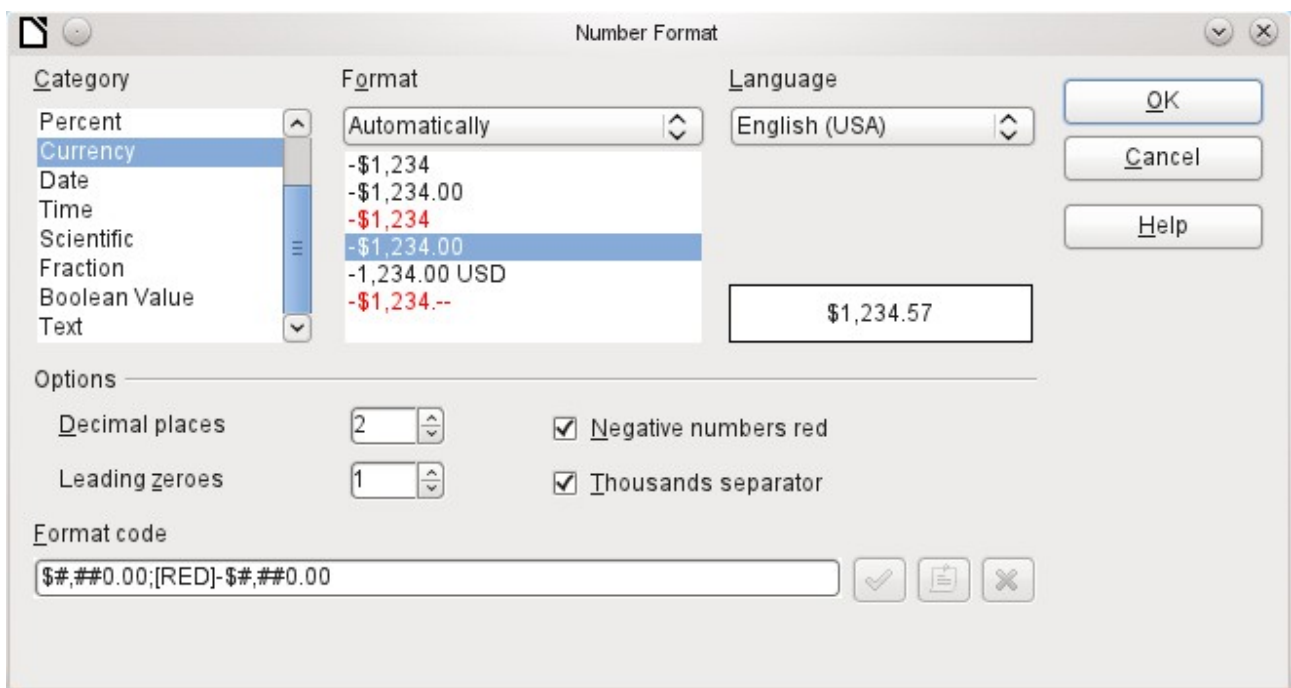


Figure 5: Formatting numbers

Special properties of graphical controls

The screenshot shows the 'General' tab of a property window. The 'Data' tab is also visible. The properties listed are:

Name.....	Image
Preserve as Link.....	Yes
Position X.....	1.57"
Visible.....	Yes
Position Y.....	0.20"
Width.....	1.57"
Height.....	1.57"
Print repeated values.....	Yes
Conditional Print Expression.....	[Empty] ...
Print When Group Change.....	No
Background Transparent.....	Yes
Background color.....	[Empty] ...
Vert. Alignment.....	Top
Graphics.....	[Empty] ...
Scale.....	Keep Ratio

A graphical control can contain graphics from both inside and outside the database. Unfortunately it is not possible at present to store a graphic such as a logo permanently in Base. Therefore it is essential that the graphic is available in the search path, even when you are presented with the choice of embedding rather than linking images and the first field *Set up as link* can be set (literally closed) to a corresponding planned functionality. This is one of several functions that are planned for Base and are in the GUI but have not actually been implemented yet—so the buttons and checkboxes have no effect.

Alternatively, of course, a graphic can be stored in the database itself and so becomes available internally. But in that case, it must be accessible through one of the fields in the query underlying the report.

To take up an *external graphic*, use the selection button beside the *Graphic* field to load it. To load a graphical database field, specify the field under the *Data* tab.

The vertical alignment setting does not seem to have any effect during the design stage. When you call up the report, however, the graphic appears in the correct position.

