This document collects all LyX Manuals picked up together into one PDF file. It targets the LyX 1.6(.4) release.

Currently there is no table of contents and it is not easy to automatically produce such a table or even bookmarks. Another drawback is that there will be a few glitches with the page numbering and formatting from time to time.

Currently these manuals are contained:

- Introduction to LyX
- The LyX Tutorial
- The LyX User’s Guide
- Additional LyX Features
- LyX’s detailed Figure, Table, Floats, Notes, Boxes and External Material
- LyX’s detailed Math manual
- Customizing LyX: Features for the Advanced User
- LyX Keyboard Shortcuts
- LFUNs documentation
Introduction to \textsc{LyX}

by the \textsc{LyX} Team\footnote{If you have comments or error corrections, please send them to the \textsc{LyX} Documentation mailing list: lyx-docs@lists.lyx.org}

August 17, 2009

Contents

1 The Philosophy of \textsc{LyX} 1
  1.1 What is \textsc{LyX}? 1
  1.2 \textsc{LyX} and Other Word Processors 3
  1.3 What is \textsc{LATEX}? 4

2 Navigating the Documentation 5
  2.1 The Format of the Manuals 6
  2.2 Units used in the Manuals 7
  2.3 The Manuals 8

3 Contributing to the \textsc{LyX} Project 9
  3.1 Contributing to \textsc{LyX} 9
  3.2 Contributing to the Documentation 10
1 The Philosophy of LyX

1.1 What is LyX?

LyX is a document preparation system. It excels at letting you create complex technical and scientific articles with mathematics, cross-references, bibliographies, indices, etc. It is very good at documents of any length in which the usual processing abilities are required: automatic sectioning and pagination, spell checking, and so forth. It can also be used to write a letter to your mom, though granted, there are probably simpler programs available for that. It is definitely not the best tool for creating banners, flyers, or advertisements (we’ll explain why later), though with some effort all these can be done, too. Some examples of what it is used for: memos, letters, dissertations and theses, lecture notes, seminar notebooks, conference proceedings, software documentation, books, articles in refereed scientific journals, scripts for plays and movies, business proposals, presentations . . .

LyX is a program that provides a modern approach to writing documents with a computer by using a markup language paradigm, an approach that breaks with the obsolete tradition of the “typewriter concept”. It is designed for authors who want professional output quickly with a minimum of effort without becoming specialists in typesetting. The job of typesetting is done mostly by the computer, not the author; with LyX, the author can concentrate on the contents of his writing.

Part of the initial challenge of using LyX comes from the change in thinking that you, the user, must make. At one time, all we had for creating documents were typewriters, so we all learned certain tricks to get around their limitations. Underlining, which is little more than overstriking with the “_” character, became a way to emphasize text. You were forced to figure out column sizes and tab stops, and set them, before creating a table. The same applied for letters and other right justified text. Hyphenation at the end of a line required a careful eye and a lot of foresight.

In other words, we’ve all been trained to worry about the little details of which character goes where. Consequently, almost all word processors have this mentality. They still use tab stops for adding whitespace. You still need to worry about exactly where on the page something will appear. Emphasizing text means changing a font, similar to changing the typewriter wheel. This is the underlying philosophy of a WYSIWYG word processor: “What You See Is What You Get”. Unfortunately, that paradigm often results in “What You See Is All You Get”.

This is where LyX differs from an ordinary word processor. You don’t concern yourself with what character goes where. You tell LyX what you’re
doing and LyX takes care of the rest, following a set of rules called a style.¹ Let’s look at a little example:

Suppose you are writing a report. To begin your report, you want a section called “Introduction.” So, you go into whatever menu it is in your word processor that changes font sizes and decide on a new font size. Then you turn on bold face. Then you type, “1. Introduction”. Of course, if you later decide that this section belongs someplace else in the document, or if you insert a new section before it, you need to change the numbering for this and all following sections, as well as any entry in the table of contents.

In LyX, you go to the pull-down on the far left of the button bar and select Section, and type “Introduction.”

Yes, that’s all. If you cut and paste the section, it will automatically be renumbered — everywhere. And if you enter references to that section correctly (by inserting cross-reference tags), LyX will automatically update them all throughout the file so that you never, ever type a section number.

Now let’s look at the problem of consistency. Five days later, you reopen your report and start Section 4. However, you forget that you were using 18pt bold instead of 16pt, so you type in the heading for Section 4 in a different font that what you used for Section 1. That problem doesn’t even exist in LyX. The computer takes care of all that silly bookkeeping about which thing has what size font, not you. After all, that’s what a computer is good at.

Here’s another example. Suppose you’re making a list. In other word processors, a list is just a bunch of tab stops and newlines. You need to figure out where to put the label for each list item, what that label should be, how many blank lines to put between each item, and so on. Under LyX, you have only two concerns: what kind of list is this, and what do I want to put in it. That’s it.

So, the basic idea behind LyX is: specify what you’re doing, not how to do it. Instead of “What You See Is What You Get,” the LyX model is “What You See Is What You Mean” or “WYSIWYM.” It’s a powerful idea that greatly simplifies the mechanics of writing documents. This is also why LyX isn’t so good for creating posters and flyers. In this case, you do want to specify exactly where everything goes, because there are no functional units like paragraphs, sections, etc. This doesn’t mean LyX is missing some cool function. It simply means that it isn’t the right tool for the job — you don’t use a screwdriver to drive in nails.

¹To be fair, most recent versions of the most popular office suites now have some sort of style sheets which follow a similar markup method. However, our experience is that they are still rarely used in practice.
1.2 Differences between \texttt{LYX} and Other Word Processors

Here’s a list of things you won’t find in \texttt{LYX}:

- The document ruler
- Tab stops
- Extra whitespace (i.e., hitting \texttt{Enter} or \texttt{Space} two or more times)

Tab stops, along with a ruler showing you the position of things on the page, are useless in \texttt{LYX}. The program worries about where things go on the page, not you. Extra whitespace is similar; \texttt{LYX} adds them where necessary, depending on context. Not being able to type two blank lines in a row will be annoying at first, but it makes more sense once you’re thinking in WYSIWYM terms.

Here are some things that exist in \texttt{LYX}, but aren’t used as you might think:

- Indenting controls
- Page breaks
- Line spacing (e.g., single spaced, double spaced, etc.)
- Whitespace, horizontal and vertical
- Fonts and font sizes
- Typefaces (bold, italic, underline, etc.)

Although they exist in \texttt{LYX}, you generally don’t need them. \texttt{LYX} will take care of these things for you, depending on what you’re doing. Different parts of the document are automatically set in a different typeface and font size. Paragraph indenting is context dependent; different types of paragraphs get indented differently. Page breaks get handled automatically, as well. In general, the space between lines, between words, and between paragraphs is variable, set by \texttt{LYX}.$^2$

Lastly, there are a few areas where we believe \texttt{LYX} (and \texttt{\LaTeX}) surpasses many word processors:

$^2$There are ways to adjust all of these (only some of which require knowledge of \texttt{\LaTeX}), either for a whole document or for a specific location in a document. See the User’s Guide and/or the Additional Features manual for details.
• Hyphenation
• Lists of any type
• Mathematics
• Tables
• Cross-referencing

Granted, many modern word processors can handle mathematical symbols, tables, and hyphenation, and many have moved towards style definitions and the WYSIWYG concept. However, they’ve only recently been able to do so, whereas \LaTeX{} is built upon the \LaTeX{} document preparation system. \LaTeX{} has been around for over 20 years, and works.

1.3 What is \LaTeX{}?

\LaTeX{} is a document preparation system designed by Leslie Lamport in 1985.\footnote{The source for the info in this section is “A Guide to \LaTeX{} 2ε,” by Helmut Kopka and Patrick Daly, which has an entry in the bibliography of the User’s Guide.} It was built up from a typesetting language called \TeX{}, created by Donald Knuth in 1984. \TeX{} takes a sequence of typesetting commands, written in a script in an ASCII file, and executes them. Many of the “tricks” of the printing trade were modeled by Knuth as computer algorithms and incorporated into \TeX{}, thus its excellent printed appearance. What comes directly out of \TeX{} is the portable document format pdf or the so-called “device independent” format file dvi. The dvi format is often used for previews and can later be converted to other formats like PostScript.

\TeX{} isn’t only a typesetting engine, it also allows you to define macros. Most people who use \TeX{} are actually using a macro package which Knuth created to hide a lot of the typesetting details. This is where Leslie Lamport enters our story. He wanted a macro package that was more user- and less typesetter-oriented, with a set of commands that consistently typeset things like sections, tables or math formulas in an uniform, consistent fashion. This is how \LaTeX{} was born.

Now, in parallel with the development and growth of \LaTeX{}, other folks were creating their own custom macro packages for \TeX{}, ones to make slides or articles for math journals and so on. Some used the raw \TeX{} facilities to do this, others began modifying \LaTeX{}. To try and unify this mess, a team of \LaTeX{}-nicians began to work on \LaTeX{} 2ε, the current version of \LaTeX{}, during the late 1980’s. This new version of \LaTeX{} has commands which provide an
easier-to-use interface to \TeX’s macro-creating commands, aid in the use of new fonts, and so on. In fact, \LaTeX{} is quite an extensive language in its own right! Users around the world have been creating their own add-ons for \LaTeX{} beyond the standard ones.

