



LibreOffice 4.0

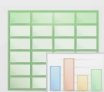
The Document Foundation

Math Guide

Using the Equation Editor



Writer



Calc



Impress



Draw



Base



Math

Copyright

This document is Copyright © 2011–2013 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

Jean Hollis Weber

Hazel Russman

Laurent Balland-Poirier

Feedback

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

Acknowledgments

This guide is adapted and updated from the *OpenOffice.org 3.3 Math Guide*. The contributors to that book are:

Daniel Carrera

Agnes Belzunce

TJ Frazier

Peter Kupfer

Ian Laurenson

Janet M. Swisher

Jean Hollis Weber

Michele Zarri

Additional material has been added from the German *Math Handbuch* for LibreOffice 3.4 (translated to English by Hazel Russman). Contributors to that book are:

Regina Henschel

Christian Kühn

Florian Reisinger

Gisbert Friege (Dmaths)

Jochen Schiffers

Additional material has been added from the French *How-To Math* for LibreOffice (translated to English by Laurent Balland-Poirier). Contributors to that book are:

Bernard Siaud

Frédéric Parrenin

Publication date and software version

Published 3 July 2013. Based on LibreOffice 4.0.4.

Contents

Copyright.....	2
Contributors.....	2
Feedback.....	2
Acknowledgments.....	2
Publication date and software version.....	2
Preface.....	6
Who is this book for?.....	6
Where to get more help.....	6
What you see may be different.....	7
What are all these things called?.....	8
Using LibreOffice on a Mac.....	9
Who wrote this book?.....	9
Frequently asked questions.....	9
What is Math?.....	11
Getting started.....	11
Creating an equation as a separate document.....	11
Inserting a formula into a Writer document.....	12
Entering a formula.....	13
The Elements window.....	13
Right-click (context) menu.....	15
Markup.....	15
Greek characters.....	16
Changing a formula.....	18
Choosing the region to alter.....	18
Carrying out changes.....	18
Formula layout.....	19
Brackets are your friends.....	19
Brackets with matrices look ugly!.....	19
Isolated and unpaired brackets.....	20
Recognizing functions in Math.....	20
Equations over more than one line.....	21
How do I add limits to my sum/integral?.....	21
How do I write a derivative?.....	22
Markup characters as regular characters.....	22
Text in a formula.....	23
How do I align my equations at the equals sign?.....	23
Changing the appearance of formulas.....	24
Changing the font size.....	24
Changing the font.....	25
Adjusting spacing in formulas.....	26
Changing the alignment.....	28

Changing the color.....	29
Formulas in Writer documents.....	29
Numbering equations.....	29
Position.....	31
Margins.....	31
Text mode.....	31
Background, borders, and size.....	31
Creating a formula library.....	32
Fast insertion of formulas.....	32
Formulas in Calc, Impress, and Draw.....	32
Graphical properties.....	32
Combining formulas with text.....	33
Formulas in charts.....	34
Customizations.....	34
Formula editor as a floating window.....	34
Add button to toolbar.....	35
Add keyboard shortcut.....	35
Customizing the catalog.....	36
Customizing operators.....	38
Space at the end of a formula.....	38
Default layout with style.....	39
Application to chemical formulas.....	40
Export and import.....	40
Export as MathML.....	40
Microsoft file formats.....	42
XHTML.....	43
Flat XML.....	43
Extensions.....	43
Dmaths – an extension for more than just faster formula input.....	43
Symbolic computation with CmathOOo and CmathOOoCAS.....	45
Technical details.....	45
OASIS Open Document Format for Office applications.....	45
Handling of formulas in Basic.....	45
Math commands - Reference.....	48
Unary / binary operators.....	48
Relations.....	49
Set operations.....	50
Functions.....	51
Operators.....	52
Attributes.....	53
Brackets.....	54
Formats.....	56
Others.....	57

Characters – Greek.....	58
Characters – Special.....	58
Reserved words in alphabetic order.....	59
Index.....	62

Preface

Who is this book for?

LibreOffice Math is an equation (formula) editor. Anyone who wants to get up to speed quickly with Math will find this book valuable.

If you have never used LibreOffice Math before, or you want an introduction to all of its components, you might like to read *Getting Started with LibreOffice 4.0* first.

Where to get more help

This book, the other LibreOffice user guides, the built-in Help system, and user support systems assume that you are familiar with your computer and basic functions such as starting a program, opening and saving files.

Help system

LibreOffice comes with an extensive Help system. This is your first line of support for using LibreOffice.

To display the full Help system, press *F1* or go to **Help > LibreOffice Help** on the main menu bar. In addition, you can choose whether to activate *Tips*, *Extended tips*, and the *Help Agent* by going to **Tools > Options > LibreOffice > General** on the main menu bar.

If *Tips* are enabled, hover the mouse pointer over any of the icons to see a small box (tooltip) with a brief explanation of the icon's function. For a more detailed explanation, select **Help > What's This?** on the main menu bar and hover the pointer over the icon.

Free online support

The LibreOffice community not only develops software, but provides free, volunteer-based support. See Table 1 and this web page: <http://www.libreoffice.org/get-help/>

Table 1: Free support for LibreOffice users

Free LibreOffice support	
Ask LibreOffice	Questions and answers from the LibreOffice community http://ask.libreoffice.org/en/questions/
Documentation	User guides, how-tos, and other documentation. http://www.libreoffice.org/get-help/documentation/ https://wiki.documentfoundation.org/Documentation/Publications
Mailing lists	Free community support is provided by a network of experienced users http://www.libreoffice.org/get-help/mailling-lists/
FAQs	Answers to frequently asked questions http://wiki.documentfoundation.org/Faq
International support	The LibreOffice website in your language. http://www.libreoffice.org/international-sites/ International mailing lists http://wiki.documentfoundation.org/Local_Mailing_Lists
Accessibility options	Information about available accessibility options. http://www.libreoffice.org/get-help/accessibility/

You can get comprehensive online support from the community through mailing lists and the Ask LibreOffice website, <http://ask.libreoffice.org/en/questions/>. Other websites run by users also offer free tips and tutorials. This forum provides community support for LibreOffice:
<http://en.libreofficeforum.org/>

This site provides support for LibreOffice, among other programs:
<http://forum.openoffice.org/en/forum/>

Paid support and training

Alternatively, you can pay for support services. Service contracts can be purchased from a vendor or consulting firm specializing in LibreOffice.

What you see may be different

LibreOffice runs on Windows, Linux, and Mac OS X operating systems, each of which has several versions and can be customized by users (fonts, colors, themes, window managers).

Illustrations

The illustrations in this guide were taken from a variety of computers and operating systems. Therefore, some illustrations will not look exactly like what you see on your computer display.

Also, some of the dialogs may be different because of the settings selected in LibreOffice. You can either use dialogs from your computer system or dialogs provided by LibreOffice (default). To change to using LibreOffice dialogs if settings have been altered:

- 1) On Linux and Windows operating systems, go to **Tools > Options > LibreOffice > General** on the main menu bar to open the dialog for general options.
- 2) On a Mac operating system, go to **LibreOffice > Preferences > General** on the main menu bar to open the dialog for general options.
- 3) Select *Use LibreOffice dialogs* in *Open/Save dialogs* and/or, in Linux or Mac OSX operating systems only, *Print dialogs* to display the LibreOffice dialogs on your computer display.
- 4) Click **OK** to save your settings and close the dialog.

Icons

The icons used to illustrate some of the many tools available in LibreOffice may differ from the ones used in this guide. The icons in this guide have been taken from a LibreOffice installation that has been set to display the Galaxy set of icons.

If you wish, you can change your LibreOffice software package to display Galaxy icons as follows:

- 1) On Linux and Windows operating systems, go to **Tools > Options > LibreOffice > View** on the main menu bar to open the dialog for view options.
- 2) On a Mac operating system, go to **LibreOffice > Preferences > View** on the main menu bar to open the dialog for view options.
- 3) In *User interface > Icon size and style* select *Galaxy (default)* from the options available in the drop-down list.
- 4) Click **OK** to save your settings and close the dialog.

Note

Some Linux operating systems, for example Ubuntu, include LibreOffice as part of the installation and may not include the Galaxy icon set. You should be able to download the Galaxy icon set from the software repository for your Linux operating system.

What are all these things called?

The terms used in LibreOffice for most parts of the *user interface* (the parts of the program you see and use, in contrast to the behind-the-scenes code that actually makes it work) are the same as for most other programs.

A *dialog* is a special type of window. Its purpose is to inform you of something, or request input from you, or both. It provides controls for you to use to specify how to carry out an action. The technical names for common controls are shown in Figure 1. In most cases we do not use the technical terms in this book, but it is useful to know them because the Help and other sources of information often use them.

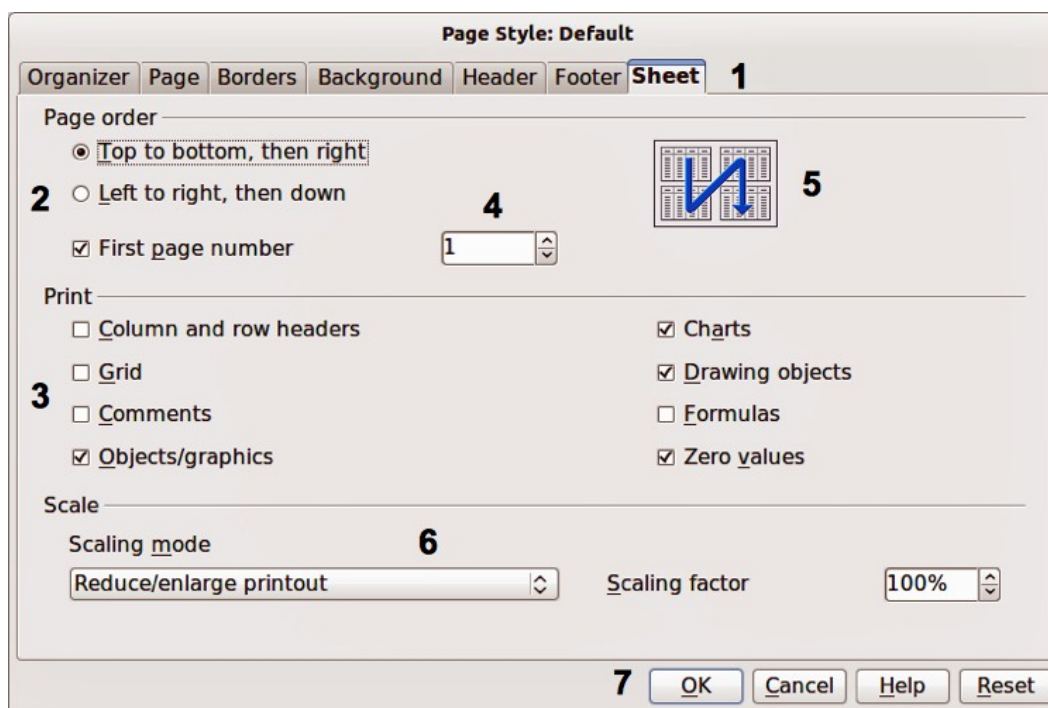


Figure 1: Dialog (from Calc) showing common controls

- 1) Tabbed page (not strictly speaking a control).
- 2) Radio buttons (only one can be selected at a time).
- 3) Checkbox (more than one can be selected at a time).
- 4) Spin box (click the up and down arrows to change the number shown in the text box next to it, or type in the text box).
- 5) Thumbnail or preview.
- 6) Drop-down list from which to select an item.
- 7) Push buttons.

In most cases, you can interact only with the dialog (not the document itself) as long as the dialog remains open. When you close the dialog after use (usually, clicking **OK** or another button saves your changes and closes the dialog), then you can again work with your document.

Some dialogs can be left open as you work, so you can switch back and forth between the dialog and your document. An example of this type is the Find & Replace dialog.

Using LibreOffice on a Mac

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

Who wrote this book?

This book was written by volunteers from the LibreOffice community, as listed on the Copyright page. Profits from sales of the printed edition will be used to benefit the community.

Frequently asked questions

How is LibreOffice licensed?

LibreOffice is distributed under the Open Source Initiative (OSI) approved Lesser General Public License (LGPL). The LGPL license is available from the LibreOffice website: <http://www.libreoffice.org/download/license/>

May I distribute LibreOffice to anyone?

Yes.

How many computers may I install it on?

As many as you like.

May I sell it?

Yes.

May I use LibreOffice in my business?

Yes.

Is LibreOffice available in my language?

LibreOffice has been translated (localized) into over 40 languages, so your language probably is supported. Additionally, there are over 70 *spelling*, *hyphenation*, and *thesaurus* dictionaries available for languages, and dialects that do not have a localized program interface. The dictionaries are available from the LibreOffice website at: www.libreoffice.org.

How can you make it for free?

LibreOffice is developed and maintained by volunteers and has the backing of several organizations.

How can I contribute to LibreOffice?

You can help with the development and user support of LibreOffice in many ways, and you do not need to be a programmer. To start, check out this webpage: <http://www.documentfoundation.org/contribution/>

May I distribute the PDF of this book, or print and sell copies?

Yes, as long as you meet the requirements of one of the licenses in the copyright statement at the beginning of this book. You do not have to request special permission. In addition, we request that you share with the project some of the profits you make from sales of books, in consideration of all the work we have put into producing them.

What is Math?

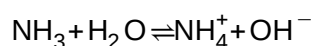
Math is LibreOffice's component for writing mathematical and chemical equations. It is most commonly used as an equation editor for text documents, but it can also be used with other types of documents or stand-alone. When used inside Writer, the equation is treated as an object inside the text document.

Note

The equation editor is for writing equations in symbolic form, as in equation 1. If you want to evaluate a numeric value, see the *Calc Guide*.

$$\frac{df(x)}{dx} = \ln(x) + \tan^{-1}(x^2) \quad (1)$$

or



Getting started

You can create an equation (formula) as a separate document or insert it into a document in Writer or another component of LibreOffice.

Creating an equation as a separate document

To create an equation as a separate document, open the Math component of LibreOffice using one of these methods:

- On the menu bar, choose **File > New > Formula**.
- On the standard toolbar, click the triangle to the left of the **New** icon and choose **Formula**.
- From the Start Center, click **Formula**.

An empty formula document opens (see Figure 2).

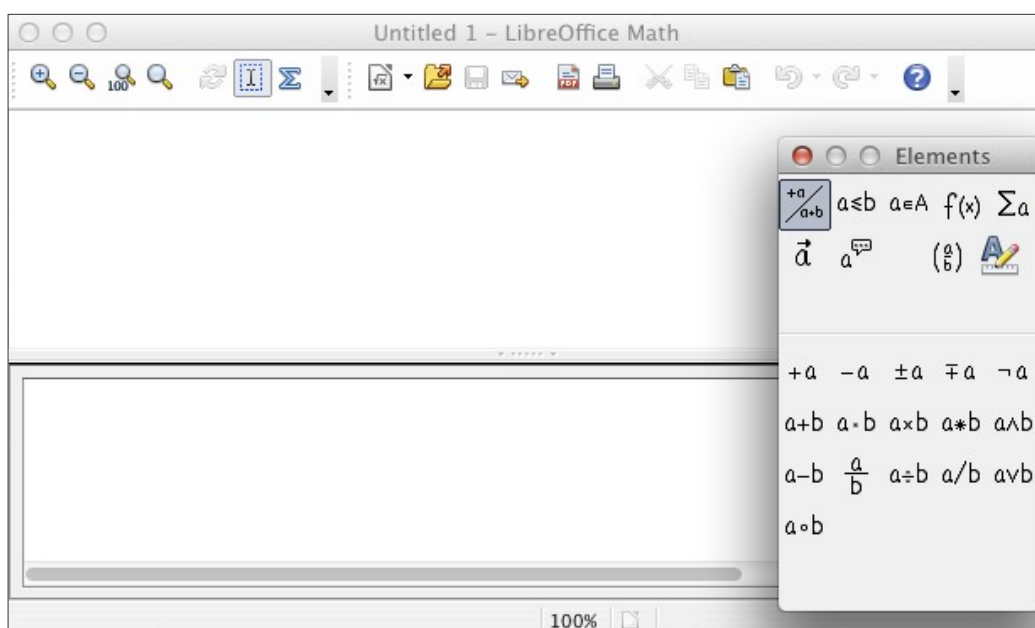


Figure 2: An empty formula document

The upper area is the preview window, where the equation will appear during and after input. The lower area is the equation editor, where the markup code for the equation is entered. The floating Elements window will also appear.

Inserting a formula into a Writer document

To insert a formula into a Writer document, open the document and then choose **Insert > Object > Formula** from the menu bar.

The formula editor opens at the bottom of the Writer window, and the floating Elements window appears. You will also see a small box with a gray border in your document, where the formula will be displayed, as shown in Figure 3.