When scaling, you can select *No*, *Keep aspect ratio*, or *Autom. Size*. This corresponds to the settings for a form:

- *No*: The image is not fitted to the control. If it is too large, a cropped version is shown. The original image is not affected by this.
- *Keep aspect ratio*: The image is fitted to the control but not distorted.
- *Automatic size*: The image is fitted to the control and in some cases may be distorted.

Incorporating charts into the report

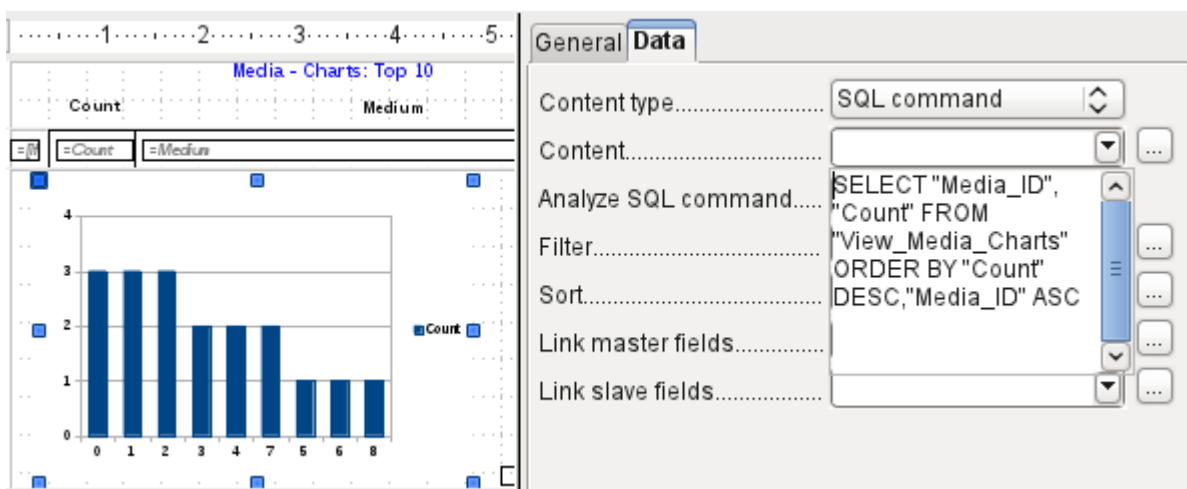
You can insert charts into a report by using the corresponding control or with **Insert > Report controls > Chart**. A chart is the only way to reproduce data that is not found in the data source specified for the report. A chart can therefore be seen as a kind of subreport, but also as a free-standing component of the report.



Chart type..... [] ...
Preview Row(s)..... 10

You must draw the place for the chart using the mouse. In the general properties, in addition to the familiar fields, you can choose a *Chart type* (see the corresponding types in Calc). In addition, you can set a maximum number of records for the preview, which will give an impression of how the chart will finally look.

Charts can be formatted in the same way as in Calc (double-click on the chart). For further information, see the description in the *LibreOffice Calc Guide*.



The screenshot shows a report titled "Media - Charts: Top 10". The report contains a bar chart with the following data:

Medium	Count
0	3
1	3
2	3
3	2
4	2
7	2
5	1
6	1
8	1

The chart is linked in the Data section of the report builder. The Data section shows the following properties:

- Content type: SQL command
- Content: []
- Analyze SQL command: SELECT "Media_ID", "Count" FROM "View_Media_Charts" ORDER BY "Count" DESC, "Media_ID" ASC
- Filter: []
- Sort: []
- Link master fields: []
- Link slave fields: []

The chart is linked in the Data section with the necessary data fields. Here, in a Media Top 10 list example, the chart shows the frequency with which particular media are borrowed. The Query Editor is used to create a suitable SQL command, as you would do for a listbox in a form. The first column in the query will be used to provide the labels for the vertical bars in the chart, while the second column yields the total number of loan transactions, shown in the height of the bars.

In the example above, the chart shows very little at first, since only limited test loans were carried out before the SQL command was issued.

The chart on the next page, from the membership database of a society (German: Mitgliederstatistik), has been prepared from a query that needs to be entered in direct SQL mode, as the graphical user interface does not understand it. For this reason *No* (German: Nein) has been chosen for *Analyze SQL command*, a choice that excludes any filtering and sorting with the internal tools of the Report Builder. These fields are therefore grayed out. In the data properties of the chart, **Query** (German: Abfrage) has been entered.