There are two ways to extend \LaTeX: classes and styles. A \textit{class} is a set of \LaTeX{} macros describing a new type of document, like a book, or an article. There are classes for slides, for physics and math journals... many universities even have a class for their thesis format! A \textit{style} differs from a class in that it doesn’t define a new type of document, but a different type of \textit{behavior} that any document can use. For example, \textsc{LyX} controls page margins and line spacing using two different \LaTeX{} style-files designed for these purposes. There are style-files for a whole slew of things: printing labels or envelopes, changing indentation behavior, adding new fonts, manipulating graphics, designing fancy page headings, customizing bibliographies, altering the location and appearance of footnotes, tables, and figures, customizing lists, etc.

Here is a summary:

\begin{itemize}
  \item \textbf{\TeX}: Typesetting language with macro capability.
  \item \textbf{\LaTeX}: Macro package built upon \TeX{}.
  \item \textbf{classes}: Descriptions of a type of document, using \LaTeX{}.
  \item \textbf{styles}: Alters the default behavior of \LaTeX{} in some way.
  \item \textbf{\textsc{LyX}}: Visual, WYSIWYM word-processor that uses \LaTeX{} to do its typesetting.
\end{itemize}

This section attempts to account for the difference between \textsc{LyX} and other word processors. Simply put, \LaTeX{} is the difference. By using \LaTeX{} as its backend, \textsc{LyX} helps you think more about \textit{what} (as in the words) you write. The computer then handles \textit{how} they should look.

\section{Navigating the Documentation}

To make it easier to answer your questions and describe all of the features of \textsc{LyX}, the documentation has been split up into several different files. Each one has its own purpose, as described below. Before you go plowing into any of those files, however, you should read this chapter thoroughly first, since it contains a lot of useful information and commentary that can save you some time.
The developing of LyX will hopefully never stop, so that some of the documentation may be incomplete or a bit out of date, though we try to keep up. Like the rest of LyX, the manuals are the work of a group of volunteers who have “Real Jobs”, families, dishes to clean, kitty litter to dispose of, et cetera. If you want to help out, be sure to read Section 3 in addition to the rest of this document.

Also, please do us a favor – if anything in these manuals confuses you, is unclear, or wrong, don’t hesitate to let us know! You can reach the current document maintainers by mailing to lyx-docs@lists.lyx.org. If you have questions which are not obviously answered in the documentation, and need help fast, there is an active users’ mailing list which you can reach at lyx-users@lists.lyx.org.

2.1 The Format of the Manuals

Some may have printed out the manuals. Others may be reading it within LyX. There are some differences between the LyX-file and the printed version. First, the title is simply at the top of the document, not formatted on a separate page as in some of the printed versions. Nor are any of the footnotes or the Table of Contents fully visible. To open a footnote, which looks like this: \footnote{1}, click on it with the left mouse button. For the Table of Contents, either click on the grey box or click on the Navigate menu, where the contents are displayed automatically.

In the printed manuals, all cross-references appear as the actual numbers for a chapter, section, subsection, and so on. Online, however, all cross-references appear as a light-grey box like the following: \ref{sec:Contrib}. If you click on that box with the left mouse button, a dialog box will appear containing a list of all the cross-references in the document. You can go to the referred section by right-clicking on the box or by clicking the button Go to Label in the opened dialog. Going back to where you came from is just as easy. Clicking on Go Back to go back to your earlier location.

Now that we’ve cleared up some of the differences between the printed and online versions of this file, we can start looking at the format of this document. You’ll occasionally notice things in different fonts:

- **Emphasized Style** is used for general emphasis, generic arguments, book titles, names of sections of other manuals, and notes from the authors.

- **Typewriter** is used for program and file names, LyX code and functions.
• Sans Serif is used for menu, button, or dialog box names, and the names of keyboard keys.

• Noun Style is used for people’s names.

• Bold is used for \LaTeX code

When we do need to reference keys, we’ll use the following prefixing convention:

• “Ctrl-” indicates the Control key.

• “Shift-” indicates the Shift key.

• “Alt-” indicates the Alt (Meta) key.

• “F1” . . . “F12” are the function keys.

• “Esc” is the escape key.


• “Insert” “Delete” “Home” “End” “PageUp” “PageDown”: these are the 6 keys that appear above the cursor keys on many PC keyboards. “PageUp” and “PageDown” are called “Prior” and “Next” on some keyboards.

• Return and Enter both refer to the same key. Some keyboards label the Return key as “Return,” others as “Enter,” still others have two keys. \LaTeX treats all of them as the same key, so we’ll use Return and Enter interchangeably.

The list with the currently set shortcuts can be found in the Help menu under Shortcuts.

2.2 Units used in the Manuals

To understand the units described in this documentation, Table 1 explains all units available in \LaTeX.
Table 1: Units

<table>
<thead>
<tr>
<th>unit</th>
<th>name/description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm</td>
<td>millimeter</td>
</tr>
<tr>
<td>cm</td>
<td>centimeter</td>
</tr>
<tr>
<td>in</td>
<td>inch</td>
</tr>
<tr>
<td>pt</td>
<td>point (72.27 pt = 1 in)</td>
</tr>
<tr>
<td>pc</td>
<td>pica (1 pc = 12 pt)</td>
</tr>
<tr>
<td>sp</td>
<td>scaled point (65536 sp = 1 pt)</td>
</tr>
<tr>
<td>bp</td>
<td>big point (72 bp = 1 in)</td>
</tr>
<tr>
<td>dd</td>
<td>didot (72 dd ≈ 37.6 mm)</td>
</tr>
<tr>
<td>cc</td>
<td>cicero (1 cc = 12 dd)</td>
</tr>
<tr>
<td>Scale%</td>
<td>% of original image width</td>
</tr>
<tr>
<td>text%</td>
<td>% of text width</td>
</tr>
<tr>
<td>col%</td>
<td>% of column width</td>
</tr>
<tr>
<td>page%</td>
<td>% of paper width</td>
</tr>
<tr>
<td>line%</td>
<td>% of line width</td>
</tr>
<tr>
<td>theight%</td>
<td>% of text height</td>
</tr>
<tr>
<td>pheight%</td>
<td>% of paper height</td>
</tr>
<tr>
<td>ex</td>
<td>height of letter $x$ in current font</td>
</tr>
<tr>
<td>em</td>
<td>width of letter $M$ in current font</td>
</tr>
<tr>
<td>mu</td>
<td>math unit (1 mu = 1/18 em)</td>
</tr>
</tbody>
</table>

2.3 The Manuals

The following list describes the contents of the basic documentation files that you find in the Help menu:

**Introduction**  This file.

**Tutorial** If you are new to LYX, and have never used \LaTeX before, you should start here. If you have used \LaTeX before, you should still read the *Tutorial*, starting with the section on “LYX for \LaTeX users.” (Skimming the rest of the document wouldn’t hurt, either.)

**User’s Guide** The primary documentation. We’ll cover most of the basic operation and available features of LYX here. The main manual assumes that you’ve read the *Tutorial*.

**Embedded Objects** Extension of the *User’s Guide*. Documents in detail how to use tables, graphics, floats, notes, program listings, and boxes. It also includes many tricks of the \LaTeX masters.
Math Extension of the User’s Guide. Documents in detail how to typeset any kind of formulas.

Additional Features Extension of the User’s Guide. Documents how to use raw \LaTeX\ commands, additional layouts, and special-purpose editing features.

Customization A description of advanced LyX features, including how to customize the overall behavior of LyX. This includes such things as keybindings, internationalization, and configuration files.

Shortcuts Tables with the currently defined LyX shortcuts.

\LaTeX\ configuration LyX investigates your system upon installation. This file contains info on what LyX learned about your installation. Check it to see if you’re missing something you might like to have.

These files will reference one another as necessary. For example, the User’s Guide contains some information on installation and customization, but refers the reader to the Customization Manual for more information.

We’ll state again an important point:

If you are new to LyX, read the Tutorial. Now.

Otherwise, you could needlessly frustrate yourself.

3 Contributing to the LyX Project

3.1 Contributing to LyX

LyX is mostly written in C++ (the \LaTeX\ importer is written in Python). It is a large project, and as a result it is not free from bugs, or the need for improvements in the source code.

3.1.1 Reporting a bug

While using LyX, you may find behavior which you consider a bug. Crashes, though rare, can happen. User interface problems are considered major bugs by the LyX team: especially helpful are indications of parts of the LyX interface you find confusing, or unclear.
LyX has a bug tracking system, which you can find at http://www.lyx.org/trac/wiki/BugTrackerHome. You should check the bug tracker before reporting any bugs, in case it has already been reported. If you have a comment on an existing bug, or wish to report a new bug, you may either use the bug tracker, or send an e-mail to the development mailing list, lyx-devel@lists.lyx.org. Archives of this list are linked from the main LyX website, http://www.lyx.org/.