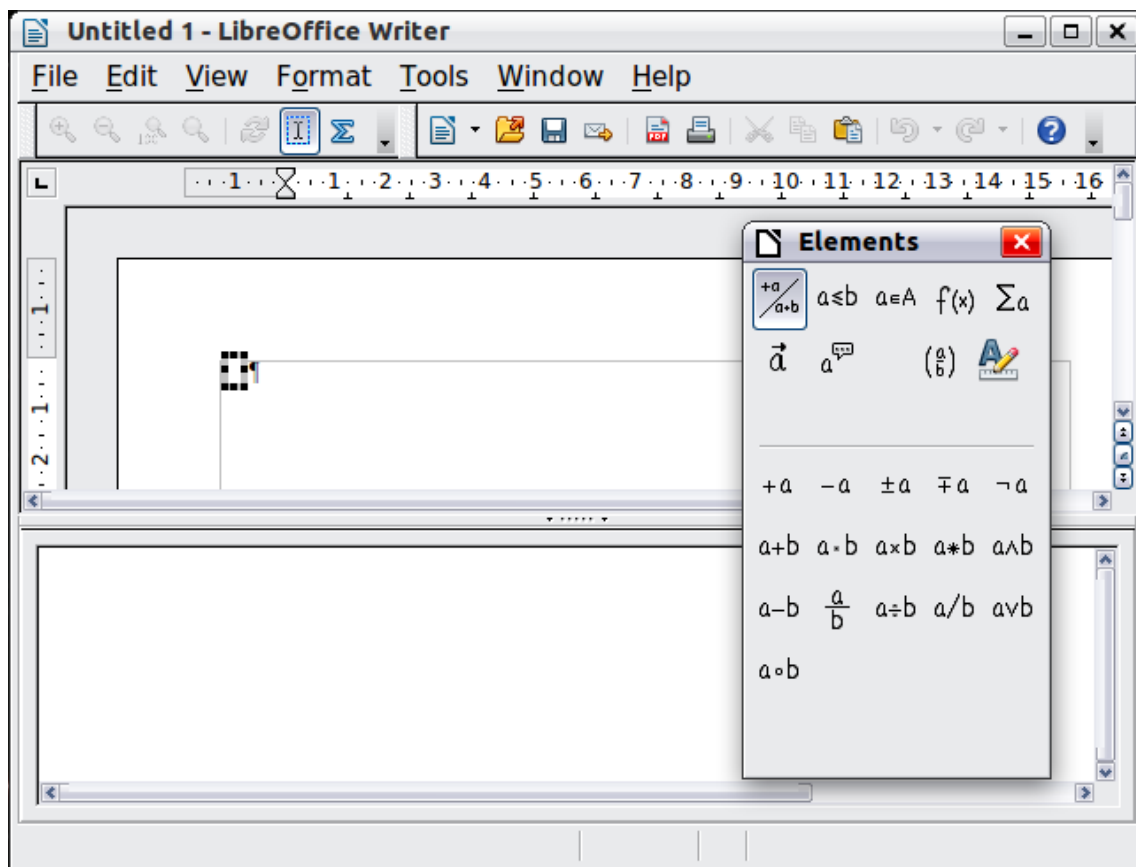


Figure 3: Equation Editor, Elements window, and location of resulting equation in Writer

Tip

If the Elements window is missing, show it with **View > Elements**.

When you are done entering the formula, you can close the editor by pressing the *Esc* key or by clicking an area outside the formula in the main document. A double-click will open the editor again, so you can edit the formula.

Formulas are inserted as OLE objects. In a Writer document, the formula is anchored as a character, so it is embedded in the continuous text. You can, as with any other OLE object, change the anchor and make the formula floating. In Calc, Impress, and Draw documents, formulas are embedded as floating OLE objects.

If you frequently need to insert formulas, it is advisable to add the Formula button to the Standard toolbar or create a keyboard shortcut. See “Add button to toolbar” or “Add keyboard shortcut” on page 35.

Entering a formula

The equation editor uses a markup language to represent formulas. For example, `%beta` creates the Greek character beta (β). This markup is designed to read similar to English whenever possible. For example, `a over b` produces a fraction: $\frac{a}{b}$.

You can enter a formula in three ways:

- Select a symbol from the Elements window.
- Right-click on the equation editor and select the symbol from the context menu.
- Type markup in the equation editor.

The context menu and the Elements window insert the markup corresponding to a symbol. This provides a convenient way to learn the LibreOffice Math markup.

Note

Click on the document body to exit the formula editor.
Double-click on a formula to enter the formula editor again.

The Elements window

The simplest method for entering a formula is the Elements window.

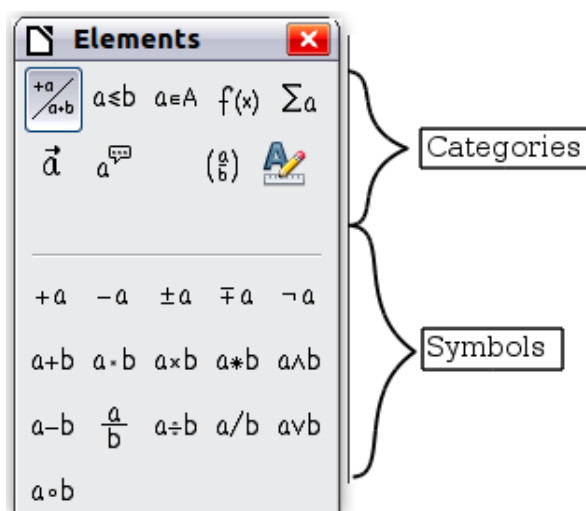


Figure 4: Symbols are divided into categories

The Elements window is divided into two main parts.

- The **top** shows the symbol categories. Click on these to change the list of symbols.
- The **bottom** shows the symbols available in the current category.

Tip

You can hide or show the Elements window with **View > Elements**.

Example 1: 5×4

For this example we will enter a simple formula: 5×4 . On the Elements window:

- 1) Select the top-left button of the categories (top) section.
- 2) Click on the multiplication symbol.

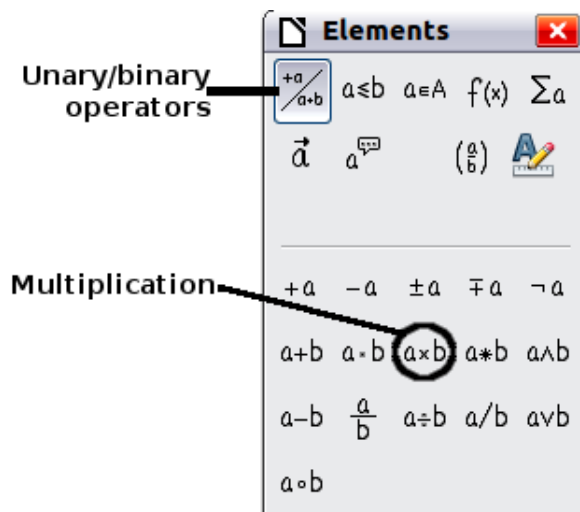


Figure 5: Selecting the multiplication symbol

When you select the multiplication symbol on the Elements window, two things happen:

- The equation editor shows the markup: `<?> times <?>`
- The body of the document shows a gray box like this: $\square \times \square$



Figure 6: Result of selecting the multiplication symbol

The `<?>` symbols shown in Figure 6 are placeholders that you can replace by other text, for example **5** and **4**. The equation will update automatically, and the result should resemble Figure 7.

Tip

During edition of a formula, reserved placeholders are indicated by squares in the formula and `<?>` in the command window. You can navigate through these placeholders thanks to **F4** and **Shift+F4**.

Tip

To keep the equation from updating automatically, select **View > AutoUpdate display**. To update a formula manually, press **F9** or select **View > Update**.

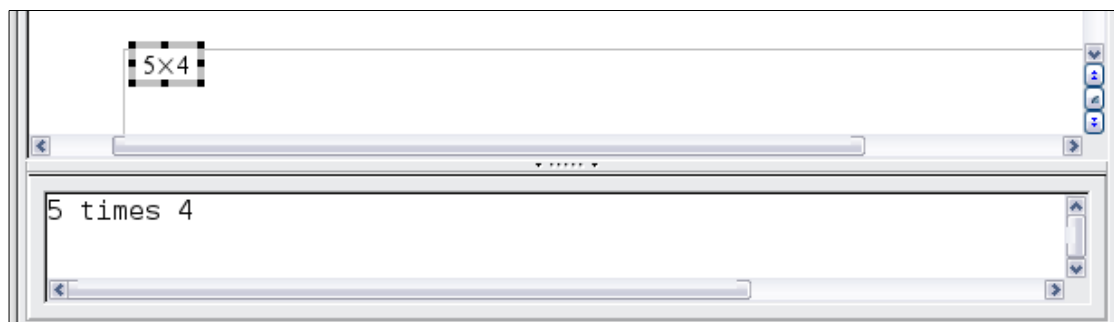


Figure 7: Result of entering 5 and 4 next to the times operator

Right-click (context) menu

Another way to access mathematical symbols is to right-click on the equation editor. This pops up the menu shown in Figure 8. The items in this menu correspond to those in the Elements window, with some extra commands.

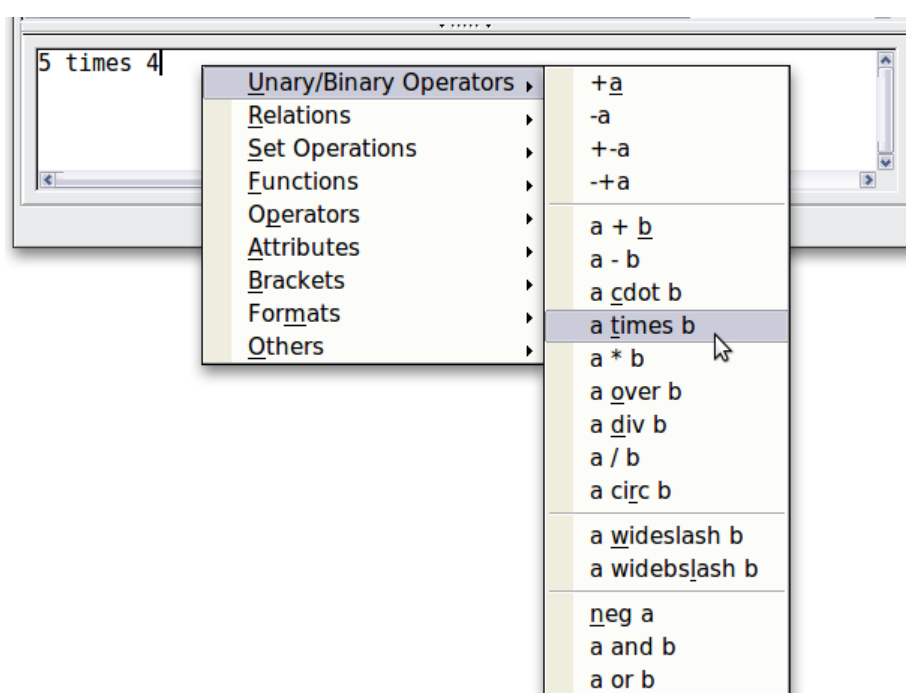


Figure 8: Right-click (context) menu

Note

Neither the window elements, nor the context menu contains a complete list of commands. For some seldom-used commands, you must always enter the markup. A complete list of commands can be found in Appendix A to this manual, page 48.

Markup

You can type the markup directly in the equation editor. For example, you can type **5 times 4** to obtain 5×4 . If you know the markup, this can be the fastest way to enter a formula.

Tip

The formula markup resembles the way the formula reads in English.

Below is a short list of common equations and their corresponding markup.

<i>Display</i>	<i>Command</i>	<i>Display</i>	<i>Command</i>
$a=b$	a = b	\sqrt{a}	sqrt {a}
a^2	a^2	a_n	a_n
$\int f(x)dx$	int f(x) dx	$\sum a_n$	sum a_n
$a \leq b$	a <= b	∞	infinity
$a \times b$	a times b	$x \cdot y$	x cdot y

Greek characters

Greek characters ($\alpha, \beta, \gamma, \theta$, etc) are common in mathematical formulas. *These characters are not available in the Elements window or the right-click menu.* Fortunately, the markup for Greek characters is simple: Type a % sign followed by the name of the character, in English.

- To write a *lowercase* character, type the name of the character in lowercase.
- To write an *uppercase* character, type the name of the character in uppercase.
- To write in *italic*, add an i between % sign and the name of the character.

A complete table of Greek characters is provided on page 58. See the table below for some examples.

<i>Lowercase</i>	<i>Uppercase</i>	<i>Italic lowercase</i>	<i>Italic uppercase</i>
%alpha → α	%ALPHA → A	%ialpha → α	%iALPHA → A
%beta → β	%BETA → B	%ibeta → β	%iBETA → B
%gamma → γ	%GAMMA → Γ	%igamma → γ	%iGAMMA → Γ
%psi → ψ	%PSI → Ψ	%ipsi → ψ	%iPSI → Ψ
%phi → φ	%PHI → Φ	%iphi → φ	%iPHI → Φ
%theta → θ	%THETA → Θ	%itheta → θ	%iTHETA → Θ

Another way to enter Greek characters is by using the Symbols catalog window. Choose **Tools > Catalog**. This window is shown in Figure 9. Under *Symbol set*, select **Greek** and double-click on a Greek letter from the list. The markup name of the character is shown below the list window.

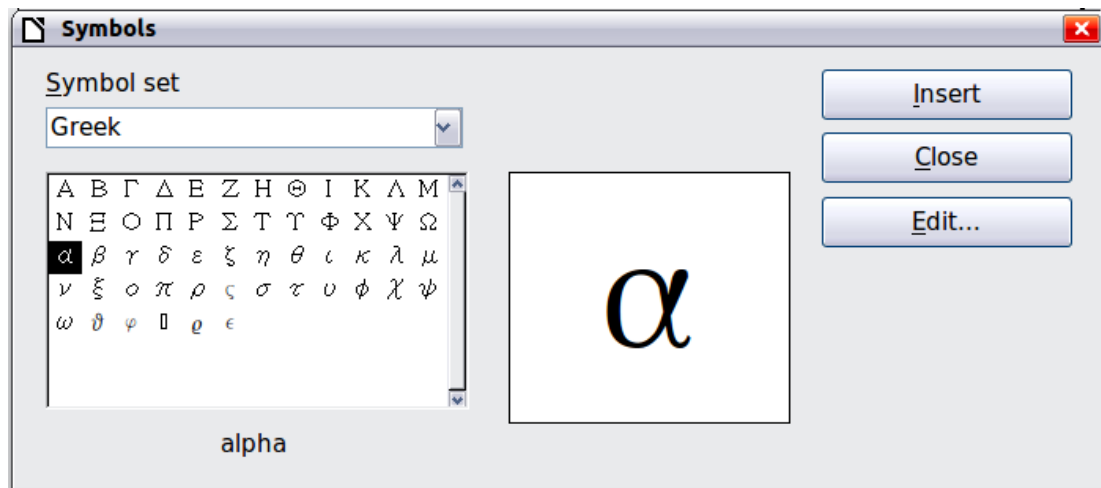


Figure 9: Symbols catalog, used for entering Greek characters and some special symbols

Example 2: $\pi \simeq 3.14159$

For this example we will suppose that:

- We want to enter the above formula (the value of pi rounded to 5 decimal places).
- We know the name of the Greek character (pi).
- But we do not know the markup associated with the \simeq symbol.

Step 1: Type % followed by the text **pi**. This displays the Greek character π .

Step 2: Open the Elements window (**View > Elements**).

Step 3: The \simeq symbol is a relation, so we click on the Relations button. If you hover the mouse over this button you see the tooltip *Relations* (Figure 10).

Figure 11 shows the Elements window after clicking the Relations button. The symbol we want is circled.

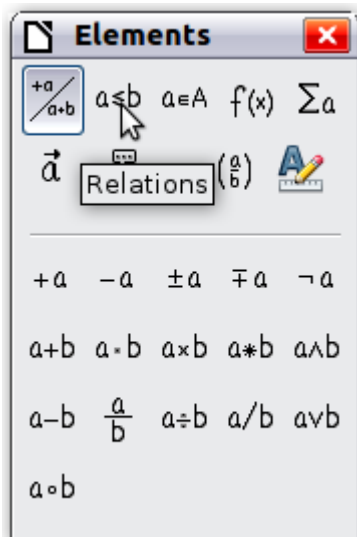


Figure 10: Tooltip indicates the Relations button

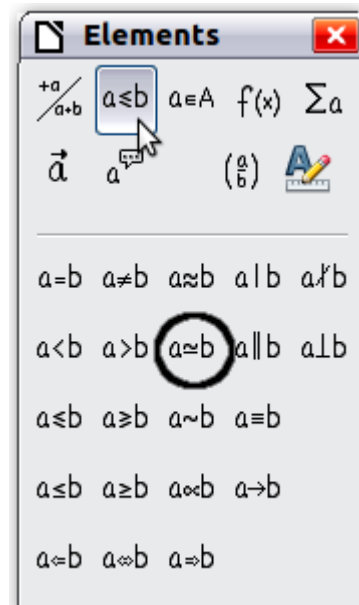


Figure 11: After selecting Relations

Step 4: Click on the $a \simeq b$ symbol. The equation editor now shows the markup `%pi<?> simeq <?>`.

Step 5: Delete the `<?>` text, press the *F4* key and type **3.14159** at the end of the equation. We end up with the markup `%pi simeq 3.14159`. The result is shown in Figure 12.

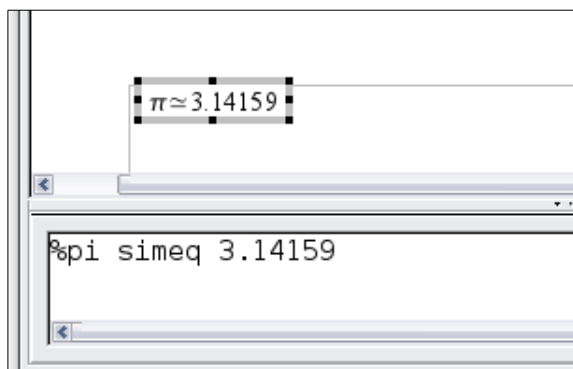


Figure 12. Final result

Changing a formula

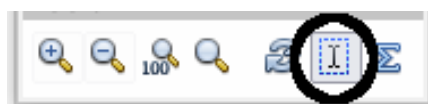
You can change a formula at any time. To switch into edit mode, double-click on the formula.

Choosing the region to alter

To get to the appropriate section in the markup code, do any one of the following:

- In the equation editor, click on the location.
- Select an area of the markup code that you wish to change.
- Click on an element in the preview area; the cursor will automatically move to the corresponding point in the equation editor.
- Double-click on an element in the preview area; the corresponding section in the equation editor will be selected.

To be able to work in the upper (preview) area in the stand-alone Math window (Figure 2), the formula cursor must be activated. Use the *Formula Cursor* button on the *Tools* toolbar.



Carrying out changes

You can change an equation by overwriting selected text or by inserting new markup code at the cursor position.