Fields are linked together in the same way as for a main form with a subform. In the report itself, the age distributions of male and female members are listed in tabular form. They are grouped by gender. Each group is given its own chart. So that the chart only contains data for the correct gender, the *Gender* field (German: Geschlecht) in the report is linked to the *Gender* in the chart.

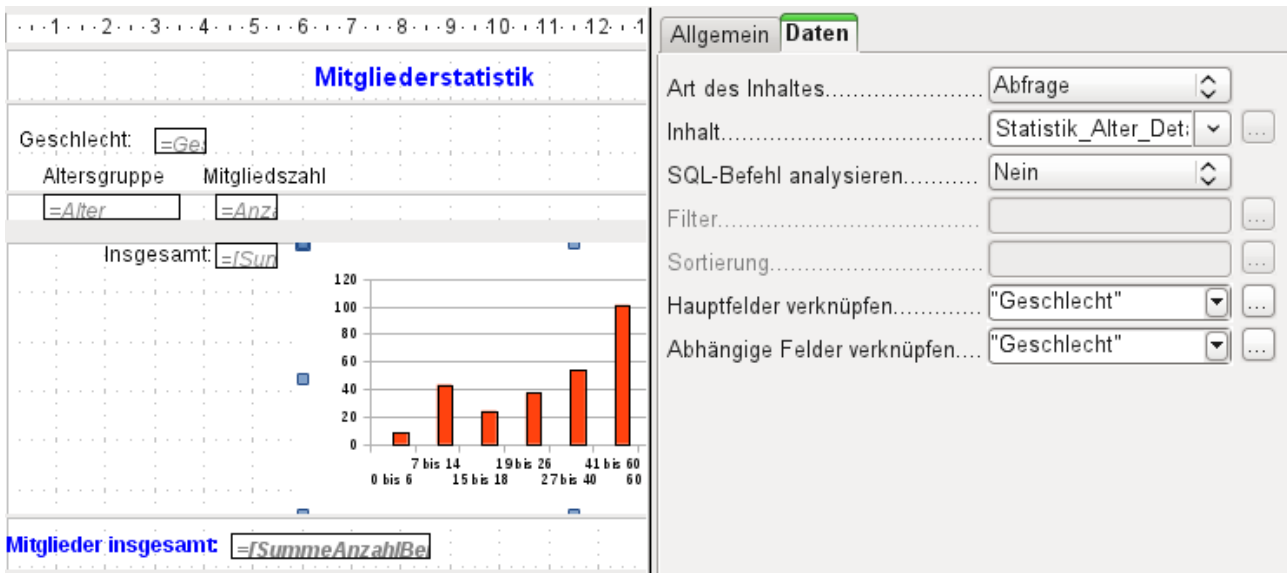


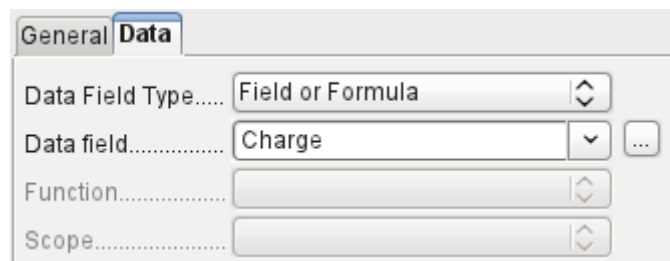
Figure 6: Linked fields for a chart, same as in forms. This example is not translated, because the report is not part of the example database.

Note

The creation of charts works at present only in Versions 3.3.x and 3.4.x. From Version 3.5, LibreOffice can no longer open these reports. They can still be produced, however.

LibreOffice is not the only program to have problems with charts. In Version 3.3 of OpenOffice.org, the display does not work either. OpenOffice.org does open the reports, but with the charts missing.

Data properties of fields



In the properties dialog, the Data tab shows by default only the database field from which the data for this report field will be read. However, in addition to the field types *Field* and *Formula*, the types *Function*, *Counter*, and *User-defined function* are available.

You can select in advance the *Sum*, *Minimum*, and *Maximum* functions. They will apply either to the current group or to the whole report. These functions can lead to problems if a field is empty (NULL). In such fields, if they have been formatted as numbers, "NaN" appears; that is, there is no numerical value present. For empty fields, no calculation is carried out and the result is always 0.

Such fields can be reformatted to display a value of 0 by using the following formula in the Data area of the view.

```
IF([numericfield];[numericfield];0)
```

This function calculates with the actual value of a field that has no value. It would seem simpler to formulate the underlying query for the report so that 0 is given instead of NULL for numeric fields.