A useful bug report will at a minimum include the version of LyX you are having the problem with. Accurate, detailed descriptions are preferred - the more time developers have to spend to pinpoint the source of a bug, the less time they have for other improvements. Mention the system and system version you are running LyX with. Give the versions of the libraries you have installed on your system, and, if relevant, the versions of external programs that LyX uses. If it’s a compilation or configuration problem, include the file config.log, and mention which compiler you are using.

3.1.2 Contributing fixes and new features

If you have made changes to LyX’s source that you think should become part of LyX, send your changes as a diff file (in unified format) to the development list referenced above, along with a change log, and a description of what your patch does.

3.2 Contributing to the Documentation

LyX’s documentation is extensive; however LyX is under constant development, and each new release adds new features. You may find some documentation needs improvement. This section describes what to do if you find an error, or have some suggestions for improving the documentation.

3.2.1 Reporting Errors in the Manuals

If you find a problem with the documentation, send a message to the mailing list lyx-docs@lists.lyx.org. The documentation team will make any necessary fixes.

3.2.2 Joining the Documentation Team.

The LyX Documentation Project, like anything else in the LyX project, can always use assistance! If you’re interested in contributing to the Documentation Project, you need to do the following:
1. Get the latest LyX source code from
   http://www.lyx.org/trac/browser/lyx-devel/trunk/lib/doc

2. Next, read the *User’s Guide* and the *Tutorial*
   The point of this exercise is to give you ideas. The *Tutorial* and *User’s Guide* is likely to be the most up-to-date of all of the documentation. You should be able to glean some insights into how we want the manuals to read and to look.

3. Contact the team at: lyx-docs@lists.lyx.org
to discuss your intended changes, and get some feedback on them.

The changes you wish to make may range from improving clarity of the text, to doing major re-structuring of the documentation. Any and all improvements are gladly received.
The LyX Tutorial

by the LyX Team

August 17, 2009

1If you have comments or error corrections, please send them to the LyX Documentation mailing list, lyx-docs@lists.lyx.org.
Contents

1 Introduction 42
   1.1 Welcome to LyX! ............................................ 42
   1.2 What the Tutorial is and what it isn’t .................. 42
       1.2.1 Getting the most out of the Tutorial ............... 43
       1.2.2 What you won’t find ................................. 43

2 Getting started with LyX 43
   2.1 Your first LyX document .................................. 43
       2.1.1 Typing, Viewing, and Exporting ..................... 44
       2.1.2 Simple Operations .................................... 44
       2.1.3 WYSIWYM: Whitespace in LyX ....................... 45
   2.2 Environments ................................................ 46
       2.2.1 Sections and Subsections .............................. 47
       2.2.2 Lists and sublists .................................... 48
       2.2.3 Other environments: Verses, Quotations, and more .... 50

3 Writing Documents 51
   3.1 Document Classes ........................................... 51
   3.2 Templates: Writing a Letter ............................... 52
   3.3 Document Titles ............................................. 53
   3.4 Labels and Cross-References .............................. 54
       3.4.1 Your first label ....................................... 54
       3.4.2 Your first cross-reference ............................ 55
       3.4.3 More fun with labels .................................. 55
   3.5 Footnotes and Margin Notes ............................... 56
   3.6 Bibliographies ............................................... 57
   3.7 Table of Contents .......................................... 58
4 Using Math
  4.1 Math Mode .......................................................... 59
  4.2 Navigating an Equation ............................................. 60
  4.3 Exponents and Indices .............................................. 61
  4.4 The Math Toolbar .................................................. 61
    4.4.1 Greek and symbols ............................................. 61
    4.4.2 Square roots, accents, and delimiters ....................... 62
    4.4.3 Fractions ..................................................... 62
    4.4.4 \TeX mode: Limits, log, sin and others .................... 63
    4.4.5 Matrices ..................................................... 63
    4.4.6 Display mode ................................................ 64
  4.5 More Math Stuff .................................................. 65

5 Miscellaneous
  5.1 Other major \LaTeX Features ..................................... 67
  5.2 \LaTeX for \LaTeX Users ......................................... 68
    5.2.1 \TeX Mode .................................................. 68
    5.2.2 Importing \LaTeX Documents — \texttt{tex2lyx} ............ 69
    5.2.3 Converting \LaTeX Documents to \LaTeX .................... 70
    5.2.4 \LaTeX Preamble ............................................ 70
    5.2.5 \BibTeX .................................................... 70
  5.3 Errors! ............................................................ 71
Chapter 1

Introduction

1.1 Welcome to LyX!

This file is designed for all of you who have never heard of \LaTeX, or don’t know it very well. Now, don’t panic - you won’t need to learn \LaTeX to use LyX. That is, after all, the whole point of LyX: to provide an almost-WYSIWYG interface to \LaTeX. There are some things you will need to learn, however, in order to use LyX effectively.

Some of you probably found your way to this document because you tried to put two spaces after a “.” or tried to put 3 blank lines between paragraphs. You found out you couldn’t and in fact, you’ll find out that most of the little tricks you’re accustomed to use in other word processors won’t work in LyX. That’s because most word processors you’ve used before allow you to manually enter all spacings, font changes, and so on. So you end up not only writing a document but typesetting it, too. LyX does the typesetting for you, in a consistent fashion, letting you focus on the important things, like the content of your writing.

So read on to learn more about LyX. Reading this tutorial is definitely worth the time.

1.2 What the Tutorial is and what it isn’t

Before we get started with this section, we want to make a quick note of something. The Tutorial uses the notation outlined in the Introduction manual. If you came to this manual first, please read the Introduction before you
continue with the Tutorial.

Now that you know which fonts mean what, we want to talk a bit about what this Tutorial is for.

1.2.1 Getting the most out of the Tutorial

This tutorial consists of examples and exercises. To get the most out of this document, you should read through the document, typing all the little things we’re telling you to type and trying out all of the exercises to see if you get them right. For convenience, you might want to print out the PDF version of this document.

If you are familiar with \LaTeX, you’ll probably be able to read the Tutorial somewhat faster, since many \LaTeX ideas are just \LaTeX ideas in disguise. However, \LaTeX have features you’ll want to learn about. Even if you don’t feel like reading the rest of the Tutorial, you should definitely check out Section 5.2, which is specifically written for experienced \LaTeX users.

1.2.2 What you won’t find

- Detailed explanations of all of \LaTeX’s features.
  Look in the User’s Guide when you need this.

- Detailed explanations of \LaTeX.
  Unnecessary. If you want to learn some of the neat tricks you can do with \LaTeX in \LaTeX, you can have a look at the Embedded Objects manual.

It’s time to move onwards, time for your first document . . .
Chapter 2

Getting started with \LaTeX

2.1 Your first \LaTeX{} document

OK. You’re ready to start writing. Before you do, there are a few things we need to mention, which will hopefully make the Tutorial more instructive and useful.

Because there is information we can’t give you in the Tutorial, the first thing that you need to do is find the other help files. This is very simple: Start up \LaTeX{}, Choose the User’s Guide from the Help menu. You may want to load the Tutorial as well (if you’re not reading it within \LaTeX{} already). This way, you can read them while you’re writing your own file.\footnote{They can also serve as good examples of how to use the many features of \LaTeX{}} Note that once you’ve got more than one document open, you can use the View menu or the document tabs to switch between them.

In this Tutorial, we’re going to assume that you have a fully working version of \LaTeX{}, as well as a \TeX{}-distribution, a DVI-, and a PDF-viewer. This should be the case on all major Linux- and BSD-distribution, as well as on Windows, where this is setup by the \LaTeX{} installer.

Finally, we’ve written a file to let you practice your \LaTeX{} skills on, it’s called example_raw.lyx. Imagine that it was typed by someone who didn’t know about any of \LaTeX{}’s great features. As you learn new \LaTeX{} functions, we’ll suggest that you fix those parts of example_raw.lyx. It also contains “subtle” hints about how to fix things.\footnote{The hints are located in yellow “Notes”. Access the text in a note by clicking on it.} If you want to cheat, or check what you’ve done, there’s also a file called example_lyxified.lyx which contains
the same text written and typeset by a LyX master.

The example files can be found in the examples directory of LyX’s installation folder. Open the raw document, and use File ▶ Save As to save a copy in your own directory for you to work on. As you fix parts of the raw document, check to see how those changes affect the DVI output.

By the way, the examples directory contains lots of other examples files. They will show you how to do various fancy things with LyX. After you read the Tutorial, or when you’re confused about how to do something fancy in LyX, take a look at these files.

2.1.1 Typing, Viewing, and Exporting

• Open a new file with File ▶ New

• Type a sentence like: This is my first LyX document!

• Save your document with File ▶ Save As.