Tip

LibreOffice also provides a means of carrying out alterations directly in the preview area. This is still under development and therefore counts as “experimental”. In order to make use of it, you must first carry out the following configuration change: go to **Tools > Options > LibreOffice > General** and check the box for **Enable experimental (unstable) features**.

The activation of this option makes it impossible, by clicking on a formula element, to gain access to the corresponding position in the markup code.

Caution

The use of experimental features can lead to program crashes and/or loss of data. Only make the above change if you can accept this risk.

Formula layout

The most difficult part of using LibreOffice Math comes when writing complicated formulas. This section provides some advice.

Brackets are your friends

LibreOffice Math knows nothing about order of operation. You must use brackets to state the order of operations explicitly. Consider the following examples.

Markup	Result	Markup	Result
2 over x + 1	$\frac{2}{x}+1$	2 over {x + 1}	$\frac{2}{x+1}$
- 1 over 2	$-\frac{1}{2}$	- {1 over 2}	$-\frac{1}{2}$

In the first example, Math has recognized that the **2** before and the **x** after the **over** belong to the fraction, and has represented them accordingly. If you want **x+1** rather than **x** to be the denominator, you must bracket them together so that both will be placed there.

In the second example, Math has recognized the minus sign as a prefix for the **1** and has therefore placed it in the numerator of the fraction. If you wish to show that the whole thing is negative, with the minus sign in front of the fraction, you must put the fraction in brackets in order to signify to Math that the characters belong together.

The braces belong solely to the layout of the markup code and are not printed. If you wish to use braces in the formula, use the commands **lbrace** and **rbrace**.

Compare the following examples:

Markup	Result	Markup	Result
x over {-x + 1}	$\frac{x}{-x+1}$	x over lbrace -x + 1 rbrace	$\frac{x}{\{-x+1\}}$

Brackets with matrices look ugly!

For background, we start with an overview of the matrix command.

Markup	Result
matrix { a # b ## c # d }	$\begin{matrix} a & b \\ c & d \end{matrix}$

Note

Rows are separated by two #’s and entries within each row are separated by one #.

The first problem people have with matrices is that brackets do not scale with the matrix:

Markup	Result
<code>(matrix { a # b ## c # d })</code>	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

LibreOffice Math provides scalable brackets. That is, the brackets grow in size to match the size of their contents. Use the commands *left(* and *right)* to make scalable brackets.

Markup	Result
<code>left(matrix { a # b ## c # d } right)</code>	$\left(\begin{matrix} a & b \\ c & d \end{matrix} \right)$

Tip

Use *left[* and *right]* to obtain square brackets. The list of all available brackets is on page 54.

Tip

If you want all brackets to be scalable, go to **Format > Spacing**. Then in the *Spacing* dialog, choose the category *Brackets* and check the option **Scale all brackets**.

These scalable brackets may also be used with any element, such as fraction, square root, and so on.

Isolated and unpaired brackets

Math expects that for every opening bracket there will be a closing one. If you forget a bracket, Math places an inverted question mark by the corresponding bracket. The inverted question mark disappears when all the brackets are matched. Sometimes forgetting a bracket causes the whole structure of the formula to fall apart. However, an unpaired bracket is sometimes necessary. In such cases, you have two options.

With non-scalable brackets, use a preceding backslash `\` to indicate that the following character should not be regarded as a bracket but as a literal character. So the half-open interval `[a;b[` is represented by `\[a;b\[` — try comparing this with `[a;b[`

Scalable brackets can also be unpaired. The same half-open interval is represented by

`left [a; b right [`

For scalable brackets, you can also use the command **none** to replace a non-existent paired bracket.

$$|x| = \begin{cases} x & \text{for } x \geq 0 \\ -x & \text{for } x < 0 \end{cases}$$

can be represented by

`abs x = left lbrace stack {x "for" x >= 0 # -x "for" x < 0} right none`

Recognizing functions in Math

In the basic installation, Math outputs variables in italics. If you enter a function, Math usually recognizes it and outputs it normally. If Math fails to recognize a function (a list of recognized functions begins on page 48), you can inform Math about it. Enter the markup code **func** before the function, and the following text will be recognized as a function.

Some functions recognized by Math need to be followed by numbers or variables. If these are missing, Math puts an inverted red question mark ? in their place, which you can only remove by correcting the formula: enter a variable or a number, or a pair of empty braces $\{ \}$ as a placeholder.

Tip

You can navigate through errors using $F3$ and $Shift+F3$.

Equations over more than one line

Suppose you want to make an equation covering more than one line; for example: $x=3$
 $y=1$

Your first reaction would be to simply press the *Enter* key. However, if you press the *Enter* key, although the markup goes to a new line, the resulting equation does not. You must type the newline command explicitly. This is illustrated in the table below.

Markup	Result
$x = 3$ $y = 1$	$x=3y=1$
$x = 3$ newline $y = 1$	$x=3$ $y=1$

If you want to continue the calculation on a new line without writing a complete new equation, this is not directly possible, because Math expects a term on the left hand side of an equals sign. You can substitute:

- Empty quotes `""`. This will automatically cause the line to be left-justified.
- Empty braces `{ }`. The line will then be centered.
- Spaces characters ``` or `~`. The line will be centered with the spaces.

The alignment of equals signs under each other is described on page 23.

As well, spacing between elements in formulas is not set by space characters in the code. You need to use special markup to add spaces: ``` (grave) for a small space, `~` for a large space. Another solution would be to add space characters between quotes, to be considered as text. Space markup at the end of a formula are ignored by default (see “Space at the end of a formula” on page 38).

How do I add limits to my sum/integral?

The sum and int commands (see complete list on page 52) can (optionally) take the parameters *from* and *to*. These are used for lower and upper limits respectively. These parameters can be used singly or together.

Markup	Result
sum from $k = 1$ to n a_k	$\sum_{k=1}^n a_k$
int from 0 to x $f(t)$ dt or int_0^x $f(t)$ dt	$\int_0^x f(t)dt$ or $\int_0^x f(t)dt$

Markup	Result
<code>int from Re f</code>	$\int_{\Re} f$
<code>sum to infinity 2^{-n}</code>	$\sum_{-\infty}^{\infty} 2^{-n}$

Note For more details on integrals and sums, see page 52.

How do I write a derivative?

Writing derivatives essentially comes down to one trick: *Tell LibreOffice it is a fraction.*

In other words, you have to use the *over* command. Combine this with either the letter *d* (for a total derivative) or the *partial* command (for a partial derivative) to achieve the effect of a derivative.

Note Notice that we have to use braces (squiggly brackets) to make the derivative.

Markup	Result
<code>{df} over {dx}</code>	$\frac{df}{dx}$
<code>{partial f} over {partial y}</code>	$\frac{\partial f}{\partial y}$
<code>{partial^2 f} over {partial t^2}</code>	$\frac{\partial^2 f}{\partial t^2}$

To write function names with primes, as is usual in school notation, you must first add the signs to the catalog. Using single and double quotes is typographically ugly. See “Customizing the catalog” on page 36.

Markup characters as regular characters

Characters that are used for controlling markup cannot be entered directly as normal characters. The characters concerned are: `%`, `{`, `}`, `&`, `|`, `_`, `^` and `"`. So, for example, you cannot write **2% = 0.02** or **1" = 2.56cm**. Two methods are available to overcome this limitation:

- Use double quotes to mark the character as text, for example **2"" = 0.02**. Obviously this is not possible for the double-quote character itself.
- Add the character to the catalog. See “Customizing the catalog” on page 36.

In some cases you can use commands:

- **lbrace** and **rbrace** give you literal braces `{}`.
- **mline** gives you the vertical line, for example **2 mline 3** for $(2|3)$

Conversion into a character entity as in html or the use of an escape character is not possible in Math.

Text in a formula

To include text in a formula, enclose it in straight double-quotes:

abs x = left lbrace matrix {x # "for " x >= 0 ## -x # "for " x < 0} right none

$$|x| = \begin{cases} x & \text{for } x \geq 0 \\ -x & \text{for } x < 0 \end{cases}$$

All characters except double quotes are permissible in text. Unfortunately the Special characters dialog is not available. If necessary, you can write the text in a text document and copy it into the equation editor via the clipboard. In this way smart quotes can be inserted, as shown below.

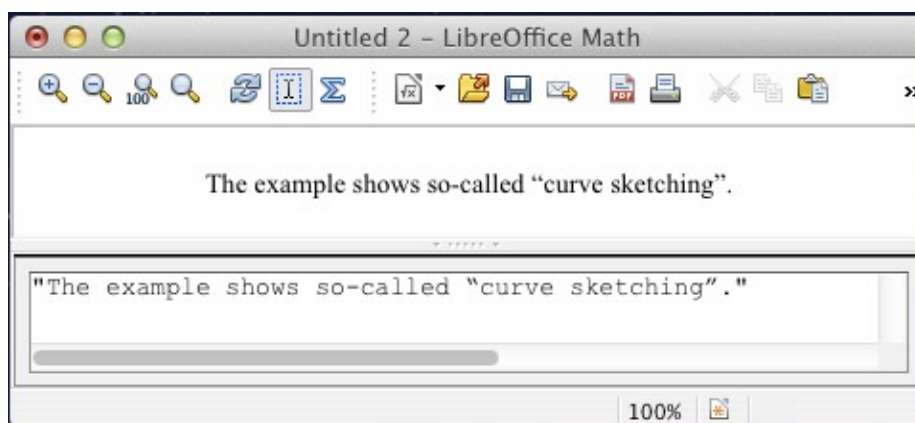


Figure 13: Smart quotes included by copy and paste from Writer.

Text is shown in the font that was selected from the **Text** list in the **Fonts** dialog (compare with the section “Changing the font” on page 25). To use a font from the lower window of the equation editor, set the attribute **Serif**, **Sans** or **Fixed** before the text.

By default, text is left-justified. You can change the justification with **alignc** or **alignr** (see “Changing the alignment” on page 28).

Commands are not interpreted within text. Use quotes to break up the text if you wish to use special formatting commands.

"In " color blue bold "isosceles" "triangles, the base angles are equal"

In **isosceles** triangles, the base angles are equal

How do I align my equations at the equals sign?

LibreOffice Math does not have a command for aligning equations on a particular character, but you can use a matrix to do this, as shown below.

Markup	Result
<pre>matrix{ alignr x+y # {}={} # alignl 2 ## alignr x # {}={} # alignl 2-y }</pre>	$\begin{array}{rcl} x+y & = & 2 \\ x & = & 2-y \end{array}$

The empty braces around = are necessary because = is a binary operator and thus needs an expression on each side. You may replace them with space characters (ˆ or ~).

You can reduce the spacing around = if you change the inter-column spacing of the matrix:

- 1) With the equation editor open, choose **Format > Spacing** from the menu bar.
- 2) In the Spacing dialog (Figure 14), click the **Category** button and select **Matrices** in the drop-down menu.
- 3) Enter **0%** for Column spacing and click **OK**.

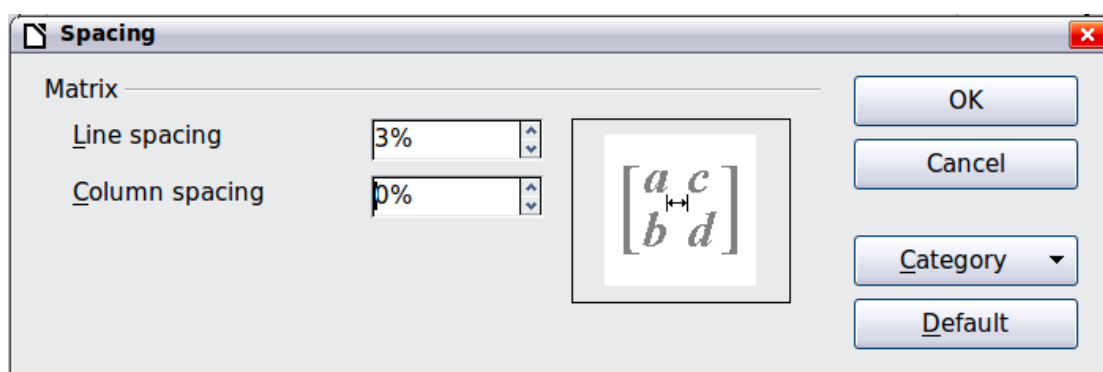


Figure 14: Changing spacing in a matrix formula

Changing the appearance of formulas

Changing the font size

This is one of the most common questions people ask about LibreOffice Math. The answer is simple, but not intuitive:

- 1) Start the formula editor and choose **Format > Font size**.
- 2) Select a larger font size under *Base size* (top-most entry).

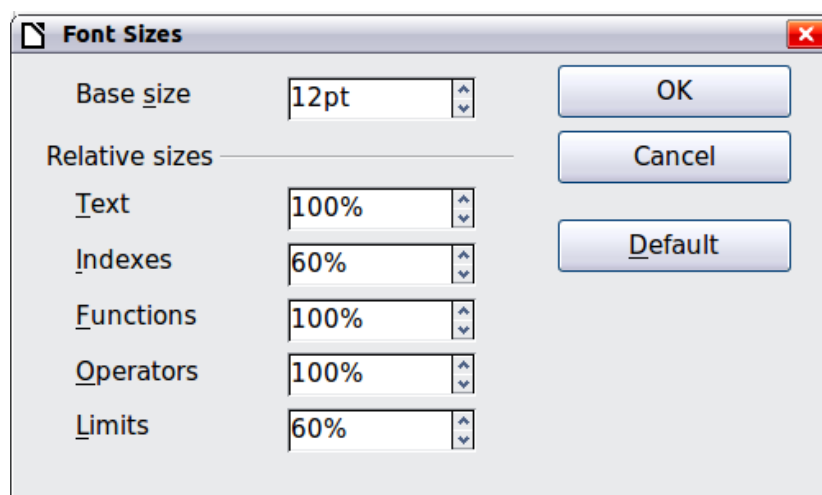


Figure 15. Edit Base size (top) to make a formula bigger

The result of this change is illustrated in Figure 16.

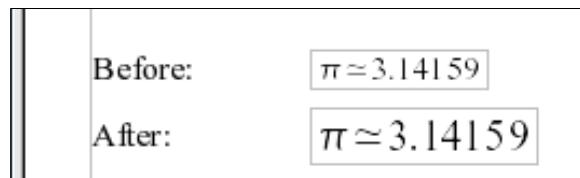



Figure 16. Result of changing the base font size

Note

The change in font size applies only to the current formula. To change the font size generally, click on the **Default** button and then **OK**. A general change in the font size might, for example, make your work easier when you are preparing a big presentation and want all the formulas in it to have a base size of 28pt—but do not forget to set the font size back to its original value when the work is finished.

Warning: this modifies only the current formula and future formulas you will write. To modify all formulas already existing in the document, you need to use a macro (see page 45).

The size of a subset of characters in a formula may be modified using the **size** command. For instance: **b size 5{a}** : b_a . In **Elements** window, last but one icon  of **Attributes** tab gives the **size** command. The value just after **size** may be absolute (numeric value) or relative to the context (base size by default): **+6**, **-3**, **/2**, or ***2** for instance.

Changing the font

The fonts used in formulas can be changed using **Format > Fonts**.

The *Formula fonts* section of the *Fonts* dialog (Figure 17) refers to the four specified formula elements. The font for operators, relationships and brackets is not affected, as these elements normally come from the OpenSymbol font. Similarly elements from the catalog (see “Customizing the catalog” on page 36) continue to be displayed in the font specified there.

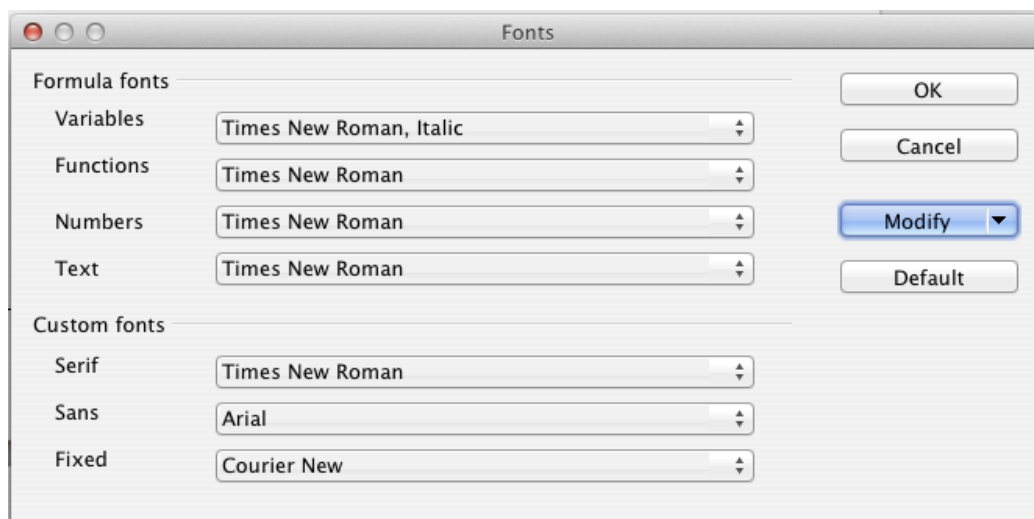


Figure 17: Fonts dialog

The *Custom fonts* section determines which font will be used when the attribute **font serif**, **font sans** or **font fixed** is specified.

To change a font, first click on the **Modify** listbox.



Choose the type of entry you wish to modify. The *Fonts* dialog opens, showing all the fonts available on your system.



Figure 18: Font dialog

If you enter an initial character in the upper text box, the list will jump to that place. By entering a few additional characters, you can specify the exact font that you want. If you do not know its name, use the scrollbar to scroll through them. Click on any name and the box below will show a preview.

Variables should be written in italics, so make sure that the *Italic* option is selected. For all other elements, use the basic (Roman) form. The style can be easily altered in the formula itself by using the commands **italic** or **bold** to set these characteristics and **nitalic** or **nbold** to unset them.

When you have chosen a new font for a formula, the old font remains in the list alongside the new one and can be selected again. This applies only to the current session; the old font is not stored permanently.