The *Counter* counts only the records that will occur either in the group or in the report as a whole. If the counter is inserted into the Details area, each record will be provided with a running number. The numbering will apply only to records in the group or in the whole report.

Finally the detailed *User-defined Function* is available. It may happen that the Report Builder itself chooses this variant, if a calculation has been requested, but for some reason it cannot correctly interpret the data source.

Functions in the Report Builder

The Report Builder provides a variety of functions, both for displaying data and for setting conditions. If these are not sufficient, user-defined functions can be created using simple calculation steps, which are particularly useful in group footers and summaries.

Entering formulas

The Report Builder is based on the Pentaho Report Builder. A small part of its documentation is at <http://wiki.pentaho.com/display/Reporting/9.+Report+Designer+Formula+Expressions>.

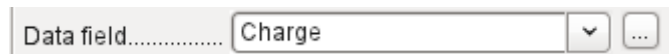
A further source is the Specifications for the OpenFormula Standard:

<http://www.oasis-open.org/committees/download.php/16826/openformula-spec-20060221.html>

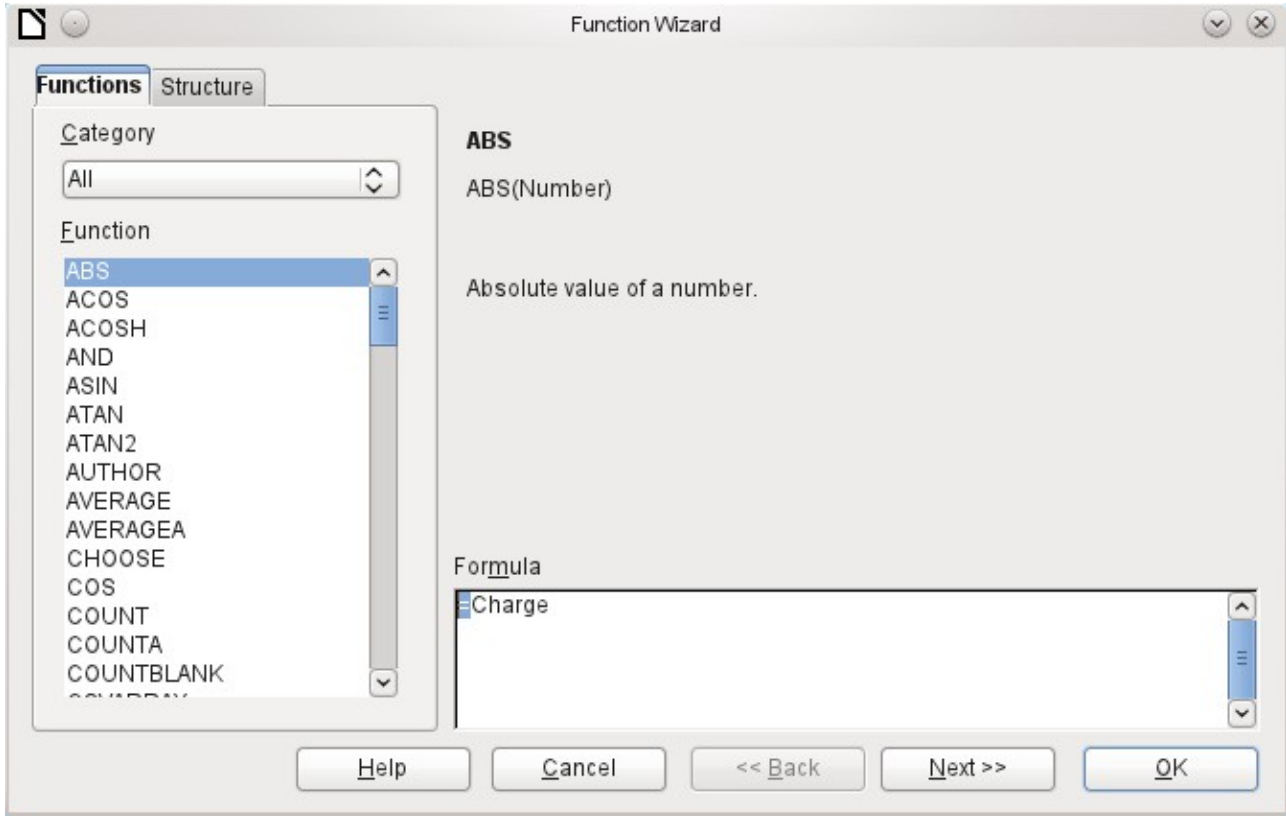
Basic principles:

Formulas start with an equals sign.	=
References to data fields are placed in square brackets.	[Field name]
If the data fields contain special characters (including spaces), the field name must also be enclosed in quotes.	["This fieldname should be in quotes"]
Text entry must always be in double quotes.	"Text entry"
The following operators are allowed.	+ , - , * (Multiplication), / (Division), % (divide the preceding number by 100), ^ (Raise to the power of the following number), & (concatenate text),
The following relationships are possible.	= , <> , < , <= , > , >=
Round brackets are allowed.	()
Default error message.	NA (Not available)
Error message for an empty field that was defined as a number.	NaN (Perhaps "not a number"?)

All formula input applies only to the current record. Relationships with previous or following records are therefore not possible.



Next to the data field is a button with three dots whenever a formula can be entered. This button starts the Function Wizard.



However there are far fewer functions than in Calc. Many functions do have Calc equivalents. There the Wizard calculates the result of the function directly.

The Function Wizard does not always work perfectly. For instance, text entries are not taken up with double quotes. However, only entries with double quotes are processed when starting the function.

The following functions are available:

Function	Description
Date and Time Functions	
DATE	Produces a valid date from numeric values for the year, the month and the day.
DATEDIF (DAY MONTH YEAR)	Returns the total years, months or days between two date values.
DATEVALUE	Converts an American date entry in text form (quoted) into a date value. The American variant that is produced can then be reformatted.
DAY	Returns the day of the month for a given date. DAY([Date field])

DAYS	Returns the number of days between two dates.
hour	Returns the hours of a given time in 24-hour format. HOUR([DateTimeField]) calculates the hours in the field.
MINUTE	Returns the minutes of a date in the internal numeric format MINUTE([Timefield]) calculates the minutes part of the time.
MONTH	Returns the month for an entered date as a number. MONTH([Datefield])
NOW	Returns the current date and time.
SECOND	Returns the seconds of a date in the internal numeric format SECOND(NOW()) shows the seconds part of the time the command is executed.
TIME	Shows the current time.
TIMEVALUE	Converts a text entry for a time into a time value for calculations.
TODAY	Shows the current date.
WEEKDAY	Returns the day of the week as a number. Day number 1 is Sunday.
YEAR	Returns the year part of a date entry.
Logical functions	
AND	Yields TRUE when all its arguments are TRUE.
FALSE	Defines the logical value as FALSE.
IF	If a condition is TRUE, then this value, else another value.
IFNA	(since LO 3.5)
NOT	Reverses the logical value of an argument.
OR	Yields TRUE when one of its conditions is TRUE.
TRUE	Defines the logical value as TRUE.
XOR	Yields TRUE when only one of the linked values is TRUE.
Rounding functions	
INT	Rounds down to the previous integer.
Mathematical functions	
ABS	Returns the absolute (non-negative) value of a number.
ACOS	Calculates the arccosine of a number. - arguments between -1 and 1. (since LO 3.5)
ACOSH	Calculates the areacosine (inverse hyperbolic cosine) – argument ≥ 1 . (since LO 3.5)
ASIN	Calculates the arcsine of a number – argument between -1 and 1. (since LO 3.5)