• Run \texttt{\LaTeX} to create a DVI file, with View ▶ DVI or the toolbar button \texttt{△}. LyX will open a DVI-viewer program displaying your document looking like when printed.\footnote{You can save time by leaving the DVI-viewer running in the background. Then, you can use View ▶ Update ▶ DVI or the toolbar button \texttt{△} and just click on the DVI-viewer window (or unminimize it) after \texttt{\LaTeX} finishes running.}

• Export the ready to print document with File ▶ Export to a format you want.

Congratulations! You’ve written your first LyX document. All of the rest is just details, which is covered in the other manuals.

2.1.2 Simple Operations

LyX can of course do most of the things you’re used to do with a word processor. It will word-wrap and indent paragraphs automatically. Here’s a quick description of how to do some simple actions.

\textbf{Undo} LyX has multiple levels of undo, which means you can undo everything you’ve done since your current editing session started, by selecting
2.1. YOUR FIRST LyX DOCUMENT

Edit→Undo (toolbar button 🔄) over and over again. If you undo too much, just select Edit→Redo (toolbar button ⏩) to get it back. Currently, undo is limited to 100 steps. Undo also doesn’t work for everything; for instance, not for changes to the document layout what is really a LyX bug.

Cut/Paste/Copy Use Edit→Cut (toolbar button ✂), Edit→Copy (toolbar button ☝), and Edit→Paste (toolbar button ⚪) to cut, copy, and paste. Or automatically paste selected text (including selections from other programs) with the middle mouse button.

Find/Replace Use Edit→Find & Replace (toolbar button 🔍) to search. In the dialog, search with the Find Next button, and use the Replace button to replace a word you’ve found.⁴ If you like, you can specify whether to make the search case-sensitive, or to search for only complete words; you can also search backwards through the document.

Character Formatting You can emphasize text (which will by default print characters in italics), set it in bold face, or in Noun Style (usually small caps, used for people’s names) from the toggle buttons in the Edit→Text Style dialog (toolbar button ab).

Toolbar There are buttons on the toolbar (just below the menus) which allow you to do some of the more popular functions, such as Paste and Print.

Of course, you haven’t yet written enough to make most of these functions useful. As you write more, though, try undoing, pasting, etc.

2.1.3 WYSIWYM: Whitespace in LyX

One of the hardest things for new users to get used to is the way that LyX handles whitespace. As many times as you hit Return, you’ll only get one

⁴Close the window when you’re done or leave it open if you find it more convenient. Most dialog boxes in LyX can operate like this. Just be sure you have the right window focus when you’re trying to type in the main LyX window or a LyX dialog.
Chapter 2. Getting Started with LyX

blank line. As many times as you hit Space, you’ll only get one space. On a blank line, LyX won’t let you type even one space. The Tab key won’t move you forward one tab stop; in fact there are no tab stops! There’s no ruler at the top of the page to let you set tabs or margins, either.

Many word processors are based on the WYSIWYG principle: “What You See Is What You Get.” LyX, on the other hand, is based on the principle that “What You See Is What You Mean.” You type what you mean, and LyX will take care of typesetting it for you, so that the output looks nice. A Return grammatically separates paragraphs, and a Space grammatically separates words, so there is no reason to have several of them in a row; a Tab has no grammatical function at all, so LyX does not support it. Using LyX, you’ll spend more of your time worrying about the content of your document, and less time worrying about the format. See the Introduction for more information on the WYSIWYM concept.

LyX does have (many) ways to fine-tune the formatting of your document. After all, LyX might not typeset exactly what you mean. The User’s Guide has information about all that. It includes HFills and vertical space — which are more powerful and versatile than multiple spaces or blank lines — and ways to change font sizes, character styles, and paragraph alignments by hand. The idea, though, is that you can write your whole document, focusing on content, and just worry about that fine-tuning at the end. With standard word processors, you’ll be distracted by document formatting throughout the writing process.

2.2 Environments

Different parts of a document have different purposes; we call these parts environments. Most of a document is made up of regular text. Section titles (chapter, subsection, etc.) let the reader know that a new topic or subtopic will be discussed. Certain types of documents have special environments. A journal article will have an abstract and a title. A letter will have neither of these, but will probably have an environment that gives the writer’s address.

Environments are a major part of the “What You See Is What You Mean” philosophy of LyX. A given environment may require a certain font style, font size, indenting, line spacing, and more. This problem is aggravated, because the exact formatting for a given environment may change: one journal may use boldface, 18 point, centered type for section titles while another uses
2.2. ENVIRONMENTS

italicized, 15 point, left justified type; different languages may have different standards for indenting; and bibliography formats can vary widely. LyX lets you avoid learning all the different formatting styles.

The Environment choice box is located on the left end of the toolbar and looks like this: . It indicates which environment you’re currently writing in. While you were writing your first document, it said “Standard,” which is the default environment for text. Now you will put a number of environments in your new document so that you can see how they work.

2.2.1 Sections and Subsections

Type the word Introduction on the first line of your LyX file, and select Section in the Environment box. Be sure to use Section and not Section*, which will be covered below. LyX numbers the section “1” and typesets the section heading (title) in a larger font. Now hit Return. Note that the Environment box changes from “Section” back to “Standard”. Section headings, like most environments, are assumed to end when you type Return. Type the document introduction:

This is an introduction to my first LyX document.

Hit Return again, and select Section from the Environment box again. LyX writes a “2” and waits for you to type a title. Type “More Stuff”, and you’ll see that LyX again sets it as a section title.

It gets better. Go to the end of Section 1 again (after “my first LyX document”) and hit Return again, and select Section from the Environment box again. Again, LyX writes “2” and waits for you to type a title. Type About This Document. Section “More Stuff”, which used to be Section 2, has been automatically renumbered to Section 3! In true WYSIWYM fashion, you just need to identify the text that makes up the section titles, and LyX takes care of numbering the sections and typesetting them.

Hit Return to get back to the Standard environment, and type the following five lines:

---

You don’t have to select the line. If nothing is selected, LyX changes the paragraph you are currently in to the selected environment. Alternatively, you can change several paragraphs to a different environment by selecting them before picking an environment.
Sections and subsections are described below.

Section Description

Sections are bigger than subsections.

Subsection description

Subsections are smaller than sections.

Click on the second line and select **Subsection** from the **Environment** box. LyX numbers the subsection “2.1”, and typesets it in a font which is bigger than regular text but smaller than the section title. Change the fourth line **Subsection** environment as well. As you probably expected, LyX automatically numbered the section “2.2”. If you put yet another section before Section 2, Section 2 will be renumbered as Section 3, and the subsections will be renumbered to “3.1” and “3.2”.

Further levels of sectioning include **Subsubsection**, **Paragraph**, and **Subparagraph**. We’ll let you play with these on your own. You may notice that paragraph and subparagraph headings are not numbered by default, and that subparagraphs are indented; see the *User’s Guide* for an explanation and how to change this. **Chapter** headings are actually the highest level of sectioning, above **Sections**, but you’re only allowed to use them in certain types (text classes) of LyX documents (see Section 3.1).

Finally, you may want to have sections or subsections that are not numbered. There are environments for this as well. If you change one of your section headings to the **Section** environment (you may have to scroll down in the **Environment** box to find it), LyX will use the same font size for the heading as it uses for a regular section, but it won’t number that section. There are corresponding “starred” heading environments for **Subsection** and **Subsubsection**. Try changing some of your sections or subsections to the starred environments, and note how the other sections’ numbers are updated.

**Exercise**: Fix the section and subsection headings in **example_raw.lyx**.

### 2.2.2 Lists and sublists

LyX has several different environments for typesetting lists. The various list environments free you from hitting **Tab** a million times when writing an outline, or from renumbering a whole list when you want to add a point in the middle of the list. Different types of documents logically require different list environments:
2.2. ENVIRONMENTS

- A slide presentation might use the `Itemize` environment’s bulleted lists to describe different points.

- An outline would use the `Enumerate` environment’s numbered lists (and lettered sublists).

- A document describing several software packages could use the `Description` environment, where each item in the list begins with a bold-faced word.

- The `List` environment is a slightly different form of the `Description` environment.

Let’s write a list of reasons why LyX is better than other word processors. Somewhere in your document, type:

```
LyX is better than other word processors because:
```

and hit Return. Now select `Itemize` from the Environment box. LyX writes a “bullet” on the line. Type in your reasons:

```
Typesetting is done for you.
Math is WYSIWYG
Lists are very easy to create!
```

List environments, unlike headings, do not end when you type Return. Instead, LyX assumes you’re going on to the next item in the list. The above will therefore result in a three-item list. If you want more than one paragraph within one list item, one way is to use the Protected Break, which you get by typing Ctrl+Return. In order to get out of the list, you need to reselect the Standard environment (or just use the keybinding, Alt+P S).

You’ve got a beautiful itemized list. You might want to run \LaTeX to see how the list looks when printed out. But what if you wanted to number the reasons? Well, just select the whole list\(^6\) and choose `Enumerate` from the Environment box. Pow! As we mentioned, if you add or delete a list item, LyX will fix the numbering.