You can choose whatever fonts you like, but if you are exchanging documents with someone else, you should choose fonts that are present on your colleague's computer.

Adjusting spacing in formulas

To increase or decrease spacing in formulas, do the following:

- 1) Go to **Format > Spacing**.
- 2) In the *Spacing* dialog, click the triangle next to **Category**, and choose an entry from the list.

- 3) Choose appropriate spacing values (the types of spacing that can be set depend on the category) and click **OK**.

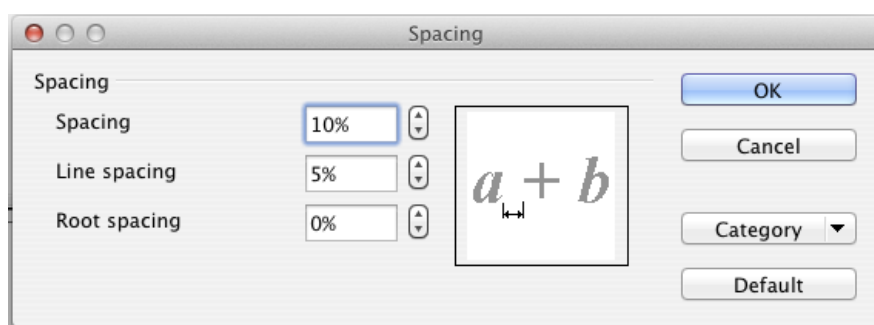



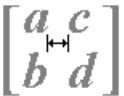


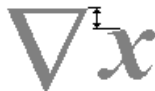
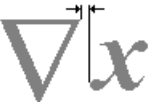
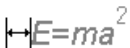

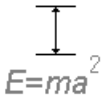
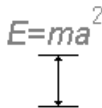
Figure 19: Spacing dialog

Note

Changes in spacing apply only to the current formula. To change spacing generally, click on the **Default** button and then **OK**.

Spacing changes are possible for the categories shown in the following table.

Category	Possible adjustment		
Spacing	$a + b$ Character spacing	$x = a$ $y = b$ Line spacing	$\sqrt{2}$ Root spacing
Indexes	m^2 Superscript	m_2 Subscript	
Fractions	$\frac{x}{y}$ Numerator	$\frac{x}{y}$ Denominator	
Fraction bars	$\frac{x}{y}$ Excesslength	$\frac{x}{y}$ Line thickness	
Limits	$\sum_{i=N}^{\quad}$ Upper limit	$\sum_{i=0}^{\quad}$ Lower limit	
Brackets	\int_x^y Excess size	\int_x^y Spacing	

Matrices	 Line spacing	 Column spacing		
Symbols	 Primary height	 Minimum spacing		
Operators	 Excess size	 Spacing		
Borders	 Left	 Right	 Top	 Bottom

Changing the alignment

The alignment settings determine how formula elements located above one another are aligned horizontally relative to each other. Use **Format > Alignment** to choose between *Left*, *Centered* or *Right* on a global basis.

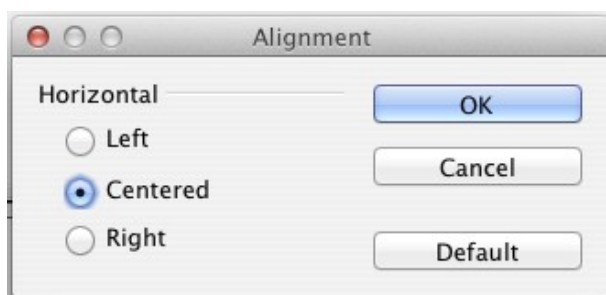


Figure 20: Alignment dialog

Here again you can use the **Default** button to make the change apply to all formulas and not just the current one. The following example shows the effect in different situations.

Alignment	Examples		
left	$\frac{x^2-9}{x}$	$\begin{pmatrix} 100 \\ 7 \end{pmatrix}$	$c^2=a^2+b^2-2ab\cos\gamma$ $\cos\gamma=\frac{c^2-a^2-b^2}{-2ab}$
centered (Default)	$\frac{x^2-9}{x}$	$\begin{pmatrix} 100 \\ 7 \end{pmatrix}$	$c^2=a^2+b^2-2ab\cos\gamma$ $\cos\gamma=\frac{c^2-a^2-b^2}{-2ab}$

Alignment	Examples
right	$\frac{x^2-9}{x}$ $\begin{pmatrix} 100 \\ 7 \end{pmatrix}$ $c^2=a^2+b^2-2ab\cos\gamma$ $\cos\gamma=\frac{c^2-a^2-b^2}{-2ab}$

It is not possible to align formulas on a particular character.

Default settings of alignment do not apply to text elements; they are always aligned left. In the following examples, the default alignment is *right* but the first example begins the second line with text (albeit empty text) and is therefore aligned left.

Markup Code	Result
<code>{1+2+3+4} over 5 + 2 over {60+70+80+90}</code> <code>newline</code> <code>""=2+1 over 150</code>	$\frac{1+2+3+4}{5} + \frac{2}{60+70+80+90}$ $= 2 + \frac{1}{150}$
<code>{1+2+3+4} over 5 + 2 over {60+70+80+90}</code> <code>newline</code> <code>{}=2+1 over 150</code>	$\frac{1+2+3+4}{5} + \frac{2}{60+70+80+90}$ $= 2 + \frac{1}{150}$

Independent of the default alignment, it is possible to align formulas explicitly using the commands **alignl**, **alignc** and **alignr**. This attribute also works for text elements.

Changing the color

Use the command **color** to change the color of a subset of a formula: **color red ABC** gives **ABC**. There are eight colors to choose between: **white**, **black**, **cyan**, **magenta**, **red**, **blue**, **green**, **yellow**.

You may give a color to a subset of a formula if it is gathered between { } or other parentheses. For instance: **A B color green {C D} E** gives **AB CDE**.

If several colors are used, the one the more inside the formula is applied, as in this example : **color blue {A B color yellow C D}** gives **AB C D**.

It is not possible to select background color: it is always transparent in Math. The background color of the whole formula is then the same as the one of the document or frame (in a text document for instance). In Writer, you can use object properties (right-click > **Object**) choose a background color for the whole formula (see "Background, borders, and size" on page 31).

Formulas in Writer documents

Numbering equations

Equation numbering is one of Writer's best hidden features. The steps are simple, but obscure:

- 1) Start a new line.
- 2) Type **fn** and then press *F3*.

The *fn* is replaced by a numbered formula:

$$E=mc^2 \quad (2)$$

Now you can double-click on the formula to edit it. For example, here is the Riemann Zeta function:

$$\zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z} \quad (3)$$

You can reference an equation (“as shown in Equation (2)”) with these steps:

- 1) Choose **Insert > Cross-reference** from the menu bar.
- 2) On the *Cross-references* tab (Figure 21), under *Type*, select *Text*.
- 3) Under *Selection*, select the equation number.
- 4) Under *Insert reference to*, select *Reference*.
- 5) Click **Insert**.

Done! If you later add more equations to the paper before the referenced equation, all the equations will automatically renumber and the cross-references will update.

Tip

To insert the equation number without parentheses around it, choose *Numbering* instead of *Reference* under *Insert reference to*.

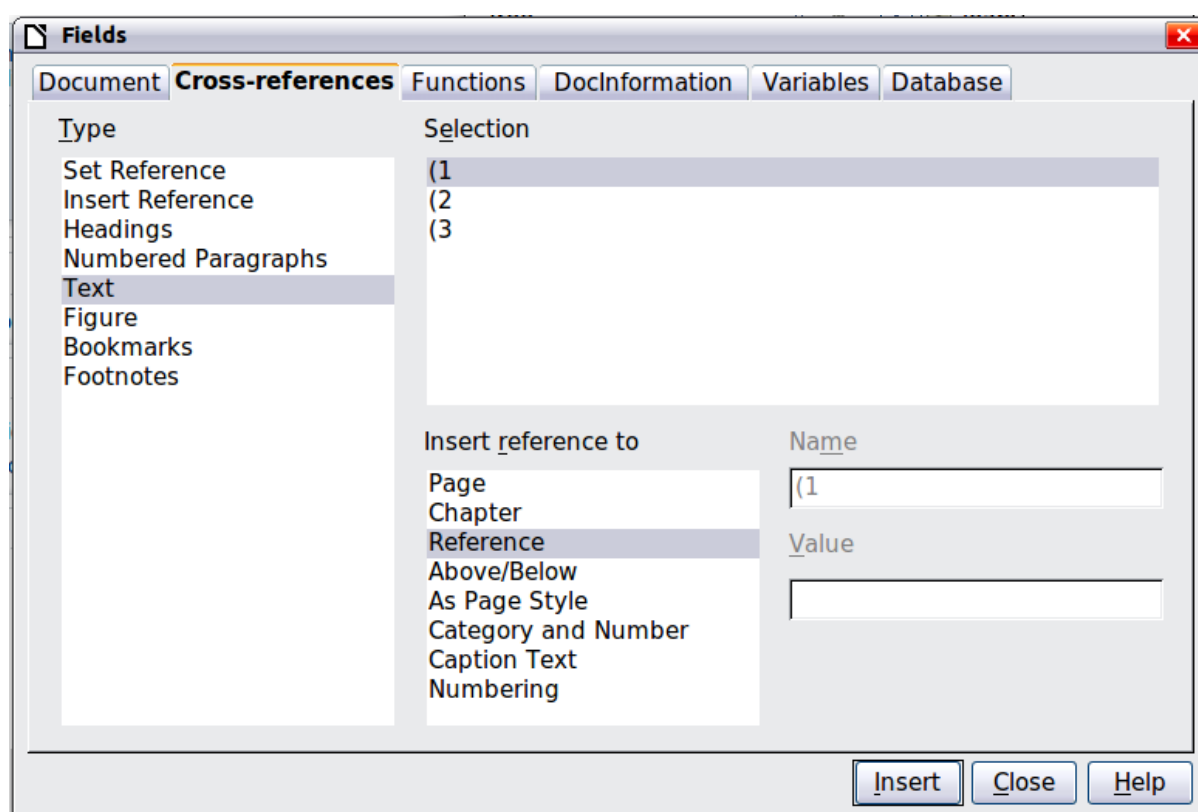


Figure 21. Inserting a cross-reference to an equation number

The AutoText inserted as a result of the above procedure consists of a 1x2 table. The left cell contains the formula and the right one the number. The number is an automatic counter named Text. You can edit the AutoText if, for example, you prefer square rather than round brackets, or if you want formula and number to be separated by tabs rather than formatted as a table. Read the section on “Using AutoText” in Chapter 3 in the *Writer Guide* (Working with Text).

Position

Normally a formula is anchored *As character* in a Writer document. But as with any other OLE object, you can change the anchoring mode to position the formula where you want it. For more information, see Chapter 11 of the *Getting Started* guide (Graphics, the Gallery, and Fontwork).

By default, formula objects anchored *As character* are automatically aligned vertically to the baseline of the surrounding text. To align the formula manually, go to **Tools > Options > LibreOffice Writer > Formatting Aids** and uncheck the option **Math baseline alignment**. This setting is stored with the document and applies to all formulas within it. New documents use the current setting from this dialog.

Margins

An inserted Math object has margins to left and right of it, separating it from surrounding text. If you do not want this, it is best to alter the frame style for formulas, as this will apply simultaneously to all existing formulas and those still to be inserted in the document. Proceed as follows:

- 1) Press **F11**. The *Styles and Formatting* dialog opens.
- 2) Click on the *Frame Styles* tab.
- 3) Find the *Formula* frame style and right-click on it.
- 4) Choose *Modify* from the context menu. The *Frame Style* dialog opens.
- 5) Switch to the *Wrap* tab of the dialog. Change the values for *Left* and *Right* in the Spacing section to 0.00.
- 6) Click **OK** to close the dialog.

This changes the spacing for all formulas that have not had their spacing adjusted manually. You can find more information on using styles in “Default layout with style” on page 39 and in Chapter 3 of the *Getting Started* guide (Using Styles and Templates).

Text mode

Larger formulas should always be in a paragraph of their own, separated from the text. When you use formula elements in running text, they can often be higher than the letter height.

However, if it is necessary to place a formula within running text, switch into the equation editor and go to **Format > Text** mode. Math will try to shrink the formula to fit the letter height. The numerators and denominators of fractions are shrunk, and the limits of integrals and sums are placed beside the integral/sum sign.

Example:

A formula in a separate paragraph:

$$\sum_{i=2}^5 i^2$$

and the same formula embedded in text mode: $\sum_{i=2}^5 i^2$

Background, borders, and size

With regard to formatting, formulas are treated as objects of the *Frame Style* type with the Formula frame style. Background color and borders can be set using this style or directly with **Format > Frame/Object**, or by right-clicking the formula and choosing **Object** from the context menu. In the basic installation, formulas have a transparent background and no borders. The size of a formula cannot be adjusted; in a Writer document it depends directly on the way the formula is constructed (see “Changing the font size” on page 24).

Creating a formula library

When you use the Math component of LibreOffice directly with **File > New > Formula**, you create documents with the file suffix .odf, each containing a single formula. You can use these to build up a library of frequently-used formulas. Embedded formulas can also be stored as separate Math documents by right-clicking on the formula and choosing **Save copy as** from the context menu. To insert such a Math document into a Writer document, use **Insert > Object > OLE Object**. Select the option *Create from file* and enter the pathname of the file or browse for it using your system's file manager by pressing the **Search** button.

Note

You cannot insert the document by dragging and dropping with the mouse, nor by using **Insert > File**.

Formulas cannot be stored in the gallery because they are not in graphical format. You can, however, store a formula as AutoText. Write the formula in a separate paragraph, select it, and go to **Edit > AutoText**. For further information see "Using AutoText" in Chapter 3 in the *Writer Guide* (Working with Text).

Fast insertion of formulas

If you already know the markup of your formula, there is a faster method to build your formula:

- 1) Write the formula markup in Writer.
- 2) Select this markup.
- 3) Insert the formula using the toolbar button, the menu, or a keyboard shortcut (see "Add keyboard shortcut" on page 35).

This method avoids opening and closing the Math window and saves a reasonable amount of time.

Formulas in Calc, Impress, and Draw

Graphical properties

Formula objects have similar properties in Calc, Impress, and Draw. They are always inserted with a transparent background and without borders. In Draw and Impress, they are assigned the *Default* graphical style; there is no assigned style for them in Calc and properties must be specified directly. The properties shown in the Styles and Formatting window, the context menu, and the **Format > Graphic** menu are limited, when not available at all.

Line, Area, Shadow

You can set values but the settings are completely ignored.

Text attributes

All text attributes, such as font and alignment, refer not to the formula text but to the text element present in all graphical objects. For a selected object, this can be accessed via the *F2* key. For further information, read the "Working with text in Draw" in Chapter 10 (Advanced Draw Techniques) of the *Draw Guide*.

Position

A formula can be positioned anywhere you like. Use the mouse, the arrow keys, or the Position and Size dialog available from **Format > Position and Size**.

Size

In documents a formula is not represented directly but replaced by a graphic. The size of this graphic is initially protected. To change the size, open the Position and Size dialog and, on the first page, in the *Protect* section, uncheck the **Size** checkbox. This activates the *Size* section directly above. Enter your modifications. When the dialog is closed, the size will immediately be protected again.

Changing the size does not affect the formula itself but only its graphical representation. In particular the basic font size of the formula (see “Changing the font size” on page 24) does not change. To revert to the size determined by the formula content, use the *Original Size* option in the context menu. This allows sizing errors that are sometimes caused by inserting a formula from a different module to be corrected.

Rotation, shear, and flipping

Rotation, shearing, and flipping (creating mirror images) are not possible; the corresponding dialog options are inactivated. If you need to do this, convert the formula to a *GDI metafile* graphic. Once you have done this, it is not a formula any more but an image. Copy the formula onto the clipboard. Choose **Edit > Paste Special** or, on the Standard Toolbar, use the Paste menu (available from the small triangle to the right of the Paste icon) and choose the option **GDI Metafile**.

Combining formulas with text

Since a formula is an OLE object, you cannot insert it into the content of a spreadsheet cell, graphic or presentation object. Thus, unlike in Writer, you cannot integrate formulas with running text. Here are some alternatives:

Text elements within the formula

You can write the text directly into the formula. For example:

The illustration of a parallelogram suggests the
relationship $\overrightarrow{AB} = \overrightarrow{CD} \Leftrightarrow \overrightarrow{AC} = \overrightarrow{BC}$.
But this does not constitute a proof.

The corresponding markup text is:

```
"The illustration of a parallelogram suggests the" newline  
"relationship " widevec AB = widevec CD dlrarrow widevec AC = widevec BC "." newline  
"But this does not constitute a proof."
```

Lines are not wrapped automatically; you must use the **newline** command.

You can use **Format > Fonts** to match the text to its surroundings. See “Changing the font” on page 25.

Grouping formulas with graphic objects

Formulas can easily be combined with other objects into a group, which is not possible in text documents. In the following example, text boxes are combined with formulas and rectangles are used to highlight parts of the formula.

Now, the **quadratic complement** is introduced

$$= x^2 - 6x + 3^2 - 3^2 - 7$$

and then elements are gathered

$$= \boxed{x^2 - 6x + 3^2} \boxed{- 3^2 - 7}$$
$$= (x - 3)^2 - 16$$

Formulas in Writer OLE objects

To be able to use all the facilities possible in text documents, create the formula and its text in Writer. Choose a page size that roughly matches your eventual objective. Do not write more than one page because page selection causes problems later. Save the document.

In Impress (or Draw or Calc), use **Insert > Object > OLE Object** and then choose the option *Create from file*. The formula is embedded as an OLE object in a document that is itself an OLE object. As a result, the formula cannot be modified in Impress. For that reason, you should always activate the option *Link to file*. Then if you need to make a change, open the text document and edit text and formula there. In Impress, use **Edit > Links** and then the **Update** button to display the stored version of the text document. Do not try to modify the formula in Impress; it causes the representation to become faulty.