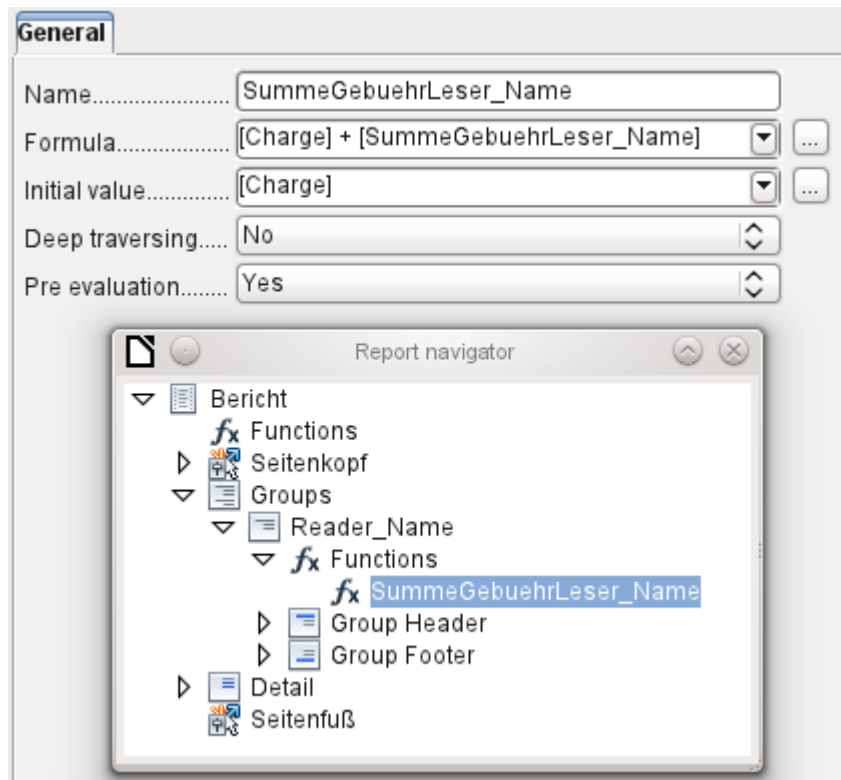
ATAN	Calculates the arctangent of a number. (since LO 3.5)
ATAN2	Calculates the arctangent of an x-coordinate and a y-coordinate. (since LO 3.5)
AVERAGE	Gives the average of the entered values. (occurs twice in the LO 3.3.4 Formula Wizard)
AVERAGEA	Gives the average of the entered values. Text is treated as zero. (since LO 3.5)
COS	Argument is the angle in radians whose cosine is to be calculated. (since LO 3.5)
EVEN	Rounds a positive number up or a negative number down to the next even integer.
EXP	Calculates the exponential function (Base 'e'). (since LO 3.5)
LN	Calculates the natural logarithm of a number. (since LO 3.5)
LOG10	Calculates the logarithm of a number (Base '10'). (since LO 3.5)
MAX	Returns the maximum of a series of numbers.
MAXA	Returns the maximum value in a row. Any text is set to zero.
MIN	Returns the smallest of a series of values.
MINA	Returns the minimum value in a row. Any text is set to zero.
MOD	Returns the remainder for a division when you enter the dividend and divisor.
ODD	Rounds a positive number up or a negative number down to the next odd integer.
PI	Gives the value of the number ' π '. (since LO 3.5)
POWER	Raises the base to the power of the exponent. (Since LO 3.5)
SIN	Calculates the sine of a number. (since LO 3.5)
SQRT	Calculates the square root of a number. (since LO 3.5)
SUM	Sums a list of numeric values
SUMA	Sums a list of numeric values. Text and Yes/No fields are allowed. Unfortunately this function (still) ends with an error message. (since LO 3.5)
VAR	Calculates the variance, starting from a sample. (since LO 3.5)

Text functions	
EXACT	Shows if two text strings are exactly equal.
FIND	Gives the offset of a text string within another string.
LEFT	The specified number of characters of a text string are reproduced starting from the left.
LEN	Gives the number of characters in a text string.
LOWER	Converts text to lower case.
MESSAGE	Formats the value into the given output format. (since LO 3.5)
MID	The specified number of characters of a text string are reproduced starting from a specified character position.
REPLACE	Replaces a substring by a different substring. The starting position and the length of the substring to be replaced must be given.
REPT	Repeats text a specified number of times.
RIGHT	The specified number of characters of a text string are reproduced starting from the right.
SUBSTITUTE	Replaces specific parts of a given text string by new text. Additionally you can specify which of several occurrences of the target string are to be replaced.
T	Returns the text, or an empty text string if the value is not text (for example a number).
TEXT	Conversion of numbers or times into text.
TRIM	Removes leading spaces and terminal spaces, and reduces multiple spaces to a single space.
UNICHAR	Converts a Unicode decimal number into a Unicode character. For example, 196 becomes 'Ä' ('Ä' has the hexadecimal value 00C4, which is 196 in decimals without leading zeros).
UNICODE	Converts a Unicode character into a Unicode decimal number. 'Ä' becomes 196.
UPPER	Returns a text string in upper case.
URLENCODE	Converts a given text into one that conforms to a valid URL. If no particular standard is specified, ISO-8859-1 is followed..
Information functions	
CHOOSE	The first argument is an index, followed by a list of values. The value represented by the index is returned. CHOOSE(2;"Apple";"Pear";"Banana") returns Pear . CHOOSE([age_level_field];"Milk";"Cola";"Beer") returns a possible drink for the given 'age_level_field' .
COUNT	Only fields containing a number or a date are counted. COUNT([time];[number]) returns 2 , if both fields contain a value (non-NULL) or else 1 or 0.
COUNTA	Includes also fields containing text. Even NULL is counted, along with boolean fields.