While the list is still selected, you can change to the other two list environments, `Description` and `List`, in order to see what they look like. For

\(^6\)LyX won’t let you select the first bullet unless you also select the paragraph before the list, which you probably don’t want to do. Similarly, you can’t select the actual number in a numbered section title. This is on purpose because the bullet or number depends on the document settings or text position, respectively.
those two environments, each list item is made up of a term, which is the item’s first word, followed by a definition, which is the rest of the paragraph (until you hit Return.) The term is either typeset in boldface (Description) or separated by a “Tab” \(^7\) (List) from the rest of the paragraph. If you want to have more than one word in the definition, then separate the words with Protected Blanks.

**Exercise:** Typeset the list in `example_raw.lyx`

You can nest lists within each other in all sorts of interesting ways. An obvious example would be writing outlines. Numbered and bulleted lists will have different numbering and bulleted schemes for sublists. See the *User’s Guide* for details on the different sorts of lists and for examples of nestings.

### 2.2.3 Other environments: Verses, Quotations, and more

There are two environments for setting quotations apart from surrounding text: Quote for short quotes and Quotation for longer ones. Computer code (the LyX-Code environment \(^8\)) is written in a typewriter font; this environment is the only place in LyX where you’re allowed to use multiple spaces to allow code indenting. You can even write poetry using the Verse style, using Return to separate stanzas, and Ctrl+Return to separate lines within a stanza. See the *User’s Guide* for more complete descriptions of all of the available LyX environments.

**Exercise:** Correctly typeset the Quote, LyX-Code, and Verse in `example_raw.lyx`

---

\(^7\)But a typesetter’s tab, which will change to fit the size of the largest term, not a pathetic, rigid, unchangeable typewriter Tab.

\(^8\)used in this Tutorial for the long typing examples
Chapter 3
Writing Documents

The previous chapter hopefully allowed you to get used to writing in LyX. It introduced you to the basic editing operations in LyX, as well as the powerful method of writing with environments. Most people who use LyX, though, will want to write documents: papers, articles, books, manuals, or letters. This chapter is meant to take you from simply writing text with LyX to writing a complete document. It will introduce you to text classes, which allow you to write different sorts of documents. It will then describe many of the additions that turn text into a document, such as titles, footnotes, cross references, bibliographies, and tables of contents.

3.1 Document Classes

Different sorts of documents should be typeset differently. For example, books are generally printed double-sided, while articles are single-sided. In addition, many documents contain special environments: letters contain some environments — such as the sender’s address and the signature — which do not make sense in a book or article. The LyX document class\(^\text{1}\) takes care of these large scale differences between different sorts of documents. This Tutorial, for example, was written in the Book document class. Document classes are another major part of the WYSIWYM philosophy; they tell LyX how to typeset the document, so you don’t need to know how.

Your document is probably being written in the Article document class.\(^\text{2}\)

\(^{1}\)for \LaTeX\ users: this is equivalent to the \LaTeX\ document class

\(^{2}\)That’s usually the default document class
Try changing to other document classes (using the Document › Settings dialog) to see how they are typeset differently. If you change your document to the Book document class and look at the Environment box, you’ll see that most of the allowed environments are the same. However, you can now use the Chapter environment. If you are ever unsure about which environments you can use in a given document class, just consult the Environment box.

Font sizes, one- or two-column printing, and page headings are just some of the ways journals’ typesettings differ from one another. As the Computer Age continues to mature, journals have begun accepting electronic submissions, creating \LaTeX “style files” so that authors can submit correctly typeset articles. \LaTeX{} is set up to support this as well. For example, \LaTeX{} supports typesetting (and extra environments) for the American Mathematics Society journals using the Article (AMS) document class.

Here is a very quick reference to some of the document classes. See the Special Document Classes section of the Additional Features manual for many more details.

<table>
<thead>
<tr>
<th>Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>article</td>
<td>one-sided, no chapters</td>
</tr>
<tr>
<td>article (AMS)</td>
<td>layout &amp; environments for American Math Society</td>
</tr>
<tr>
<td>report</td>
<td>longer than article, two-sided</td>
</tr>
<tr>
<td>book</td>
<td>report + front and back matter</td>
</tr>
<tr>
<td>presentation</td>
<td>transparencies</td>
</tr>
<tr>
<td>letter</td>
<td>lots of extra environments for address, signature...</td>
</tr>
</tbody>
</table>

3.2 Templates: Writing a Letter

One way to write a letter would be to open a new file, and choose a Letter class in the Document › Settings dialog. While this is the most obvious way to write a letter, it seems like extra work. Every time you write a business letter, you want to have your address, the address you’re sending to, a body, a signature, etc. \LaTeX{} therefore has a template for letters, which contains a sample letter; once you have a template, you can just replace a couple parts of the letter with your text each time you write a letter.

Open a new file with File › New from Template. Select letter.1yx as the template. Save and print the file to see how the various environments are typeset.
When you look at the Environment box, you’ll see several environments, like the My Address environment, which don’t even exist in most other document classes. Others, like Quote and Description, are familiar. You can play around for a while to figure out how the various environments work. You’ll notice for example that the Signature environment has the word “Signature:" in red before the actual text of the signature. This word doesn’t show up in the actual letter, as you’ll see if you view/export the file. It’s just there to let you know where the signature goes. Also, note that it doesn’t matter where in the file the Signature line is placed. Remember, \texttt{LYX} is WYSIWYM; you can put the Signature environment anywhere you want, but \texttt{LYX} knows that in the printout, the signature should be at the end.

A template is just a regular \texttt{LYX} file. This means you can fill in your address and signature and save the file as a new template. From now on, any time you want to write a letter, you can use the new template to save time. We don’t have to suggest an actual “exercise” here; just write a letter to someone!\footnote{One warning, if you’re writing from a template. If you erase all of the text in an environment — for example, if you erase the whole My Address field so that you can replace it with your own — and then you move the cursor without writing any text, the environment may disappear. This is because most environments cannot exist without any text in them. Just reselect the environment from the Environment box to get it back.}

Templates can be a huge time-saver, and we urge you to use them whenever possible. In addition, they can help a person learn how to use some of the fancier document classes. Finally, they may be useful for a person who is configuring \texttt{LYX} for a bunch of less computer-aware users. When they’re first learning \texttt{LYX}, it will be much less intimidating if they have a letter template customized for their company, for example.

3.3 Document Titles

\texttt{LYX} (like \texttt{LATEX}) considers the title — which may contain the actual title, the author, the date, and even an abstract of a paper — to be a separate part of the document.

Go back to your \texttt{LYX} document and make sure it’s using the Article document class.\footnote{You should not be using the letter any more, since the Letter document class doesn’t allow titles.} Type a title on the first line, and change the line to the Title environment. On the next line, type your name and change it to the Author environment.
environment. On the next line, write the date in the \texttt{Date} environment. Type a paragraph or two summarizing your document using the \texttt{Abstract} environment. Notice how the title is presented when it’s printed out. If you changed the document format to Book, you’ll get a separate title page, like the first page of this tutorial.

\textbf{Exercise}: Fix the title, date, and author in \texttt{example_raw.lyx}

\section{3.4 Labels and Cross-References}

You can label section headings, list items, formulas, footnotes, and floats\textsuperscript{5} in your document. Once you do so, you can refer to this section in other parts of the document, using cross-references. You can refer either to the section’s number, or to the page that the section appears on. As with section numbering, \LaTeX{} also takes care about cross-reference numbering for you. Automatic labels and cross-references are one of the best advantages of \LaTeX{} (and \TeX{}) over conventional word processors.

\subsection{3.4.1 Your first label}

Go to our second section, whose title is “About This Document”. Click at the end of the section title line, and select \texttt{Insert} > \texttt{Label} or the toolbar button \texttt{Label}. A dialog asks you for a label name, and gives you a suggestion. When you click on \texttt{OK}, the label name will be placed in a box next to the section title.

By the way, you could have put the label right anywhere within the section as well; section references will refer to the last section or subsection whose heading comes before the label. However, putting it on the same line as the section title (or, perhaps, on the first line of the section’s text) ensures that page references will reference the beginning of the section.

So far you haven’t done anything — the DVI output will look exactly the same, since labels don’t show up in the printed document. However, now that you have added a label, you can refer to that label with cross-references. We’ll do that next.

\textsuperscript{5}Floats are explained in the \textit{User’s Guide} and the \textit{Embedded Objects} manual.
3.4. LABELS AND CROSS-REFERENCES

3.4.2 Your first cross-reference

Place the cursor somewhere in section 2 of your document. Type

If you want to know more about this document, then see section, which can be found on page.

Now set the cursor after the word “section” and choose Insert > Cross Reference or the toolbar button ⊿. The Cross-reference dialog pops up. It shows a list of the possible labels you can reference. At the moment, there should be only one, “sec:About-This-Document”. Select it (it may be selected by default), and click Apply. Now put the cursor after the word “page”, and change the reference format to use the page number then click Apply. (To be really correct, you should put a Protected Blank in between the word “Section” and the reference. Same for the page reference.)

Alternatively to that method, you can right-click on a label and use in the appearing context menu Copy as Reference. The cross-reference to this label is now in the clipboard and can be copied to the actual cursor position via the menu Edit > Paste (shortcut Ctrl+V).