Formulas in charts

A chart is itself an OLE object. Therefore you cannot call up the equation editor within it. There is no freestanding module of LibreOffice that creates charts, so you cannot use anything similar to the above method using Writer either. However you can create a formula outside the charts editor, copy it to the clipboard, activate the chart for editing, and paste in the formula from the clipboard. This automatically converts the formula into a metafile graphic. If you wish to make further modifications, you must discard it and repeat the procedure.

Customizations

Here are a few ways to customize your use of Math.

Formula editor as a floating window

The formula editor can cover a large part of the Writer window. To turn the formula editor into a floating window, do this:

- 1) Hover the mouse over the editor frame, as shown in Figure 22.
- 2) Hold down the *Control* key and double-click.

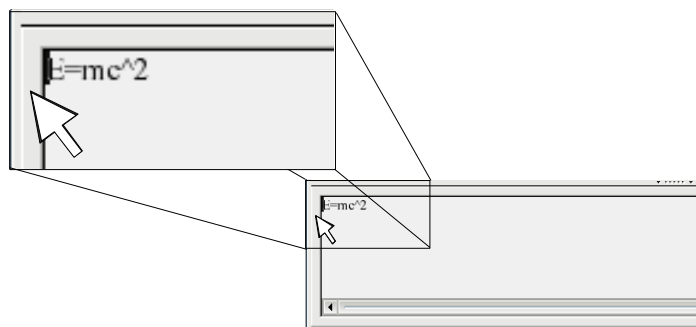


Figure 22: Turning the formula editor into a floating window

Figure 23 shows the result. You can dock the floating window again by using the same steps. Hold down the *Control* key and double-click the window frame.

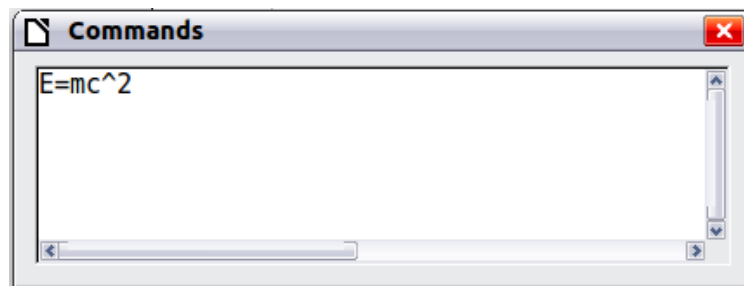




Figure 23: Formula editor as a floating window

Add button to toolbar

A **Formula** button is by default present on the **Insert** toolbar in *Writer* and *Impress*. You can add it to other toolbars:

- 1) **View > Toolbars > Customize** or right-click at the beginning of a toolbar and choose **Customize toolbar**. Choose the **Toolbars** tab and the toolbar you wish to modify.
- 2) Select the position where you want to insert the button.
- 3) Click on **Add**.
- 4) Select the **Insert** category, then the **Formula** command corresponding to icon  (the command  corresponds to an equivalent of a spreadsheet formula).

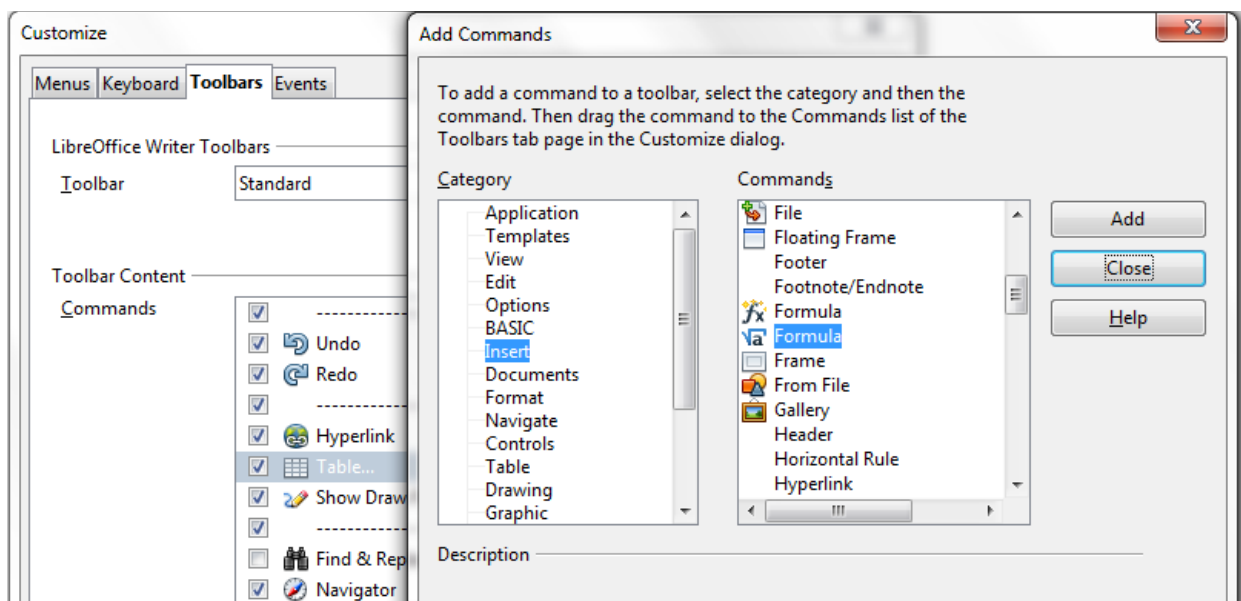


Figure 24: Add a button to insert a formula

- 5) Click on **Add**. You can continue to add buttons or click on **Close**.
- 6) On the Customize dialog, use the arrows to move buttons up or down the list before validating with **OK**.

Add keyboard shortcut

You can add a keyboard shortcut to insert a formula:

- 1) Menu **Tools > Customize, Keyboard** tab.

- 2) Select the level of the shortcut: *LibreOffice* to benefit from it in all components (if the shortcut is not used in this component) or only in the current component.
- 3) Select the **Insert** category, then the **Formula** command (the one not corresponding to **F2** shortcut; that command corresponds to the equivalent of a spreadsheet formula).
- 4) Select the keys to apply, then click on **Modify**.
- 5) Proceed the same way for other shortcut with the same level, then validate with **OK**.

If using your shortcut causes a formula bar to appear (as in Calc), that means you chose the wrong *Formula* command.

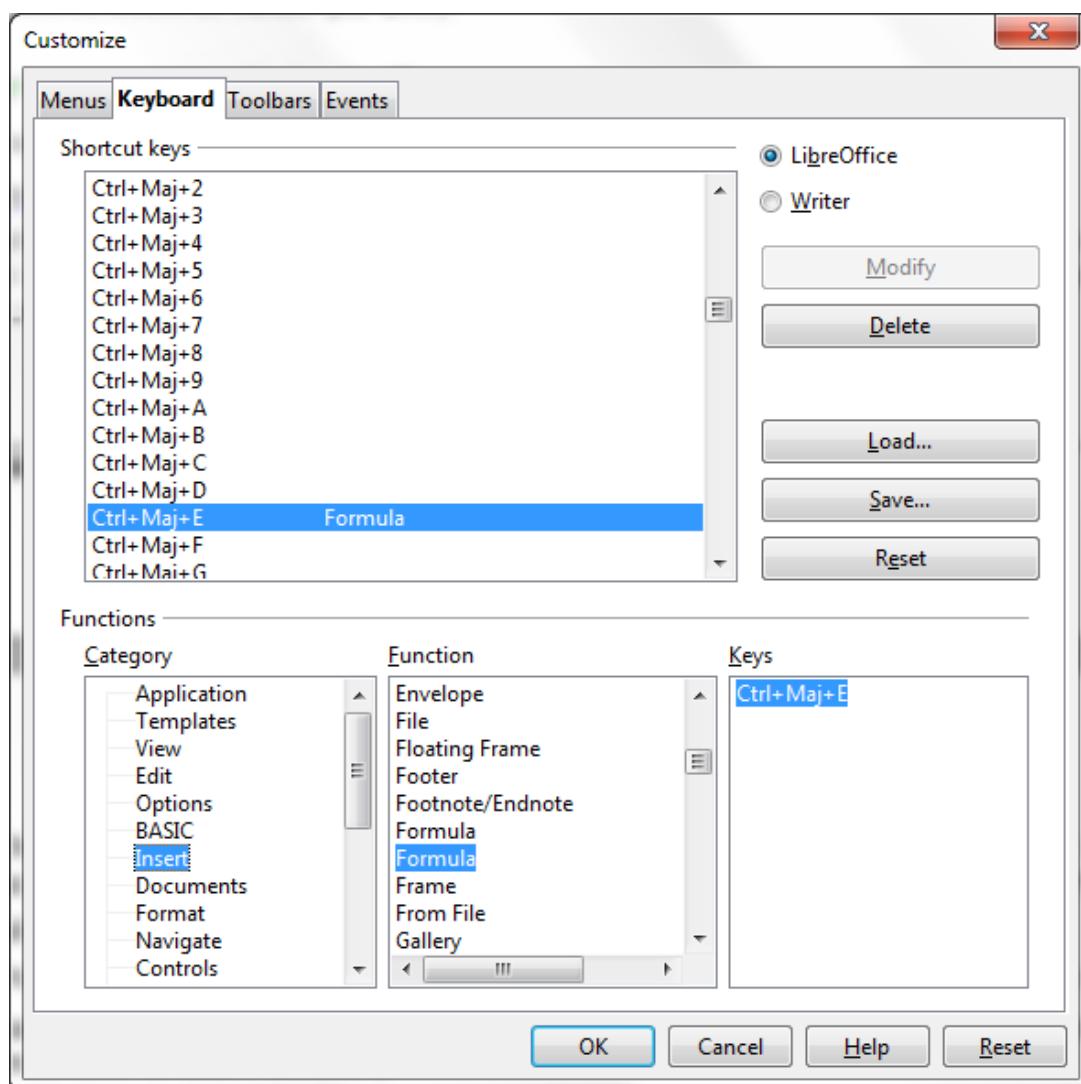



Figure 25: Add a keyboard shortcut

Customizing the catalog

If you need to use a symbol that is not available in Math, you can add it to the catalog. The following example shows the procedure for symbols that are used in marking school work.

- 1) Go to **Tools > Catalog** or click on the catalog button  to open the Symbols catalog (Figure 9).
- 2) Click the **Edit** button. The Edit symbols window (Figure 26) opens.
- 3) The symbols are arranged in so-called symbol sets. In the lower part of the window, choose an available set for your new symbol from the Symbol set list, for example the *Special* set. Or you can type the name of a new symbol set directly into the box.

- 4) From the Font list, choose a font that contains the desired symbol.
- When you have selected a font, its characters appear in the font summary window. You can scroll through it using the scrollbar at the side, or use the Subset list to go straight to the place you need.
- To follow this example, choose the font *DejaVu Sans* and the subset *General punctuation*.
- 5) Click on the desired symbol (here Ux2032). It appears enlarged in the right-hand preview box (see Figure 27). Make sure that the symbol set is set to **Special**.
 - 6) In the Symbol field, enter a name for the symbol, for example *prime*.
 - 7) If the name is not already in use, the **Add** button becomes active. Click on it.
 - 8) You can immediately add more symbols. In the example, the “U+2033” symbol, named *dblprime*, and the “U+2034” symbol, named *triprime*, are added. (See Figure 28.)
 - 9) Click **OK** to close the dialog.

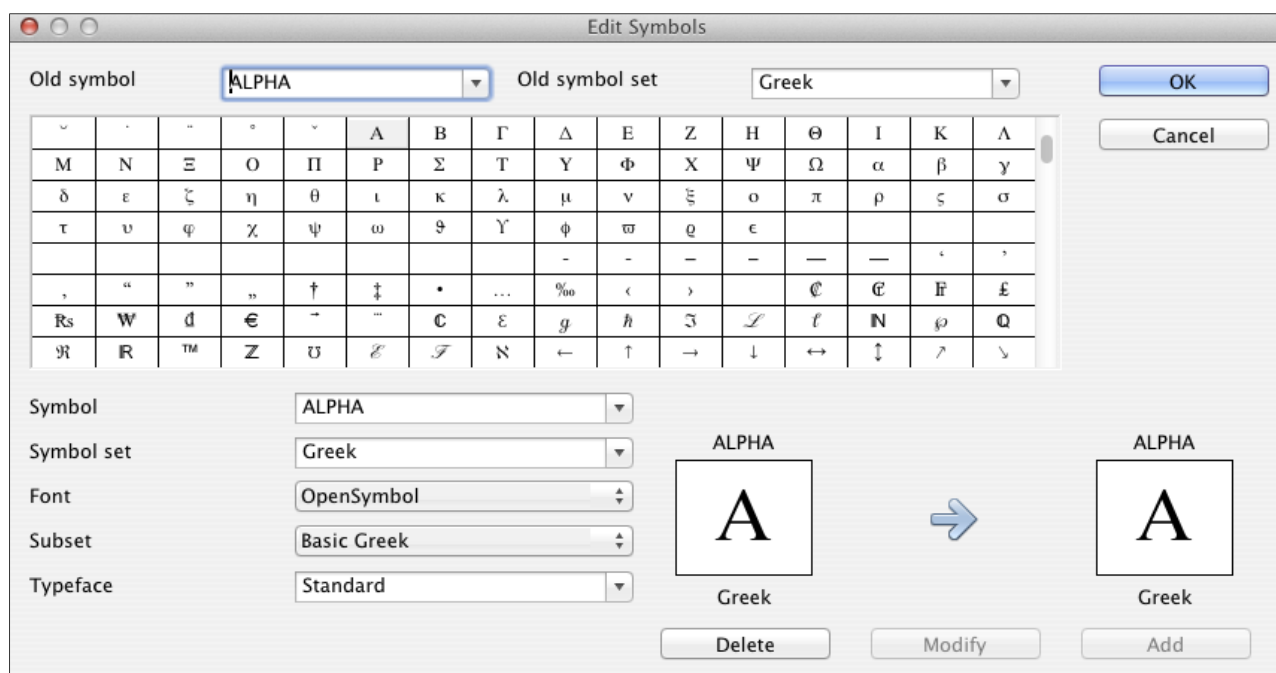


Figure 26: Edit Symbols dialog

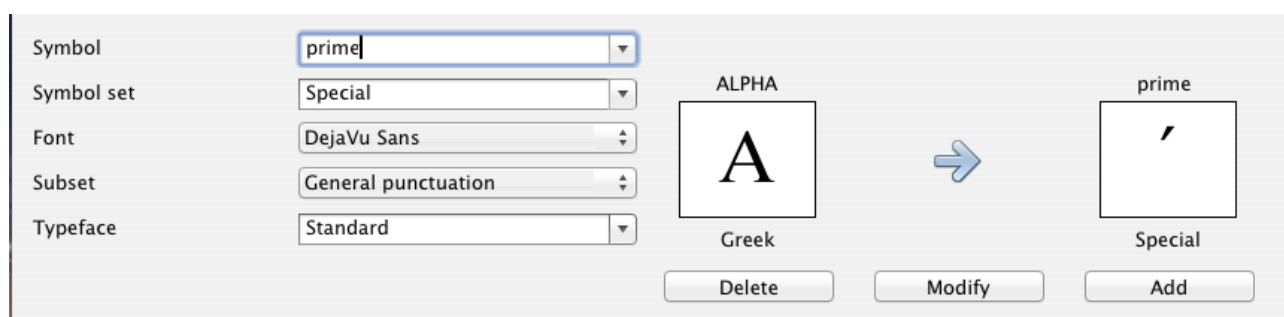


Figure 27: Edit Symbols dialog: characters selected

Now the Symbol set view area shows the new symbol. It can be selected just like the other symbols, either from here, or by writing them directly into the equation editor in the form %prime.

Caution



Symbols (Greek or Special), in contrast to commands, are case sensitive (lowercase – uppercase).

There are numerous free fonts containing a great number of mathematical symbols. The “STIX¹” font is worthy of special mention here. It was developed specially for writing mathematical/technical texts. The DejaVu² and Lucida³ fonts also have a wide range of symbols.

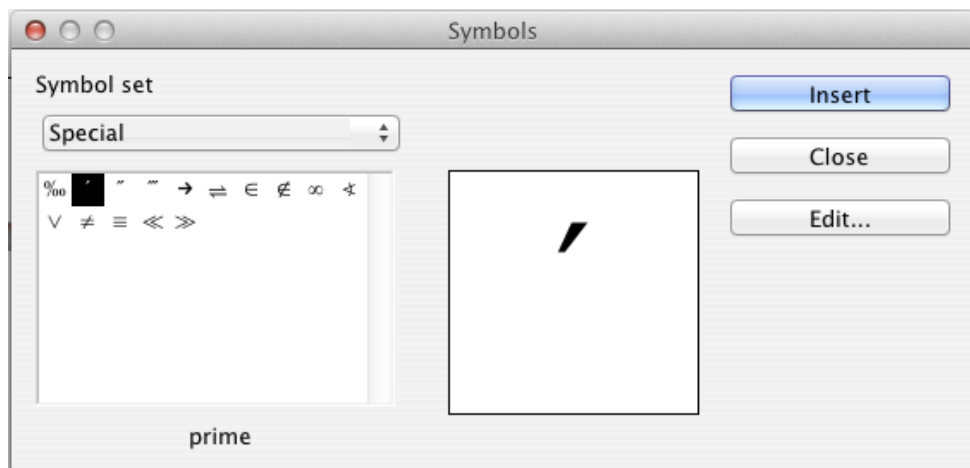


Figure 28: Catalog with new symbol

Note

Many symbols occur in more than one font. If you want to exchange documents with others, take care to use a font that is installed on their machine.

In the basic LibreOffice installation, only those user-defined symbols that actually occur in the document are stored with it. Sometimes it is useful to embed all the user-defined symbols, for example when the document is going to be further edited by another person. Go to **Tools > Options > LibreOffice Math** and on the **Settings** page, uncheck the option **Embed only used symbols (smaller file size)**. This setting is only available when you are working on a Math document.