COUNTBLANK	Counts the empty fields in a region.
HASCHANGED	Checks if the named column has changed. However no information about the column is provided.
INDEX	Works with regions (since LO 3.5).
ISBLANK	Tests if the field is NULL (empty).
ISERR	Returns TRUE if the entry has an error other than NA. ISERR(1/0) gives TRUE
ISERROR	Like ISERR, except that NA also returns TRUE.
ISEVEN	Tests if a number is even.
ISLOGICAL (<i>ISTLOG</i>)	Tests if this is a Yes/No value. ISLOGICAL(TRUE()) or ISLOGICAL(FALSE()) yield TRUE , Text values such as ISLOGICAL("TRUE") yield FALSE .
ISNA	Tests if the expression is an error of type NA.
ISNONTEXT	Tests if the value is not text.
ISNUMBER	Tests if something is numeric. ISNUMBER(1) yields TRUE , ISNUMBER("1") yields FALSE
ISODD	Tests if a number is an odd number.
ISREF	Tests if something is a field reference. ISREF([Fieldname]) yields TRUE , ISREF(1) yields FALSE .
ISTEXT	Tests if the content of the field is text.
NA (<i>NV</i>)	Returns the error code NA .
VALUE	(since LO 3.5)
User defined	
CSVARRAY	Converts CSV text into an array. (since LO 3.5)
CSVTEXT	Converts an array into CSV text. (since LO 3.5)
NORMALIZEARRAY	(since LO 3.5)
NULL	Returns NULL.
PARSEDATE	Converts text into a date. Uses the SimpleDateFormat. Requires a date in text as described in this date format. Example: PARSEDATE("9.10.2012";"dd.MM.yyyy") yields the internally usable number for the date. (since LO 3.5)
Document information	
AUTHOR	Author, as read from the Tools → Options → LibreOffice → User data . This is not therefore the actual author but the current user of the database.
TITLE	Returns the title of the report.

User-defined functions

You can use user-defined functions to return specific intermediate results for a group of records. In the above example, a function of this sort was used to calculate the fines in the *Reader_Name_Footer* area.



In the Report Navigator the function is displayed under *Reader_Name* group. By right-clicking on this function, you can define additional functions by name.

The properties of the function *SummeGebuehrLeser_Name* are shown above. The formula adds the field *Charge* to the value already stored in the function itself. The initial value is the value of the *Charge* field on the first traverse of the group. This value is stored in the function under the function name and is reused in the formula, until the loop is ended and the group footer is written.

Deep traversing seems to have no function for now, unless charts are being treated here as subreports.

If *Pre evaluation* is activated for the function, the result can also be placed in the group header. Otherwise the group header contains only the corresponding value of the first field of the group.

User-defined functions can also reference other user-defined functions. In that case you must ensure that the functions used have already been created. Pre-calculation in functions that refer to other functions must be excluded.

$$[\text{SumMarksClass}] / ([\text{ClassNumber}] + 1)$$

refers to the *Class* group. The content of the *Marks* field is summed and the sum for all the records is returned. The sum of the marks is divided by the sum of the records. To get the correct number, 1 must be added as shown with *[ClassNumber]*. This will then yield the average marks.

Formula entry for a field

Using **Data > Data field** you can enter formulas that affect only one field in the Details area.

```
IF([boolean_field];"Yes";"No")
```

sets the allowable values to "Yes" or "No" instead of TRUE and FALSE.

Conditional print



The general properties of group headers, group footers, and fields include a *Conditional Print Expression* field. Formulas that are written in this field influence the content of a field or the display of an entire region. Here, too, you can make use of the Function Wizard.

```
[Fieldname]="true"
```

causes the content of the named field to be displayed only if it is true.

Many forms of conditional display are not fully determined by the specified properties. For instance, if a graphical separator line is to be inserted after the tenth place of a list of competition results, you cannot simply use the following conditional display command for the graphic:

```
[Place]=10
```

This command does not work. Instead the graphic will continue to appear in the *Details* section after each subsequent record.

It is safer to bind the conditional display to a *group footer* rather than to the graphic, if this is not otherwise needed. The line is positioned in the *group footer*. Then the line does actually appear after the 10th place, when formulated as above. But in that case the content that would previously have appeared in the *Details* section must be stored in the *group header*.