\LaTeX{} puts the references in a box right where the cursor was. In the printed document, this reference marker will be replaced with either the page or section number (depending on what you selected in the Cross-reference dialog). View your document as DVI, and you’ll see that on the last page we refer to “Section 2” and “Page 1” (or whatever page Section 2’s title is on).

Conveniently, a cross-reference acts as a hyperlink when you are editing a document in \LaTeX{}; clicking on it will pop up the Cross-Reference dialog, clicking Go to Label will move the cursor to the referenced label.

3.4.3 More fun with labels

We told you that \LaTeX{} takes care about numbering cross-references; now you can test that. Add a new section before Section 2. Update the DVI view, and — voilà! — the section cross reference changed to “3”! Change the section “About this Document” to a subsection, and the cross-reference will reference Subsection 2.1 instead of Section 3. The page reference won’t change unless you add a whole page of text before the label, of course.

If you want some more practice with labels, then try putting a new label where your first cross-reference was, and refer to that label from elsewhere.
in the document. If you’ll be inserting cross-references often, it may be convenient to leave the Cross-reference dialog open.

If you want to make sure that the cross-referencing gets the pages right even for larger documents, Copy a couple pages of text from the User’s Guide to the clipboard, and Paste the stolen text into your document.\footnote{By the way, copying a chapter title may cause an error, because chapters aren’t allowed in the article class, see section 3.1. If this happens, just delete the chapter title.}

Exercise: Fix the references in example\_raw.lyx

### 3.5 Footnotes and Margin Notes

Footnotes can be added using the toolbar button \footnotemark\footnoterefname or the menu Insert\hypertarget{Footnote}. Click at the end of the word “\texttt{LYX}” somewhere in your document and click the \footnotemark button. A footnote box appears where you can enter the text of the footnote. \texttt{LYX} should place the cursor at the beginning of the footnote box. Type

\begin{quote}
\texttt{LYX} is a typesetting word processor.
\end{quote}

Now click on the button labelled “foot”. The footnote box is closed, leaving the button showing where the footnote marker will be in the printed text; this is called “folding” the footnote. You can unfold the footnote at any time and re-edit its text by clicking again on the “foot” button.

You may wonder why the footnote button is a word instead of a number. The answer is that \texttt{LYX} takes care about the footnote numbering for you in the printed text. You can see this yourself by looking at the DVI file (or printout). If you add other footnotes, \texttt{LYX} will renumber the footnotes. Since \texttt{LYX} (well, \texttt{LATeX}, actually) takes care of the footnote numbering, there’s really no need to put the numbers in the \texttt{LYX} file.

A footnote can be cut and pasted like normal text. Go ahead; try it! All you need to do is select the footnote button\footnote{It may be easier to select it using the keyboard. You might accidentally open the footnote if you’re trying to select the marker itself with the mouse.} and Cut and Paste it. In addition, you can change regular text to a footnote, by selecting it and hitting the \footnotemark button; change a footnote to regular text by hitting the Backspace key when the cursor is in the first position of a footnote, or by hitting the Delete key when the cursor is in the very last position of the footnote, respectively.
3.6. BIBLIOGRAPHIES

Margin notes can be added using the menu Insert\ Marginal Note or the toolbar button Marginal Note. Margin notes are like footnotes, except that:

- the on-screen boxes say “margin” instead of “foot”
- the notes will be placed in the margin, instead of below the text
- margin notes are not numbered

Change your LyX footnote back to text, then select and change it to a margin note. Run \LaTeX{} again to see what the margin note looks like.

**Exercise:** Fix the footnote in example_raw.lyx

3.6 Bibliographies

Bibliographies (at least in the exact sciences) are similar to cross references. The bibliography contains a list of references at the end of the document, and they can be referenced from within the document. Like section titles, LyX and \LaTeX{} make your job easier by automatically numbering the bibliography items and changing citations when the item numbers change.

Go to the end of the document and switch to the Bibliography environment. Now, each paragraph you type will be a reference. Type “The Lyx Tutorial, by the LyX Documentation Team” as your first reference. Note that LyX automatically puts a number in a box before each reference. Click on the boxed reference number, and the Bibliography item dialog box appears. The Key is to refer to this reference within the LyX document, the Label appears in output. When no Label is set (default), you will see the number of the bibliography in the output. Change now the Key field to “lyxtutorial” to make it easy to remember.

Now pick somewhere in your document that you would like to insert a reference. Do so with Insert\ Citation or the toolbar button Citation. A Citation dialog appears. The right panel in this dialog lists all the bibliography entries, and this field allows you to choose which bibliography item you want to cite. Select “lyxtutorial” (right now, that’s the only item in the bibliography), then use the Add button in the center to insert it. (You can have multiple citations in the same place by transferring a number of keys this way.) Now view your file as DVI, and you’ll see that the citation appears in brackets in the text, referring to the bibliography at the end of the document.
The Text after field in the Citation dialog will put a remark (such as a reference to a page or chapter within the referenced book or article) in the brackets after the reference. If you want the references to have labels instead of numbers in the printed output (for example, some journals would use “[Smi95]” to refer to a paper written by Smith in 1995), use the Label field in the Bibliography item dialog. As usual, see the User’s Guide for details.

**Exercise:** Fix the bibliography and citation in example_raw.lyx

### 3.7 Table of Contents

You may want to put a table of contents at the beginning of your document. LyX makes this very easy to do. Just hit Return after your document title and before your first section title and choose Insert > List / TOC > Table of Contents. The words “Table of Contents” will appear in a button on the first line of the document.

This may not appear to be very useful. However, if you look at your DVI file, you will see that a table of contents has been generated, listing the various sections and subsections in your document. As usual, if you reorder sections or create new ones, you will see those changes in the DVI file when you update it.

The table of contents is not printed in the on-screen version of the document to keep the overview in your file. But you can display the table of contents in a separate window by clicking on the table of contents button, or by using Document > Outline or the toolbar button 📊. This menu will work even if you don’t have a table of contents inset in your document. This is a very useful tool where you can move around your document parts. Clicking on a (sub)section title in the Outline window will highlight that line and move the display (in the LyX editing window) to that place in the document. You can also use the arrow keys to move up and down in the table of contents. You may therefore find it convenient to leave this window open throughout editing sessions. You can get similar functionality from the Navigate menu, though, where the table of contents appears automatically.

To get rid of the Table of Contents, you can delete the table of contents button just like any other text.

**Exercise:** Fix the table of contents in example_raw.lyx
Chapter 4

Using Math

\LaTeX is used by many scientists because it outputs great looking equations, avoiding the control characters used by word processors and their equation editors. Many of these scientists are frustrated, however, because writing equations in \LaTeX is more like programming than writing. Happily, \LyX has WYSIWYM support for equations. If you are used to \LaTeX, you’ll find that all of the usual \LaTeX math commands can be typed in normally, but they will show up in a WYSIWYM fashion. If, on the other hand, you’ve never written in \LaTeX, then the Math Panel will allow you to write professional-looking math quickly and easily.

4.1 Math Mode

Somewhere in your \LyX document, type:

I like what Einstein said, \( E=mc^2 \), because it’s so simple.

Now, that equation doesn’t look very good in \LyX and in the output; there’s no space between the letters and the equals sign, and you’d like to write an actual superscript for the “2”. That bad typesetting happened because we didn’t tell \LyX that we were writing a mathematical expression, so it typeset the equation like regular old text.

Instead, we create a formula that will get typeset properly. In order to create a formula, just click the toolbar button \( \frac{a+b}{c} \) or use the menu Insert\( \rightarrow \) Math\( \rightarrow \) Inline Formula. \LyX will insert a little blue square, which is an empty
math formula. Now just type \textit{E}=mc^2 again. The expression is typed in blue, and the blue square disappears as soon as the formula is not empty. Now type \texttt{Esc} to leave the equation. The purple markers disappear, leaving the cursor to the right of the expression, and now if you type something, it will be regular text.

Run \LaTeX{} and look at the output. Notice that the expression was typeset nicely, with spaces between the letters and the equals sign, and a superscript “2”. Letters in math mode are assumed to be variables, and come out in italics. Numbers are just numbers.

This math editor is another example of the WYSIWYM philosophy. In \LaTeX{}, you write a mathematical expression using text and commands like \texttt{\sqrt{}}; this can be frustrating, because you can’t see what an expression looks like until you \LaTeX{} the file, and may have to spend time to find e.g. missing brackets. \LYX{} doesn’t attempt to get the expression to look perfect (WYSIWYG), but it gives you an extremely good idea of what the expression will look like. \LaTeX{} then takes care of the professional typesetting.

### 4.2 Navigating an Equation

Now let’s change \textit{E}=mc^2 to \textit{E}=1+mc^2. Use the arrow keys to move the cursor into the expression. Note that when you enter the expression, the purple markers appear to let you know you’re editing math. Now you can use \texttt{Left} and \texttt{Right} to move the cursor past the equals sign, and just type “1+”. Again, you can use the arrow keys or \texttt{Esc} to leave the formula.