Customizing operators

A name, to be recognized as a function to be written with regular font not italic, must be preceded by markup code **func**.

An operator, to be enlarged and to accept limits as integral or sum, must be preceded by markup code **oper**. It is possible to use a letter, a name or a personal symbol (see “Customizing the catalog” on page 36). $\sum_{i \in I} A_i$ is written **oper F from {i in I} A_i**.

You can modify the behavior of an operator or change one of your personal symbols in an operator, using markup codes **boper**, to create a binary operator, and **uoper**, to create a unary operator.

Space at the end of a formula

The grave accent ` inserts a small additional space into the formula; the tilde ~ inserts a larger one. In the basic installation, these symbols are ignored when they occur at the end of a formula. However, when working with running text, it may be necessary to include spacing at the end of formulas as well. This customization is only available when you are working with a Math document, and not when you are embedding a Math object.

1 Font files for STIX are available from <http://www.stixfonts.org>

2 Font files for DejaVu Sans are available from <http://www.dejavu-fonts.org>

3 Lucida Sans belongs to the JRE package, which is probably already on your computer.

Open a Math document with **File > New > Formula**. Open the customization dialog with **Tools > Options** and choose **LibreOffice Math**. On the **Settings** page, uncheck the option **Ignore ~ and at the end of the line**.

Default layout with style

In Writer, formulas are formatted according to the frame style *Formula*. In the Style and Formatting window (displayed with *F11*), click on the third button, *Frame Styles*. Right-click on **Formula** and select **Modify**. By this means, you can directly modify all formulas in your document, regarding spacing (page 31) or background (page 31), except if you manually modify formula formatting.

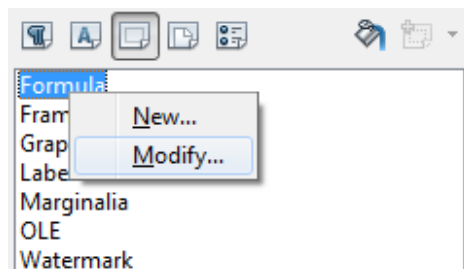


Figure 29 : Modify Frame Style Formula

To apply this style in all your new Writer documents, you must include the **Formula** style in your default template. To do so:

- 1) Create a new Writer document and modify the **Formula** frame style as you wish.
- 2) Save the document with **File > Save as Template**.
- 3) To define this template as your default template, select **File > New > Templates**. In the Template Manager dialog, open the folder containing the template that you want to set as the default, then select the template. Choose the **Set As Default** button above the list of templates (Figure 30.)

New text documents will be based on this template.

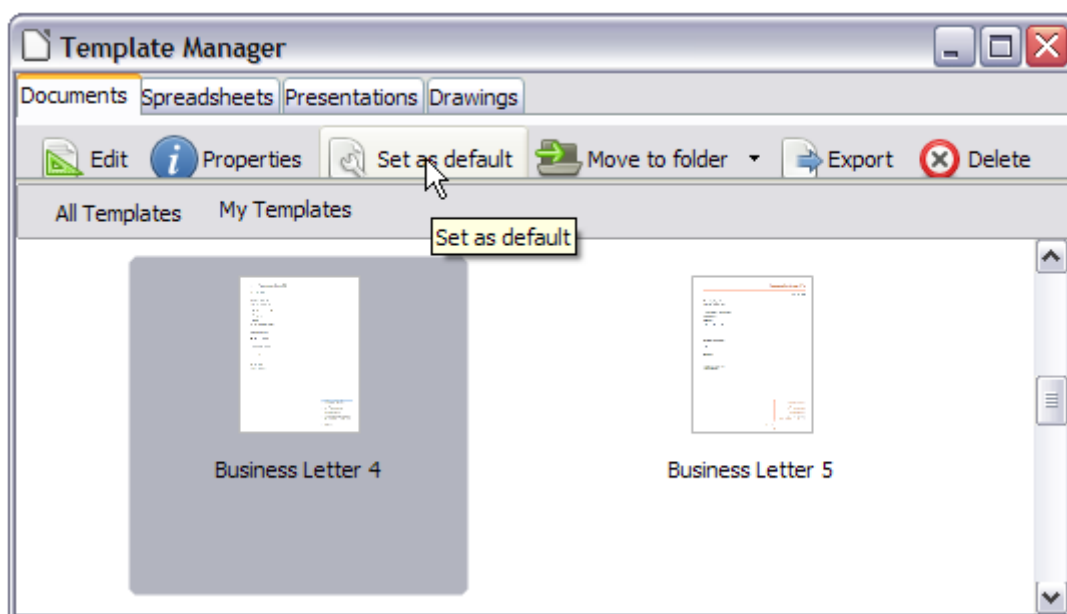


Figure 30: Setting a default template using the Template Manager dialog

To return to the default template supplied when LibreOffice was installed, go to the Template Manager dialog (Figure 31), click the **Action Menu** icon on the right, point to **Reset Default Template** on the drop-down menu, and click **Text Document**.

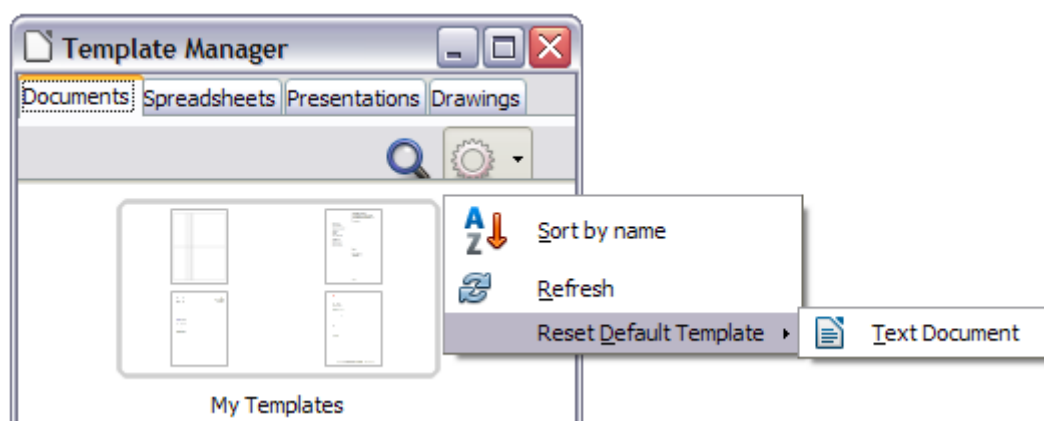


Figure 31: Resetting the default template

Application to chemical formulas

Math was designed to build mathematical formulas, but it can also be used to write chemical formulas. In chemistry, formulas look like H_2O : names are usually non-italic uppercase. To write chemical formulas with Math, begin with setting non-italic for variables (see “Changing the font” on page 25).

Below are some examples of chemical formulas.

Construction	Example	Entry
Molecules	H_2SO_4	H_2 SO_4 (please note the space!)
Isotopes	$^{238}_{92}\text{U}$	U 1sub 92 1sup 238
Ions	SO_4^{2-}	SO_4^{2- {}} or SO_4^{2" - "}

Notes : **1sub** or **1sup** are abbreviations for *left subscript* and *left superscript*. Empty braces after **2-** are necessary to avoid errors as there is no right member after the minus sign.

For reversible reactions, there is no satisfactory double arrow in Math. If you have a font with a correct symbol, use the method described in “Customizing the catalog” on page 36. For example, DejaVu fonts have these double arrows \rightleftharpoons \rightleftharpoons \rightleftharpoons \rightleftharpoons .

Otherwise, you can find a special character in a document and copy it, for instance in this formula:

$\text{C} + \text{O} \rightleftharpoons \text{CO}$ “ \rightleftharpoons ”. You will find other double arrows here:

<http://dev.w3.org/html5/html-author/charref> from character x021C4.

Export and import

Export as MathML

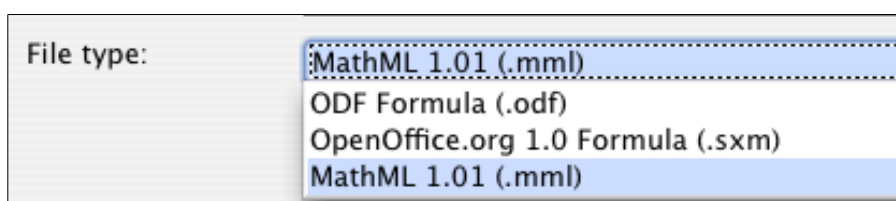
In addition to exporting documents as PDFs, as described in Chapter 10, Printing, Exporting, and Emailing, of the *Getting Started* guide, Math offers the possibility of exporting formulas as MathML. Unfortunately browsers do not follow this standard perfectly. If there are subsequent faults in display, the fault lies with your browser!

As an example, take this formula:

Markup	Result
<pre>%zeta (z) = sum from {n = 1} to infinity {1 over {n^z}} newline left(matrix {a # b ## c # d}right) newline "Text" newline int_0^x f(t) dt newline x = 3 y = 1</pre>	$\zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z}$ $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ Text $\int_0^x f(t) dt$ $x=3 \ y=1$

This formula combines various possibilities that Math offers, which makes it useful for testing.

If you are working in a separate Math document, go to **File > Save as** to open the Save as dialog. Choose MathML from the list of available file formats, to save your formula as MathML.



For an embedded formula, choose **Save copy as** from the context menu.

Result

This formula was subsequently tested in several browsers.

$\zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z}$ $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ Text $\int_0^x f(t) dt$ $x = 3y = 1$ Firefox 14.0.1	$\zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z}$ $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ Text $\int_0^x f(t) dt$ $x = 3y = 1$ Opera 12.00
<pre><?xml version="1.0" encoding="UTF-8"?> <math xmlns="http://www.w3.org/1998/Math/MathML"> <semantics> <table> <mtr> <mtd> <mrow> <mo stretchy="false">∑</mo> <mrow></pre> Internet Explorer 9	

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <semantics>
    <mtable>
      <mtr>
        <mttd>
          <mrow>
            <mo stretchy="false">{</mo>
          <mrow>
```

Chromium 13 and Google Chrome 20.0

Safari 5.1 only opens the file manager to save the data.

Microsoft file formats

The options in **Tools > Options > Load/Save > Microsoft Office** control how the import and export of Microsoft Office file formats is carried out in regard to formulas.

Loading

If [L] is checked, LibreOffice converts Microsoft formulas into native format when a document is loaded. This is possible if the formulas were created with MathType⁴ (up to version 3.1) or with the Microsoft Equation Editor. The latter is a reduced, older version of MathType licensed by Microsoft and included in the Microsoft Office package.

Formulas created by newer versions of MathType or by the new Microsoft formula editor “OMML⁵ Equation Editor” (Microsoft Office Suite 2010 and to some extent 2007) cannot be converted.

If a document created in Microsoft Office 2010 and containing an OMML formula is saved into a .doc file format, Microsoft Office converts the formula into a graphic. Only this graphic is then accessible to LibreOffice.

If you load a .docx document that contains OMML formulas, these fail in conversion whether [L] is checked or not.

Inside Microsoft Office, formulas created with MathType or Microsoft Equation Editor are treated as OLE objects. If [L] is not checked, LibreOffice maintains this. Double-clicking on the object launches MathType and new formulas can be inserted using **Insert > Object > OLE Object**. This setup is to be recommended if you have MathType installed and wish to use it to create and edit formulas.

Saving

If [S] is checked, LibreOffice converts the formula into a form that can be read and modified by Microsoft Equation Editor and MathType. When [S] is not checked, the formula is treated as an OLE object on conversion into a .doc format, and remains linked to LibreOffice. A double-click on the object in Word will therefore attempt to launch LibreOffice.

If you save in the .docx format, formulas are not converted, whether [S] is checked or not.

Opening OpenDocument texts in Microsoft Word 2010

When you use Microsoft Office 2010, it is possible to open LibreOffice files in .odt format. Microsoft Office 2010 reports an error but nevertheless displays a “repaired” document. In this document, any original formulas are converted to OMML. If you save it again in .odt format, Microsoft Office 2010 converts the formulas into MathML, and these formulas can then be edited in

⁴ <http://www.dessci.com/en/>

⁵ Office Math Markup Language

LibreOffice. Take care that symbols that belong together are enclosed in grouping brackets, even if these are not necessary for display in LibreOffice itself. This enables LibreOffice to correctly translate the formula into MathML. For example, use `sum from {i=1} to n {i^2}` rather than `sum from i=1 to n i^2`.

In this procedure, the annotation element is lost. As a result, LibreOffice no longer shows the original Math markup code in the equation editor, but instead generates a new markup code from the internal MathML notation.

XHTML

The XHTML file format is available only as an export format. For this reason you will find the option under **File > Export** rather than **File > Save as**. If this option is missing in your setup, you can install the necessary filter (run LibreOffice Setup, select **Modify**, and then select **Optional Components > XSLT Sample Filters**). The replacement graphic for the formula is embedded in the file, but the original Math markup code, included as an annotation element in MathML, is absent in this case (compare with the section on “Export as MathML” on page 40).

Whether the document is correctly displayed depends on the browser. Firefox shows the document correctly including the formulas. Opera and Internet Explorer have some problems with the embedded graphic.

The older filter “HTML Document (OpenOffice.org Writer)(.html)” does not save formulas at all but uses only a poorly resolved graphic in .gif file format.

Flat XML

This format can be both read and saved by LibreOffice. If the option is not present under **File > Save** or **File > Save as**, you can install the necessary filter. In this format, the whole content of the document, including the template, is saved within an XML document. In other words, it is not a compressed folder, as is usually the case for saved OpenDocument files. Any formulas are included as MathML elements, similar to the results of exporting as MathML. However individual formulas are not saved, only the whole document including the formulas. This format is suitable for direct entry into an external version control system.

Extensions

If you use Math frequently, it may be interesting to install one of these two extensions, which ease formula editing. Both programmers do not compete, but help one each other when it is possible. However, *Dmaths* and *CmathOOo* **cannot be simultaneously installed**. They are both complementary; you may install them on separate logins to test them before choosing.

Dmaths – an extension for more than just faster formula input

Dmaths is a mathematical macro package for LibreOffice Writer, which can be installed as an extension. It can be downloaded from <http://extensions.libreoffice.org/extension-center/dmaths>.

After installing Dmaths, close and reopen LibreOffice. Writer now shows a new menu (Figure 32), a new (customizable) toolbar with 36 buttons (Figure 33), a new toolbar with 26 buttons for use with Gdmath (Figure 34), a new toolbar with 10 buttons for use with AHmath3D (Figure 35), and two short toolbars to switch the longer ones on and off and to access a few further functions (Figure 36).

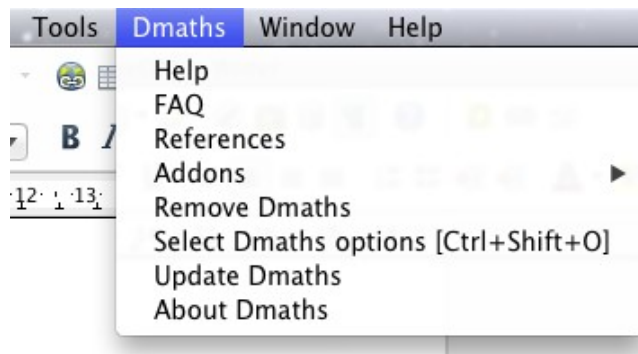


Figure 32: Dmaths menu

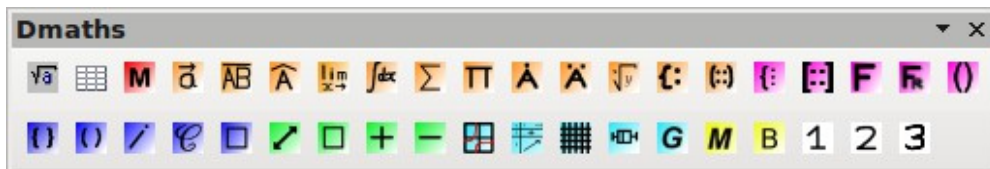


Figure 33: Customizable Dmaths toolbar



Figure 34: Gdmath toolbar



Figure 35: AHmath3D toolbar



Figure 36: Toolbars for switching on and off further functions

Dmaths offers numerous tools for creating mathematical documents:

- Single-click creation of Math formulas from text input (orange symbols)
- Creation of Math formulas using an input dialog (violet symbols)
- Tagging text (blue symbols)
- Changing the properties of existing formulas (green symbols)
- Drawing graphics, grids, statistical diagrams and geometrical figures (cyan symbols)
- Drawing and modification of geometric objects (gray Gdmath toolbar, also available in LibreOffice Draw)
- Drawing 3D objects (AHmath 3D toolbar with wire models)

The menu provides access to further add-ons, configuration, and a detailed *Dmaths Help Guide*.

Dmaths was initially created by the French mathematician, Didier Dorange, but is now available in German, English, and Spanish as well as French. The original website is www.dmaths.org.

Symbolic computation with CmathOOo and CmathOOoCAS

CmathOOo is a program that, thanks to word processing, allows you to enter mathematical expressions using same syntax as with a high school calculator. *CmathOOoCAS* extension enables symbolic computation as with Xcas⁶ directly from word processing; formatting and layout of the result is built with *CmathOOo*. This extension can be downloaded from the Apache OpenOffice extension website, <http://extensions.services.openoffice.org/fr/project/CmathOOo> or from Christophe Devalland's website <http://cdeval.free.fr>. Direct link to the page is <http://cdeval.free.fr/spip.php?article85>.

Technical details

OASIS Open Document Format for Office applications

The ODF specification does not lay down a completely new standard for Math formulas but takes over the Standard Mathematical Markup Language (MathML) developed by the World Wide Web Consortium (W3C) for representing formulas on websites (compare <http://www.w3.org/TR/2003/REC-MathML2-20031021>). Of the two types provided, Presentation markup and Content markup, ODF uses the former. It therefore describes the representation of a formula rather than its mathematical meaning.

The MathML markup language differs from the one used in the equation editor. So, for example, in the formula **int from {i=1} to 3 {f(t) {nitalic d}t}**, the parts **from... to...** are not translated but converted into MathML similarly to **int_{i=1}^3**. The original formula in the equation editor is a StarMath⁷ 5.0 encoding. To prevent it from being lost, it is stored as an annotation element. Other applications do not need either to evaluate nor to save this element, nor do they themselves create it.

In LibreOffice a formula is always associated with a replacement graphic. In the container formats – .odf, .odt, and so on – these graphics are stored in a separate folder and are internally linked to their document. However if you use the file format Open Document (Flat XML) or export to XHTML, these graphics must be embedded. For this purpose they are encoded in BASE64.

Handling of formulas in Basic

This section is not an introduction to Basic but simply describes some specific aspects of formulas. You can get more specific information about types and services at <http://api.libreoffice.org/common/ref/com/sun/star/module-ix.html>. If you look up the notations shown here in the index, you will quickly get to the relevant descriptions.

Formulas in a Writer document

In a Writer document, all OLE objects, including formulas, belong to a collection of type *SwXTextEmbeddedObjects*:

```
oCurrentController = ThisComponent.GetCurrentController()  
oTextDocument = oCurrentController.Model  
oEmbeddedObjects = oTextDocument.EmbeddedObjects
```

This collection is numbered continuously from 0. You can find a given object by:

```
nEndIndex = oEmbeddedObjects.Count - 1  
for nIndex=0 to nEndIndex  
    oMathObject = oEmbeddedObjects.GetByIndex(nIndex)
```

⁶ <http://www-fourier.ujf-grenoble.fr/~parisse/giac.html>

⁷ LibreOffice is based on the StarOffice suite, developed by StarDivision.

To determine if this object is actually a formula, test whether it supports a service that is supported by formulas:

```
oModel = oMathObject.Model
if oModel.supportsService("com.sun.star.formula.FormulaProperties") then
```

Alternatively test if the CLSID, which every OLE object possesses, is that belonging to a Math object:

```
if oMathObject.CLSID = "078B7ABA-54FC-457F-8551-6147e776a997" then
```

Using the model, you can then alter the properties of the formula, for example the basic font size:

```
oModel.BaseFontHeight = 12
```

To make this change visible, the formula must be redrawn. The

ExtendedControlOverEmbeddedObject property provides some methods that specifically apply to OLE objects:

```
oXCOEO = oMathObject.ExtendedControlOverEmbeddedObject
oXCOEO.update()
```

As an illustration, here is a macro modifying all formatting of all formulas already written in a document:

Sub ChangeFormatFormule

```
oCurrentController = ThisComponent.getCurrentController()
oTextDocument = oCurrentController.Model
oEmbeddedObjects = oTextDocument.EmbeddedObjects
nEndIndex = oEmbeddedObjects.Count-1
for nIndex=0 to nEndIndex
    oMathObject = oEmbeddedObjects.getByIndex(nIndex)
    oModel = oMathObject.Model
    if (not isNull(oModel)) then
        if(not isEmpty(oModel)) then
            if oModel.supportsService("com.sun.star.formula.FormulaProperties") then
                ' or if oMathObject.CLSID = "078B7ABA-54FC-457F-8551-6147e776a997" then
                    oModel.BaseFontHeight = 11
                    policeCommune= "Liberation Serif"
                    ' Variables
                    oModel.FontNameVariables= policeCommune
                    oModel.FontVariablesIsItalic=true
                    oModel.FontVariablesIsBold=false
                    ' Functions
                    oModel.FontNameFunctions = policeCommune
                    oModel.FontFunctionsIsItalic=false
                    oModel.FontFunctionsIsBold=false
                    ' Numbers
                    oModel.FontNameNumbers= policeCommune
                    oModel.FontNumbersIsItalic=false
                    oModel.FontNumbersIsBold=false
                    ' Text
                    oModel.FontNameText= policeCommune
                    oModel.FontTextIsItalic=false
                    oModel.FontTextIsBold=false
                    ' Update
                    oXCOEO = oMathObject.ExtendedControlOverEmbeddedObject
                    oXCOEO.update()
            endif ' if formula
        endif ' if not empty
    endif ' if not null
next nIndex
ThisComponent.reformat() ' Met à jour tous les éléments du document
End Sub
```

Check help page⁸ on *FormulaProperties* to know all other properties you can modify.

Formulas in a Draw/Impress/Calc document

In a Draw, Impress or Calc document, OLE objects are similarly treated as graphics. After you have accessed a particular object, test if it is an OLE object, and then if it is a formula. The core of your macro is then something like:

```
if oShape.supportsService("com.sun.star.drawing.OLE2Shape") then
  if oShape.CLSID = "078B7ABA-54FC-457F-8551-6147e776a997" then
    oModelFormula = oShape.Model
    oModelFormula.BaseFontHeight = 12
```

In this case an explicit update is not necessary.

⁸ <http://api.libreoffice.org/common/ref/com/sun/star/formula/FormulaProperties.html>

Math commands - Reference

Unary / binary operators

Operation	Command	Display
+sign	+1	$+1$
−sign	−1	-1
+/- sign	+−1 or plusminus 1	± 1 or ± 1
−/+ sign	−+1 or minusplus 1	∓ 1 or ∓ 1
Addition +	a + b	$a+b$
Subtraction (−)	a − b	$a-b$
Dot product	a cdot b	$a \cdot b$
Multiplication (X)	a times b	$a \times b$
Multiplication (asterisk)	a * b	$a * b$
Division (as a fraction)	a over b	$\frac{a}{b}$
Division (as an operator)	a div b	$a \div b$
Division (with a slash)	a / b	a/b
Concatenation	a circ b	$a \circ b$
Division (with a wide slash)	a wideslash b	$a \big/ b$
Wide backslash	a widebslash b	$a \big\backslash b$
Boolean not	neg a	$\neg a$
Boolean and	a and b or a & b	$a \wedge b$ or $a \wedge b$
Boolean or	a or b or a b	$a \vee b$ or $a \vee b$
Backslash	a bslash b	$a \backslash b$
Direct sum	a oplus b	$a \oplus b$
	a ominus b	$a \ominus b$
Tensorial product	a otimes b	$a \otimes b$
	a odot b	$a \odot b$
	a odivide b	$a \oslash b$
Customizable unary operator	uoper monOp b	$monOp b$
Customizable binary operator	a boper monOp b	$a \ monOp b$

Relations

Operation	Command	Display
Is equal	<code>a = b</code>	$a=b$
Is not equal	<code>a <> b</code> or <code>a neq b</code>	$a\neq b$ or $a\neq b$
Less than	<code>a < b</code> or <code>a lt b</code>	$a<b$ or $a<b$
Less than or equal to	<code>a <= b</code> <code>a leslant b</code>	$a\leq b$ $a\leq b$
Very small	<code>a ll b</code> or <code>a << b</code>	$a\ll b$ or $a\ll b$
Greater than	<code>a > b</code> or <code>a gt b</code>	$a>b$ or $a>b$
Greater than or equal to	<code>a >= b</code> <code>a geslant b</code>	$a\geq b$ $a\geq b$
Very big	<code>a gg b</code> or <code>a >> b</code>	$a\gg b$ or $a\gg b$
Approximately	<code>a approx b</code>	$a\approx b$
Similar to	<code>a sim b</code>	$a\sim b$
Similar to or equal	<code>a simeq b</code>	$a\approx b$
Congruent	<code>a equiv b</code>	$a\equiv b$
Proportional	<code>a prop b</code>	$a\propto b$
Parallel	<code>a parallel b</code>	$a\parallel b$
Orthogonal to	<code>a ortho b</code>	$a\perp b$
Divides	<code>a divides b</code>	$a b$
Does not divide	<code>a ndivides b</code>	$a\nmid b$
Toward	<code>a toward b</code>	$a\rightarrow b$
Arrow left	<code>a dlarrow b</code>	$a\leftarrow b$
Double arrow left and right	<code>a dlrarrow b</code>	$a\leftrightarrow b$
Arrow right	<code>a drarrow b</code>	$a\Rightarrow b$
Precedes	<code>a prec b</code>	$a<b$
Succeeds	<code>a succ b</code>	$a>b$
Precedes or equal to	<code>a preccurlyeq b</code>	$a\leq b$
Succeeds or equal to	<code>a succcurlyeq b</code>	$a\geq b$
Precedes or equal to	<code>a precsim b</code>	$a\lesssim b$
Succeeds or equal to	<code>a sucssim b</code>	$a\gtrsim b$
Does not precede	<code>a nprec b</code>	$a\nless b$
Does not succeed	<code>a nsucc b</code>	$a\ngtr b$
Definition	<code>a def b</code>	$a\stackrel{\text{def}}{=}b$
Image from	<code>a transl b</code>	$a\mapsto b$
Origin from	<code>a transr b</code>	$a\mapsto b$

Set operations

Operation	Command	Display
Is in	<code>a in B</code>	$a \in B$
Is not in	<code>a notin B</code>	$a \notin B$
Owns	<code>A owns b</code> or <code>A ni b</code>	$A \ni b$ or $A \ni b$
Intersection	<code>A intersection B</code>	$A \cap B$
Union	<code>A union B</code>	$A \cup B$
Difference	<code>A setminus B</code>	$A \setminus B$
Quotient	<code>A slash B</code>	A / B
Subset	<code>A subset B</code>	$A \subset B$
Subset or equal to	<code>A subseteq B</code>	$A \subseteq B$
Superset	<code>A supset B</code>	$A \supset B$
Superset or equal to	<code>A supseteq B</code>	$A \supseteq B$
Not subset	<code>A nsubset B</code>	$A \not\subset B$
Not subset or equal	<code>A nsubseteq B</code>	$A \not\subseteq B$
Not superset	<code>A nsupset B</code>	$A \not\supset B$
Not superset or equal	<code>A nsupseteq B</code>	$A \not\supseteq B$
Empty set	<code>emptyset</code>	\emptyset
Aleph	<code>aleph</code>	\aleph
Set of natural numbers	<code>setN</code>	\mathbb{N}
Set of integers	<code>setZ</code>	\mathbb{Z}
Set of rational numbers	<code>setQ</code>	\mathbb{Q}
Set of real numbers	<code>setR</code>	\mathbb{R}
Set of complex numbers	<code>setC</code>	\mathbb{C}

Functions

Operation	Command	Display
Absolute value	<code>abs{a}</code>	$ a $
Factorial	<code>fact{a}</code>	$a!$
Square root	<code>sqrt{a}</code>	\sqrt{a}
n^{th} root	<code>nroot{n}{a}</code>	$\sqrt[n]{a}$
Power	<code>a^{b}</code>	a^b
Exponential	<code>func e^{a}</code>	e^a
Natural logarithm	<code>ln(a)</code>	$\ln(a)$
Exponential function	<code>exp(a)</code>	$\exp(a)$
Logarithm	<code>log(a)</code>	$\log(a)$
Sine	<code>sin(a)</code>	$\sin(a)$
Cosine	<code>cos(a)</code>	$\cos(a)$
Tangent	<code>tan(a)</code>	$\tan(a)$
Cotangent	<code>cot(a)</code>	$\cot(a)$
Hyperbolic sine	<code>sinh(a)</code>	$\sinh(a)$
Hyperbolic cosine	<code>cosh(a)</code>	$\cosh(a)$
Hyperbolic tangent	<code>tanh(a)</code>	$\tanh(a)$
Hyperbolic cotangent	<code>coth(a)</code>	$\coth(a)$
Arcsine	<code>arcsin(a)</code>	$\arcsin(a)$
Arccosine	<code>arccos(a)</code>	$\arccos(a)$
Arctangent	<code>arctan(a)</code>	$\arctan(a)$
Arccotangent	<code>arccot(a)</code>	$\text{arccot}(a)$
Area hyperbolic sine	<code>arsinh(a)</code>	$\text{arsinh}(a)$
Area hyperbolic cosine	<code>arcosh(a)</code>	$\text{arcosh}(a)$
Area hyperbolic tangent	<code>artanh(a)</code>	$\text{artanh}(a)$
Area hyperbolic cotangent	<code>arcoth(a)</code>	$\text{arcoth}(a)$

Operators

All operators can be used with the limit functions (“from” and “to”).

Operation	Command	Display
Limit	<code>lim{a}</code>	$\lim a$
Lower limit	<code>liminf{a}</code>	$\liminf a$
Upper limit	<code>limsup{a}</code>	$\limsup a$
Sum	<code>sum{a}</code>	$\sum a$
Product	<code>prod{a}</code>	$\prod a$
Coproduct	<code>coprod{a}</code>	$\coprod a$
Integral	<code>int{a}</code>	$\int a$
Double integral	<code>iint{a}</code>	$\iint a$
Triple integral	<code>iiint{a}</code>	$\iiint a$
Contour integral	<code>lint a</code>	$\oint a$
Double curved integral	<code>llint a</code>	$\oint\!\!\!\oint a$
Triple curved integral	<code>lllint a</code>	$\oint\!\!\!\oint\!\!\!\oint a$
Lower bound shown with summation symbol	<code>sum from {3} b</code>	$\sum_3 b$
Upper bound shown with product symbol	<code>prod to {3} r</code>	$\prod^3 r$
Upper and lower bounds shown with integral	<code>int from {r_0} to {r_t} a</code>	$\int_{r_0}^{r_t} a$
Customized operator	<code>oper Op from 0 to 1 a</code>	$\overset{1}{Op}_0 a$

Attributes

Operation	Command	Display
Acute accent	<code>acute a</code>	á
Grave accent	<code>grave a</code>	à
Reverse circumflex	<code>check a</code>	ă
Breve	<code>breve a</code>	ă
Circle	<code>circle a</code>	å
Dot	<code>dot a</code>	à
Double dot	<code>ddot a</code>	ä
Triple dot	<code>dddots a</code>	ã
Line above	<code>bar a</code>	ā
Vector arrow	<code>vec a</code>	\vec{a}
Tilde	<code>tilde a</code>	ã
Circumflex	<code>hat a</code>	â
Wide vector arrow	<code>widevec abc</code>	\overrightarrow{abc}
Wide tilde	<code>widetilde abc</code>	\widetilde{abc}
Wide circumflex	<code>widehat abc</code>	\widehat{abc}
Line over	<code>overline abc</code>	\overline{abc}
Line under	<code>underline abc</code>	\underline{abc}
Line through	<code>overstrike abc</code>	\overleftrightarrow{abc}
Transparent (useful to get a placeholder of a given size)	<code>phantom a</code>	
Bold font	<code>bold a</code>	a
Not bold font	<code>nbold a</code>	a
Italic font ⁹	<code>ital "a" or italic "a"</code>	<i>a</i> or <i>a</i>
Not italic font	<code>nitalic a</code>	a
Font size	<code>size 16 qv</code>	qv
Font size	<code>size +12 qv</code>	qv
Font size	<code>size *1.5 qv</code>	qv
Following item in sans serif font ¹⁰	<code>font sans qv</code>	qv
Following item in serif font	<code>font serif qv</code>	qv

⁹ Unquoted text that is not a command is considered to be a variable. Variables are, by default, italicized.

¹⁰ There are three custom fonts: sans serif (without kicks), serifs (with kicks), and fixed (non-proportional). To change the actual fonts used for custom fonts and the fonts used for variables (unquoted text), numbers and functions, use **Format > Fonts** (see page 25).

Operation	Command	Display
Following item in fixed font	font fixed qv	<i>qv</i>
Make color of following text cyan ¹¹	color cyan qv	<i>qv</i>
Make color of following text yellow	color yellow qv	<i>qv</i>
Make color of following text white	color white qv	<i>qv</i>
Make color of following text green	color green qv	<i>qv</i>
Make color of following text blue	color blue qv	<i>qv</i>
Make color of following text red	color red qv	<i>qv</i>
Make color of following text black	color black qv	<i>qv</i>
Make color of following text magenta	color magenta qv	<i>qv</i>
Make color green returns to default color black	color green X qv	<i>Xqv</i>
Brace items to change color of more than one item	color green {X qv}	<i>Xqv</i>

Brackets

Operation	Command	Display
Group brackets (used for program control)	{a}	<i>a</i>
Round Brackets	(a)	(<i>a</i>)
Square Brackets	[b]	[<i>b</i>]
Double Square Brackets	ldbracket c rdbracket	[[<i>c</i>]]
Braces	lbrace w rbrace	{ <i>w</i> }
Angle Brackets	langle d rangle	⟨ <i>d</i> ⟩
Operator Brackets	langle a mline b rangle	⟨ <i>a b</i> ⟩
Upper half square brackets	lceil a rceil	[<i>a</i>]
Lower half square brackets	lfloor a rfloor	[<i>a</i>]
Single line	lline a rline	<i>a</i>
Double line	ldline a rdline	<i>a</i>
Scalable round brackets (add the word “left” before a left bracket and “right” before a right bracket)	left (stack{a # b # z} right)	$\left(\begin{matrix} a \\ b \\ z \end{matrix}\right)$
Square brackets scalable (as above)	left [a over b right]	$\left[\frac{a}{b}\right]$

¹¹ For all coloring, the color will apply only to the text immediately following the command until the next space is encountered. In order to have the color apply to more characters, place the text you want in color in curly brackets.

Operation	Command	Display
Double square brackets scalable	<code>left lbracket a over b</code> <code>right rdbar</code>	$\left[\frac{a}{b} \right]$
Brace scalable	<code>left lbrace a over b</code> <code>right rbrace</code>	$\left\{ \frac{a}{b} \right\}$
Angle bracket scalable	<code>left langle a over b</code> <code>right rangle</code>	$\left\langle \frac{a}{b} \right\rangle$
Operator brackets scalable	<code>left langle a over b</code> <code>mline c right rangle</code>	$\left\langle \frac{a}{b} \middle c \right\rangle$
Upper half square brackets scalable	<code>left lceil a over b</code> <code>right rceil</code>	$\left\lceil \frac{a}{b} \right\rceil$
Lower half square brackets scalable	<code>left lfloor a over b</code> <code>right rfloor</code>	$\left\lfloor \frac{a}{b} \right\rfloor$
Line scalable	<code>left lline a over b</code> <code>right rline</code>	$\left \frac{a}{b} \right $
Double line scalable	<code>left ldline a over b</code> <code>right rdline</code>	$\left\ \frac{a}{b} \right\ $
Unpaired brackets (add <code>left</code> before left bracket and <code>right</code> before right bracket)	<code>left langle a over b</code> <code>right rfloor</code>	$\left\langle \frac{a}{b} \right\rfloor$
Isolated bracket	<code>left lbrace stack</code> <code>{a=2#b=3} right none</code>	$\left\{ \begin{array}{l} a=2 \\ b=3 \end{array} \right.$
Over brace scalable	<code>{the brace is above}</code> <code>overbrace a</code>	$\overbrace{\text{the brace is above}}^a$
Under brace scalable	<code>{the brace is below}</code> <code>underbrace {f}</code>	$\underbrace{\text{the brace is below}}_f$

Formats

Operation	Command	Display
Right superscript	<code>a^{b}</code>	a^b
Right subscript	<code>a_{b}</code>	a_b
Left superscript	<code>a \sup{b}</code>	$^b a$
Left subscript	<code>a \sub{b}</code>	$_b a$
Center superscript	<code>a \csup{b}</code>	$\overset{b}{a}$
Center subscript	<code>a \csub{b}</code>	$\underset{b}{a}$
New line	<code>asldkfjo \newline sadkfj</code>	$\begin{array}{l} asldkfjo \\ sadkfj \end{array}$
Small gap (grave)	<code>stuff `stuff</code>	$stuff\,stuff$
Large gap (tilde)	<code>stuff~stuff</code>	$stuff\;stuff$
No gap	<code>nospace { x + y }</code>	$x+y$
Normal	<code>x+y</code>	$x+y$
Vertical stack of 2	<code>binom{a}{b}</code>	$\begin{array}{c} a \\ b \end{array}$
Vertical stack, more than 2	<code>stack{a # b # z}</code>	$\begin{array}{c} a \\ b \\ z \end{array}$
Matrix	<code>matrix{ a # b ## c # d }</code>	$\begin{array}{cc} a & b \\ c & d \end{array}$
Align character to left (text is aligned center by default)	<code>stack{Hello world # alignl(a)}</code>	$\begin{array}{c} \text{Hello world} \\ (a) \end{array}$
Align character to center	<code>stack{ Hello world # alignc(a)}</code>	$\begin{array}{c} \text{Hello world} \\ (a) \end{array}$
Align character to right	<code>stack{ Hello world # alignr(a)}</code>	$\begin{array}{c} \text{Hello world} \\ (a) \end{array}$
Equations aligned at '=' (using 'matrix')	<code>matrix{ a # "=" # alignl{b} ## { } # "=" # alignl{c+1} }</code>	$\begin{array}{rcl} a & = & b \\ & & = c+1 \end{array}$
Equations aligned at '=' (using 'phantom')	<code>stack{ alignl{a} = b # alignl{phantom{a} = c+1} }</code>	$\begin{array}{rcl} a & = & b \\ & & = c+1 \end{array}$

Others

Operation	Command	Display
Infinity	infinity or infty	∞ or ∞
Partial	partial	∂
Nabla	nabla	∇
There exists	exists	\exists
There does not exist	notexists	\nexists
For all	forall	\forall
H bar	hbar	\hbar
Lambda bar	lambdabar	$\bar{\lambda}$
Real part	re	\Re
Imaginary part	im	\Im
Weierstrass p	wp	\wp
Reversed epsilon	backepsilon	ϵ
Left arrow	leftarrow	\leftarrow
Right arrow	rightarrow	\rightarrow
Up arrow	uparrow	\uparrow
Down arrow	downarrow	\downarrow
Dots at bottom	dotslow	\cdots
Dots at middle	dotsaxis	\cdots
Dots vertical	dotsvert	\vdots
Dots diagonal upward	dotsup or dotsdiag	\ddots or \ddots
Dots diagonal downward	dotsdown	\ddots

Caution



In localized versions of Writer, the markup names of Greek and special characters are localized. If this document is not localized to the same language, then the names below *may not work* for input. You may still use the Symbol catalog (Figure 9) to select the desired character by its glyph. This will also display the character's localized markup name.

Once entered, the characters will display properly in any language.

Contrary to commands, special characters are case sensitive.

Characters – Greek

Italic characters can be obtained by adding an *i* after the per cent character. For instance, %iPI *Π* instead of %PI Π .

%ALPHA	Α	%BETA	Β	%GAMMA	Γ	%DELTA	Δ	%EPSILON	Ε
%ZETA	Ζ	%ETA	Η	%THETA	Θ	%IOTA	Ι	%KAPPA	Κ
%LAMBDA	Λ	%MU	Μ	%NU	Ν	%XI	Ξ	%OMICRON	Ο
%PI	Π	%RHO	Ρ	%SIGMA	Σ	%TAU	Τ	%UPSILON	Υ
%PHI	Φ	%CHI	Χ	%PSI	Ψ	%OMEGA	Ω		
%alpha	α	%beta	β	%gamma	γ	%delta	δ	%epsilon	ε
%varepsilon	ε	%zeta	ζ	%eta	η	%theta	θ	%vartheta	ϑ
%iota	ι	%kappa	κ	%lambda	λ	%mu	μ	%nu	ν
%xi	ξ	%omicron	ο	%pi	π	%varpi	ϖ	%rho	ρ
%varrho	ϱ	%sigma	σ	%varsigma	ς	%tau	τ	%upsilon	υ
%phi	φ	%varphi	φ	%chi	χ	%psi	ψ	%omega	ω

Characters – Special

Some special characters correspond to the same symbol as an operator but there is no syntax verification (presence of left or right elements).

%perthousand ‰	%tendto →	%element ∈
%noelement ∉	%infinite ∞	%angle ∠
%and ∧	%or ∨	%notequal ≠
%identical ≡	%strictlygreaterthan ≫	%strictlylessthan ≪

Reserved words in alphabetic order

A reserved word is an expression with a particular use and is controlled by LibreOffice. You cannot use it as a variable name. To be able to use these words without interference from Math, you need to place them between ". The commented list follows.

<code>`</code> : small space. See 'grave' operator.	<code>alignl</code> : left alignment
<code>^</code> : superscript. See 'hat' and 'widehat' operators.	<code>alignm</code> : (centered vertical alignment) ¹²
<code>_</code> : subscript	<code>alignr</code> : right alignment
<code>-</code> : subtraction	<code>alignt</code> : (top vertical alignment) ¹²
<code>-+</code> : '-+' sign with '-' over '+'	<code>and</code> : logical operator and
<code>.</code> : point; need a character before	<code>approx</code> : sign approximately, two ~ one over each
<code>(</code> : opening parenthesis	<code>arccos</code> : function arc cosine
<code>)</code> : closing parenthesis	<code>arccot</code> : function arc cotangent
<code>[</code> : opening square bracket	<code>arcosh</code> : area hyperbolic cosine
<code>]</code> : closing square bracket	<code>arcoth</code> : area hyperbolic cotangent
<code>{</code> : opening bracket for grouping characters	<code>arcsin</code> : function arc sine
<code>}</code> : closing bracket for grouping characters	<code>arctan</code> : function arc tangent
<code>*</code> : product	<code>arsinh</code> : function area hyperbolic sinus
<code>/</code> : division on one line	<code>artanh</code> : function area hyperbolic tangent
<code>\</code> : before a bracket (square bracket, parenthesis or brace), treat it as a normal character. See 'bslash' and 'setminus' operators.	<code>backepsilon</code> : upside down epsilon.
<code>&</code> : Boolean operator 'and'	<code>bar</code> : add bar to next character.
<code>#</code> : separation between elements in a table	<code>binom</code> : put 2 elements one over each
<code>##</code> : separation between lines in a matrix	<code>black</code> : black color for 'color'
<code>%</code> : indicate special name (displayed if name not recognized)	<code>blue</code> : blue color for 'color'
<code>%%</code> : comment	<code>bold</code> : bold font
<code>+</code> : addition	<code>boper</code> : next character is treated as a binary operator (unchanged size)
<code>+-</code> : '+-' sign with '+' over '-'	<code>breve</code> : half circle turned up
<code><</code> : operator less than	<code>bslash</code> : operator \, as difference
<code><?></code> : indicate placeholder around operators built with models. Displayed as a small square in the formula. <i>F4</i> and <i>Maj+F4</i> keys to navigate through them.	<code>cdot</code> : operator '.' for multiplication
<code><<</code> : operator much less than	<code>check</code> : upside down circumflex accent
<code><=</code> : operator less or equal	<code>circ</code> : 'round' operator for composition of functions
<code><></code> : operator different	<code>circle</code> : add a circle over next character
<code>=</code> : operator equality	<code>color</code> : set color: black, blue, cyan, green, magenta, red, white, yellow
<code>></code> : operator greater than	<code>coprod</code> : coproduct (upside down Π) or direct sum
<code>>=</code> : operator greater or equal	<code>cos</code> : function cosine
<code>>></code> : operator much greater than	<code>cosh</code> : hyperbolic cosine
<code> </code> : logical operator or	<code>cot</code> : cotangent
<code>~</code> : large space. See operator 'tilde'.	<code>coth</code> : hyperbolic cotangent
<code>abs</code> : function absolute value	<code>csub</code> : centered subscript
<code>acute</code> : acute accent for one character	<code>csup</code> : centered superscript
<code>aleph</code> : 1 st letter of Hebrew alphabet (cardinal number).	<code>cyan</code> : cyan color for 'color'
<code>alignb</code> : (bottom vertical alignment) ¹²	<code>dddots</code> : add three points over: triple derivative in physics
<code>alignc</code> : centered horizontal alignment	<code>ddots</code> : add two points over: double derivative in physics
	<code>def</code> : equal sign with DEF overwritten
	<code>div</code> : divide sign '/' with '-' in the middle
	<code>divides</code> : operator (Sheffer bar), same as logical sign 'or'
	<code>dldarrow</code> : left double arrow
	<code>dlrarrow</code> : left and right double arrow (equivalent)

¹² These obsolete shortcuts are for **align-bottom**, **align-mid** and **align-top**, (vertical alignments). *Math* recognizes them, but nothing is changed.

<code>dot</code> : add a point over: derivative in physics	<code>liminf</code> : inferior limit operator
<code>dotsaxis</code> : align ... horizontally in the middle	<code>limsup</code> : superior limit operator
<code>dotsdiag</code> : align three points with a +45° slop	<code>lint</code> : line integral (with a circle)
<code>dotsdown</code> : align three points with a -45° slop	<code>ll</code> : much lower than operator
<code>dotslow</code> : align ... on the bottom	<code>lline</code> : left bar to open absolute value ' '
<code>dotsup</code> : align three points with a +45° slop	<code>llint</code> : double line integral (with a circle)
<code>dotsvert</code> : align vertically 3 points	<code>lllint</code> : triple line integral (with a circle)
<code>downarrow</code> : down arrow	<code>ln</code> : function natural logarithm
<code>drarrow</code> : right double arrow (imply)	<code>log</code> : function decimal logarithm
<code>emptyset</code> : empty set	<code>lsub</code> : left subscript
<code>equiv</code> : equivalent (equal sign with three strikes)	<code>lsup</code> : left superscript
<code>exists</code> : there exist (reversed E)	<code>lt</code> : operator less than '<'
<code>exp</code> : exponential function	<code>magenta</code> : magenta color for 'color'
<code>fact</code> : factorial function (add '!' after)	<code>matrix</code> : defines a matrix
<code>fixed</code> : font attribute	<code>minusplus</code> : '−' sign, plus under '-'
<code>font</code> : select a font	<code>mline</code> : vertical line ' ' (angular bracket operator: langle ... mline ... rangle)
<code>forall</code> : whatever (upside down A)	<code>nabla</code> : nabla operator, upside down Δ.
<code>from</code> : goes with 'to' for limits of integrals, sums, etc.	<code>nbold</code> : not bold font.
<code>func</code> : transforms a variable in function	<code>ndivides</code> : does not divide operator, / vertically stroked
<code>ge</code> : greater or equal, horizontal '=' sign	<code>neg</code> : operator no
<code>geslant</code> : greater or equal, oblique '=' sign	<code>neq</code> : operator different
<code>gg</code> : much greater then '>>'	<code>newline</code> : new line
<code>grave</code> : add a grave accent	<code>ni</code> : inverted sign is element of
<code>green</code> : green color for 'color'	<code>nitalic</code> : not italic font
<code>gt</code> : operator plus grand que	<code>none</code> : combined with 'left' or 'right' to indicate invisible left or right bracket
<code>hat</code> : add a circumflex accent	<code>notin</code> : is not element of operator
<code>hbar</code> : strik h (ħ), reduced Planck's constant (divided by π)	<code>nroot</code> : n th root
<code>iiint</code> : triple integral, three times integral sign	<code>nsubset</code> : is not strictly include in operator
<code>iint</code> : double integral, twice integral sign	<code>nsubseteq</code> : is not include or equal in operator
<code>im</code> : function imaginary part	<code>nsupset</code> : reversed is not strictly include in operator
<code>in</code> : is element of	<code>nsupseteq</code> : reversed is not include or equal in operator
<code>infinity</code> : infinity symbol	<code>odivide</code> : operator / in a circle
<code>infty</code> : infinity symbol	<code>odot</code> : operator '·' in a circle
<code>int</code> : simple integral	<code>ominus</code> : operator '−' in a circle
<code>intersection</code> : intersection operator	<code>oper</code> : transforms next variable in a large operator with limits (like Σ)
<code>ital</code> : italicized font	<code>oplus</code> : direct sum operator, + in a circle
<code>italic</code> : italicized font	<code>or</code> : logical operator or, upside down ^
<code>lambdabar</code> : strike lambda	<code>ortho</code> : orthogonal operator, perpendicular symbol
<code>langle</code> : < to open '<...>' (angular bracket operator: langle ... mline ... rangle)	<code>otimes</code> : tensor product operator, x in a circle
<code>lbrace</code> : visible left brace '{'	<code>over</code> : division operator, to write division with horizontal fraction bar
<code>lceil</code> : left square bracket without bottom	<code>overbrace</code> : put next element over previous one with an horizontal brace
<code>ldbracket</code> : double left square bracket	<code>overline</code> : add an horizontal bar over next element
<code>ldline</code> : left double line (norm) ' '	<code>overstrike</code> : strike-through characters
<code>le</code> : less or equal, horizontal '=' sign	<code>owns</code> : reversed is element of
<code>left</code> : next character is treated as an opening bracket	<code>parallel</code> : parallel operator ' '
<code>leftarrow</code> : left arrow	<code>partial</code> : round d for partial derivative
<code>leslant</code> : less or equal, oblique '=' sign	<code>phantom</code> : invisible element, to let empty space
<code>lfloor</code> : left square bracket without top	
<code>lim</code> : limit operator	

plusminus : operator '+' with '+' over	sub : subscript
prod : product operator, Π	subset : strictly include operator
prop : proportional operator	subseteq : include or equal operator
rangle : '>' to close '<...>' (angular bracket operator: langle ... mline ... rangle)	sum : sum operator, Σ
rbrace : visible right brace	sup : superscript
rceil : right square bracket without bottom	supset : reversed strictly include operator
rdbracket : double right square bracket	supseteq : reversed include or equal operator
rdline : right double line (norm) ' '	tan : tangent function
re : real part function	tanh : hyperbolic tangent function
red : red color for 'color'	tilde : add a tilde '~' over next character
rfloor : right square bracket without top	times : multiplication operator, 'X'
right : next character is treated as a closing bracket	to : goes with 'from' for limits of integrals, sums, etc.
rightarrow : right arrow	toward : arrow to the right
rline : right bar to close absolute value ' '	transl : 2 small joined circles, the one of the left is filled (sign of correspondence Image from)
rsub : subscript	transr : 2 small joined circles, the one of the right is filled (sign of correspondence Origin of)
rsup : superscript	underbrace : put next element under previous one with an horizontal brace
sans : font option	underline : add an horizontal bar under next element
serif : font option	union : union operator, \cup
setC : set of complex numbers	uoper : next character is treated as a unary operator (unchanged size)
setminus : operator '\', subtraction of sets	uparrow : up arrow
setN : set of natural numbers	vec : add a small arrow over next character
setQ : set of rational numbers	white : white color for 'color'
setR : set of real numbers	widebslash : operator with a big '\'
setZ : set of integers	widehat : add an hat over next element
sim : operator equivalent, write one ~	wideslash : division operator with a big '/'
simeq : operator similarly equal, write a double ~	widetilde : add a tilde on next element
sin : function sine	widevec : add an arrow over next element
sinh : function hyperbolic sine	wp : Weierstrass p function
size : change font size	yellow : yellow color for 'color'
slash : operator slash '/'	
sqrt : operator square root	
stack : define a stack of elements separated with '#'	

Index

B

brackets (Math) 19
brackets commands (Math) 54

C

characters – special (Math) 58
characters – Greek (Math) 58
chemical formulas 40

D

derivative markup (Math) 22
Documentation 6

E

equation
 numbering 29
equation editor 11
 brackets 19
 derivative markup 22
 Elements window 13
 equations over more than one line 21
 floating window 34
 font size 24
 formula layout 19
 limits to sum/integral 21
 markup 15
 matrix markup 19
 right-click menu 15

F

formats commands (Math) 56
formula editor
 See: equation editor 34
formula layout 19

function commands (Math) 51

H

Help system 6

I

International Support 6

L

licensing of LibreOffice 9
limits to sum/integral 21

M

mathematical and chemical equations 11
mathematical markup 15
mathematical symbols 13
matrix markup (Math) 19
miscellaneous commands (Math) 57

N

numbering equations 29

O

Open Source Initiative (OSI) 9

R

relational operator commands (Math) 49

S

support 7

U

unary / binary operator commands (Math) 48
User guides 6