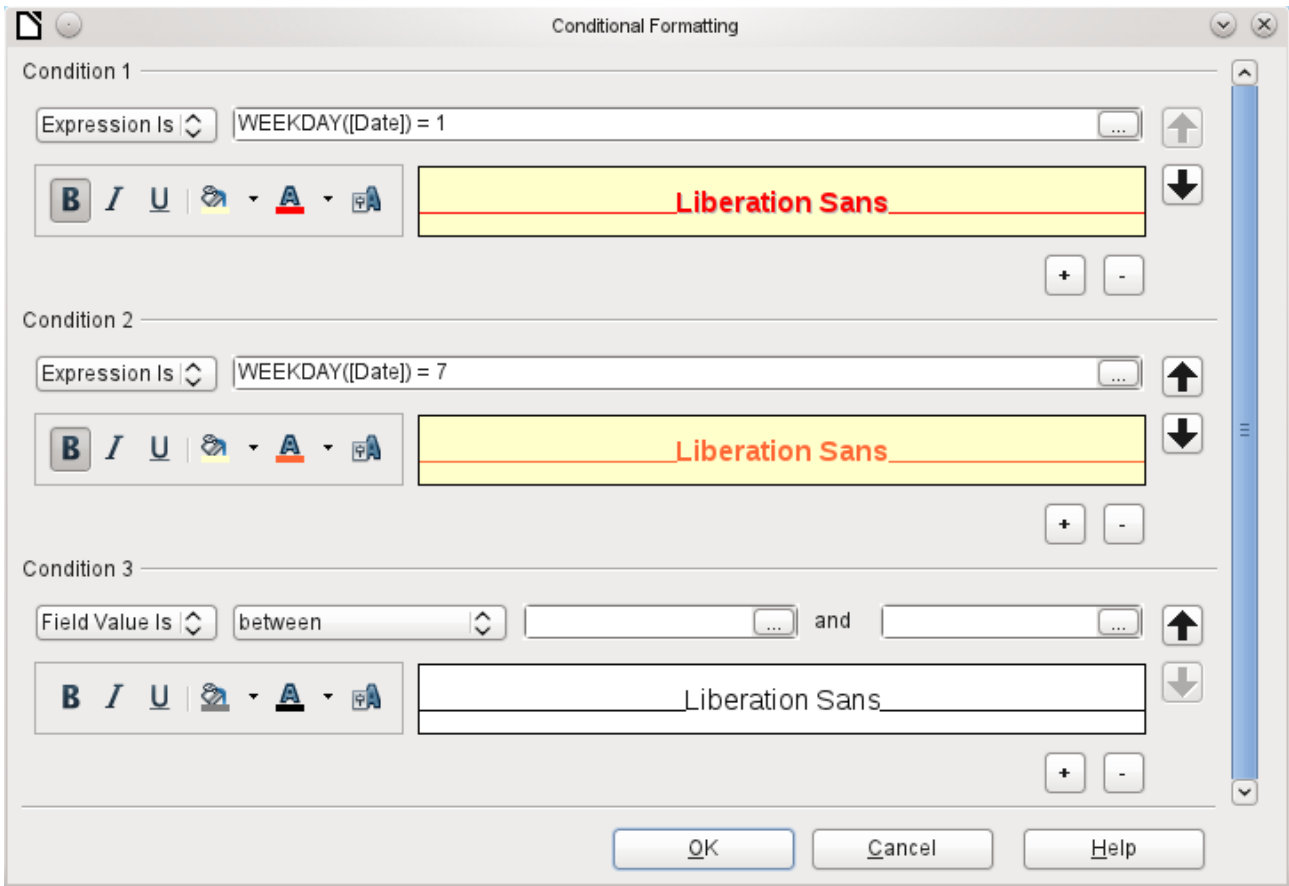
Conditional formatting

Conditional formatting can be used, for example, to format a calendar so that weekends are shown differently. Choose **Format > Conditional formatting** and enter:

```
WEEKDAY([Date])=1
```

and the corresponding formatting for Sundays.

If you use 'Expression is' in conditional formatting, you can enter a formula. As is usual in Calc, several conditions can be formulated and are evaluated sequentially. In the above example, first Sunday is tested for, then Saturday. Finally there might be a query about the content of the field. So for example the content 'Holiday' would lead to a different format.



Note

The Report Builder is a plug-in. If additional uncorrectable errors occur (formulas not implemented, too long text shown as an empty field, and so on), it is sometimes necessary to delete parts of the report or simply create it afresh.