Other than the special keys described below, typing in math mode is like editing regular text. Use \texttt{Delete} (or \texttt{Backspace}) to delete things. Select text either with the arrow keys or with the mouse. \texttt{Edit}→\texttt{Undo} works in math mode as well as cut and paste. One thing to be careful of: If you are left or right outside a formula and you press \texttt{Delete} or \texttt{Backspace}, respectively, you delete the whole formula. Luckily, you can just use \texttt{Undo} to get it back.

What if you want to change \textit{E}=mc^2 to \textit{E}=mc^{2.5}+1? Again, you can use the mouse to click in the right place. However, you can also use the arrow keys. If the cursor is just after the “c” but before the “2”, then press \texttt{Up} and the cursor is moved to the level of the superscript, just before the “2”. Add the “.5”. Now, hitting \texttt{Down} will move the cursor back to the regular level. When you hit \texttt{Space} instead of \texttt{Down}, the cursor will be placed after the superscript (so that you can then type the “+1”).
### 4.3 Exponents and Indices

An exponent can be entered from the Math Toolbar (see below), but it’s actually simpler just to type the caret key, “^”. LYX will place another blue rectangle in the superscript, so that whatever you write next will be superscripted, and in a smaller font size. Everything you type until you hit a Space (or Esc to exit the formula entirely) will be in the superscript.

Writing a subscript (index) is just as easy — start one by typing the underscore key, “_”. You can subscript and superscript both subscripts and superscripts like this: \( A_{a_0+b^2} + C^{a_0+b^2} \).

**Exercise:** Put equation 1 of example_raw.lyx into math mode.

### 4.4 The Math Toolbar

The Math Toolbar is a convenient way to enter symbols or to perform complicated formula operations. Many of these operations can be accomplished from the keyboard or the Edit > Math or Insert > Math menus. However, we’re going to concentrate on using the Math Toolbars, just to let you know what’s out there; you can learn keyboard shortcuts later, from other manuals.

The math toolbar is shown when the cursor is in a formula and can also be turned on manually in the menu View > Toolbars. When you click there on “Math” the toolbar will be shown permanently at the bottom; this state is visualized in the Toolbars menu with a checkmark. When you click in this state again on “Math” in the Toolbars menu, the math toolbar is only shown when the cursor is within a formula; this state is visualized by the renaming of the menu entry from “Math” to “Math (auto)”.

#### 4.4.1 Greek and symbols

The Math Toolbar which allow you to choose from a large array of symbols used in math: various arrows, relations, operators, and sums and integrals. Note that subscripting and superscripting allow you to put lower and upper limits on sums and integrals.

“Nothing you can do that can’t be done... All you need is ♥.”
4.4.2 Square roots, accents, and delimiters

To type a square root, just click on the button $\sqrt{}$. The square root appears, and the cursor is in a new insertion point inside the square root. You can type variables, numbers, other square roots, fractions, whatever you want. LyX will automatically resize the square root to fit what’s inside.

Accenting a character ($\overline{v}$) or group of characters ($\overline{a + b}$) is done the same way. Decorations are available from the toolbar via the button $\hat{}$. Click on a decoration, and LyX will insert that decoration with an insertion point under (or over) it. Just type what you want in the insertion point. There are two sets of decorations: those that resize with the text you type, and those that have fixed size, and are most appropriate for a single letter.

Delimiters such as parentheses, brackets, and braces work similarly, but are a bit more complicated. Hit the delimiter button [ ] to pop up the Delimiter dialog. Your current selection of delimiters is displayed in a box. It’s a pair of parentheses by default, but you can choose a pair of braces, a brace and a parenthesis, or choose the empty square to have something like “$a = \langle 7 \rangle$” (the empty delimiter is displayed as a broken line in LyX, but won’t show up in the output).

If you’re lazy, you can type actual parentheses in math mode, rather than using the Delimiter dialog. However, those parentheses will be the same size as regular text, which will look bad if you have a big fraction or matrix inside the parentheses. So better use in this case one of the three delimiter buttons that insert directly e.g. a ( ) pair.

You can also put delimiters or a square root sign or a decoration on already existing formula parts. Select the portion of the formula that you want to adjust, and then click on the button you want from the Math Toolbar. Try using this to change Newton’s second law from scalar to vector form ($f = ma$ to $\overrightarrow{f} = m \overrightarrow{a}$). Once you’ve learned about matrices, this is how you’ll put parentheses or brackets around them.

4.4.3 Fractions

To create a fraction, click on the fraction button $\frac{a}{b}$ in the Math Toolbar. LyX writes two insertion points in a fraction. As you would expect, you can use arrow keys or the mouse to move around a fraction. Click on the top square and type “1”. Now hit Down and type “2”. You’ve made a fraction!
4.4. **THE MATH TOOLBAR**

Of course you can type anything within each of the two boxes: variables with exponents, square roots, other fractions, whatever.

**Exercise:** Put equation 2 of `example_raw.lyx` into math mode.

### 4.4.4 **\TeX** mode: Limits, log, sin and others

Because letters in math mode are considered to be variables, if you type “sin” in math mode, LyX thinks you are typing the product of the three variables $s$, $i$, and $n$. The three letters will be typeset in italics, when what you really wanted was the word “sin” typeset in Roman. In addition, LyX won’t put a space between the word “sin” and the “x” (pressing Space will exit the formula). So how do you get “$\sin(x)$” instead of “$sin(x)$”?

Click on the **Math Toolbar** button and then on “sin” in the appearing function list. The word “sin” is displayed in LyX in black, and set in upright roman type. The whole word is treated as one symbol, so if you type Backspace, it will delete the whole word. Now type “(x)”, which will be written in blue italics, like you expect in a formula. In the output, the expression will be correctly typeset. Try it out.

The function list include other trigonometric functions and their inverses, hyperbolic functions, logarithms, limits, and quite a few others. These functions can take subscripts and superscripts, important for typing “$\cos^2 \theta$” or “$\lim_{n \to \infty}$”.

**Exercise:** Put equation 3 of `example_raw.lyx` into math mode.

### 4.4.5 **Matrices**

Click on the matrix button in the **Math Toolbar**. The appearing dialog allows you to choose how many rows and columns you want in your matrix. Choose 2 rows and 3 columns and hit OK. LyX prints 6 insertion points in a $2 \times 3$ matrix. As usual, you can put any sort of formula expression (a square root, another matrix, etc.) in each insertion point. You can also leave some of the insertion points empty if you want.

Tab can be used to move horizontally between the columns of a matrix. Alternatively, you can use the arrow keys to move around - hitting Right at the end of one box will move to the next box, Down will move to the next row, etc.
If you need to change the number of rows and columns, use the menu Edit → Rows & Columns or the math toolbar buttons \( \frac{\text{ɪ}}{\text{m}} \), \( \frac{\text{x}}{\text{m}} \), \( \frac{\text{ɪ}}{\text{m}} \), \( \frac{\text{x}}{\text{m}} \).

See the User’s Guide for information on how to change the horizontal alignment of each column, and how to change the vertical position of the whole matrix. Note that if you want to write a table containing text, you should use \LaTeX{}’s wonderful table support, rather than trying to write text in a matrix.

### 4.4.6 Display mode

All of the expressions we have written so far have been on the same line as the text that came before and after them, otherwise known as inline expressions. This is fine for short, simple expressions, but if you want to write larger ones, or if you want your expressions to stand out from the text, you need to write them in display mode. In addition, only displayed expressions can be labeled and numbered (see the User’s Guide), and multi-line equations must be in display mode.

Click on the display button \( \frac{\text{ɪ}}{\text{m}} \) in the Math Toolbar, which represents a couple lines of text before and after a centered blue box. \LaTeX{} inserts a formula, but the insertion point is on a new line, and it’s centered within that line. Now type an expression and run \LaTeX{} to see how it looks. The display button is actually a toggle; use it now to change a couple of your expressions to display mode and back.

Display mode has a couple differences from inline mode:

- The default font is larger for a few symbols, like \( \sum \) and \( \int \)
- Subscripts and superscripts for limits and sums (but not integrals) are written under rather than next to the symbols
- Text is centered

Other than these differences, though, displayed expressions and inline expressions are very similar.

One final note about the way displayed formulas are typeset: Be careful about whether you’re putting your equation into a new paragraph or not. If your formula is in the middle of a sentence or paragraph, then don’t press
Return. Doing so will cause the text after the formula to start a new paragraph. That text will therefore eventually be indented, depending on your document paragraph settings, which is probably not what you want.

**Exercise:** Put the various equations in `example_raw.lyx` into display mode, and see how they’re typeset differently.

**Exercise:** Using various tools you’ve learned in this section, you should be able to write an equation like\(^1\):

\[
f(x) = \begin{cases} 
  \log_8 x & x > 0 \\
  0 & x = 0 \\
  \sum_{i=1}^{5} \alpha_i + \sqrt{-\frac{1}{x}} & x < 0 
\end{cases}
\]

# 4.5 More Math Stuff

\LaTeX{}’s math editor can do plenty more. By now, you’re familiar with the basics, so we refer to the *User’s Guide* for tips on how to:

- Labeling and numbering expressions
- Multi-line equations
- Change typefaces, e.g. to write bold-face text in an expression.
- Fine-tune font sizes and spacing within an expression. (Don’t worry about this until your final draft!)
- Write macros. These are very powerful, because you just define them once at the top of the document, and then you can use them throughout the document.
- Do lots of other things that can’t be mentioned in this Tutorial.

---

\(^1\)After you’ve done it the hard way, give Insert > Math > Cases Environment a try.
Chapter 5

Miscellaneous

5.1 Other major \LaTeX{} Features

We haven’t gone through all the possible commands in \LaTeX{}, and we aren’t planning on it. As usual, see the User’s Guide and the Embedded Objects manual for more information. We’ll just mention a couple more major things \LaTeX{} can do:

- \LaTeX{} has WYSIWYM support for tables. Use the Insert $\triangleright$ Table (toolbar button $\leftarrow$) to get a table. Click on the table with the right button to get a Table Settings dialog box which allows extensive table editing.

- \LaTeX{} also supports including pictures in any format within documents. (You guessed it: Insert $\triangleright$ Graphics (toolbar button $\leftarrow$). Then browse for the figure file, rotate or scale it, etc.) Tables and figures can have captions, and \LaTeX{} will automatically generate lists of figures and/or tables.

- \LaTeX{} is heavily configurable. Everything from how the \LaTeX{} window looks to how the output comes out can be configured in a number of ways. Much configuration is done through Tools $\triangleright$ Preferences. For more information on this, check out Help $\triangleright$ Customization.

- \LaTeX{} is being developed by a team of programmers on five continents. Therefore, \LaTeX{} has better support for non-English languages (such as Dutch, German, French, Greek, Czech, Turkish, \ldots) than many word
processors. Even the right-to-left languages Arabic, Farsi, and Hebrew and the Asian languages Chinese Japanese, and Korean are supported. You can write documents in other languages and you can also configure LYX to show its menus and error messages in other languages.

- The LYX menus feature keybindings. This means that you can do File → Open by pressing Alt+F followed by O or by using the binding which is shown next to it in the menu (Ctrl+O by default). Keybindings are also configurable. For information on this, check out Help → Customization.

- LYX can read \LaTeX{} documents. See section 5.2.2.

- Spellchecking, thesaurus, and word count facilities are available.

- Generation of indexes and nomenclatures/glossaries is supported.

## 5.2 LYX for \LaTeX{} Users

If you don’t know anything about \LaTeX{}, you don’t have to read this section. Actually, you might want to learn about \LaTeX{}, and then read this chapter. However, some who begin to use LYX will be familiar with \LaTeX{}. If you are such a person, you may be wondering if LYX can really do everything \LaTeX{} can do. The short answer is that LYX can do pretty much everything \LaTeX{} can do in one form or another, and it definitely simplifies most parts of writing a \LaTeX{} document.

Because this is just a tutorial, we are only going to mention things that new LYX users will most likely be interested in. In the interests of keeping the Tutorial short, we will give only minimal information here. The Additional Features and the Embedded Objects manual have a great deal of information on differences between LYX and \LaTeX{}, and how to do various \LaTeX{} tricks in LYX.

### 5.2.1 \TeX{} Mode

Anything that you enter in \TeX{} mode will be passed straight to \LaTeX{}, and will be displayed in red on the screen. You can use \TeX{} commands in LYX by choosing Insert → \TeX{} Code (toolbar button \TeX{}). This creates a box where everything within it is passed straight to \LaTeX{}. 
In a math formula, \TeX mode is handled a bit differently. \TeX mode is there entered by typing a backslash. The backslash is not written out, but anything you type afterwards will be in red. You exit \TeX mode by typing \texttt{Space} or some other non-alphabetic character, like a number, underscore, caret, or parenthesis. Once you exit \TeX mode, if \LyX knows the \TeX command you’ve typed in, it will convert it to WYSIWYM. So if you type “\texttt{\gamma}” in a formula and then press \texttt{Space}, \LyX will change the red “gamma” to a blue “γ”. This will work for almost all, non-complicated math macros. This may be faster than using the Math Toolbar, and will be especially convenient for experienced \LaTeX users.

As a special case, if you type a brace in \TeX mode, then the beginning and ending braces will be inserted in red, then take you \textit{out} of \TeX mode and place the cursor between the braces. This makes it more convenient to type commands that \LyX doesn’t know which take an argument.

\LyX can’t do absolutely everything that \LaTeX can do. Some fancy functions are not supported at all, while some work but aren’t WYSIWYM. \TeX mode allows users to get the full flexibility of \LaTeX, while having all the convenient features of \LyX, like WYSIWYM math, tables, and editing. \LyX could never support every \LaTeX package. However, by typing \texttt{\usepackage{foo}} in the preamble (see section 5.2.4.2), you can use any package you want — although you won’t have WYSIWYM support for that package’s features.

5.2.2 Importing \LaTeX Documents — tex2lyx

You can import a \LaTeX file into \LyX by using the File $\triangleright$ Import $\triangleright$ \LaTeX (plain) menu in \LyX. This will call the program \texttt{tex2lyx} which will create a file \texttt{foo.lyx} from the file \texttt{foo.tex} and then open that file. If the translation doesn’t work, you can try calling \texttt{tex2lyx} from the command line, possibly using fancier options.

\texttt{tex2lyx} will translate most legal \LaTeX, but not everything. It will leave things it doesn’t understand in \TeX mode, so after translating a file with \texttt{tex2lyx}, you can look for red text and hand-edit it to look right.

\texttt{tex2lyx} has its own manpage. Read it to find out about which \LaTeX commands and environments aren’t supported, bugs (and how to get around them), and how to use the various options.
5.2.3 Converting \LaTeX Documents to \LaTeX

You might wish to convert a \LaTeX Document to a \LaTeX file. For example, a co-worker or co-author who doesn’t have \LaTeX might want to read it. Select File>Export>\LaTeX. This will create a file whatever.tex from the whatever.lyx file you are editing. \LaTeX always creates temporary \LaTeX files when viewing or printing files.

5.2.4 \LaTeX Preamble

5.2.4.1 Document Class

The Document>Settings dialog takes care of many of the options that you would input in a \documentclass command. Change the class, default font size and paper size here. Put any extra options to the \documentclass command in the Extra Options area.

5.2.4.2 Other Preamble Matter

If you have special commands to put in the preamble of a \LaTeX file, you can use them in a \LaTeX document as well. Select Document>Settings>\LaTeX Preamble and type in the dialog window (or from the document settings dialog, depending on the frontend). Anything you type will (like with \TeX mode) be sent directly to \LaTeX.

5.2.5 Bib\TeX

\LaTeX has support for Bib\TeX, which allows you to build databases of bibliographical references to be used in multiple documents. Select Insert>List / TOC>Bib\TeX Bibliography to include a Bib\TeX file. In the Database field you load Bib\TeX files, in the Style field you can load Bib\TeX style files.

After you’ve done this, you can use citations from any bibliographies you’re including with Insert>Citation (see section 3.6). \LaTeX will take care of running Bib\TeX. The box in the Citation dialog will show a list of all the references in your Bib\TeX file.
5.3 Errors!

Sometimes when you \LaTeX{} a document, there will be errors, things that \LaTeX{} or \LAM{X} can’t understand. When this happens, \LaTeX{} will open a \LaTeX{} Errors dialog. Clicking on individual errors in this dialog will take you to the place in the \LaTeX{} document where the error occurs and also display the detailed \LaTeX{} error message.
## Contents

1. **Getting Started** 42  
   1.1. What is LyX? 42  
   1.2. How LyX Looks 42  
   1.3. HELP 43  
   1.4. Basic LyX Setup 43  
   1.5. \LaTeX{} Setup 43  

2. **How to work with LyX** 43  
   2.1. Basic File Operations 43  
   2.2. Basic Editing Features 44  
   2.3. Undo and Redo 45  
   2.4. Mouse Operations 45  
   2.5. Navigating 46  
   2.6. Input / Word Completion 46  
   2.7. Basic Key Bindings 47  

3. **LyX Basics** 49  
   3.1. Document Types 49  
      3.1.1. Introduction 49  
      3.1.2. Document Classes 49  
         3.1.2.1. Overview 49  
         3.1.2.2. Modules 51  
         3.1.2.3. Properties 51  
      3.1.3. Document Layout 52  
      3.1.4. Paper Size and Orientation 52  
      3.1.5. Margins 53  
      3.1.6. Important Note 53  
   3.2. Paragraph Indentation and Separation 53  
      3.2.1. Introduction 53  
      3.2.2. Paragraph Separation 54  
      3.2.3. Fine-Tuning 54  
      3.2.4. Line Spacing 54  
   3.3. Paragraph Environments 55  
      3.3.1. Overview 55  
      3.3.2. Standard 56  
      3.3.3. Document Title 56


<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.4. Headings</td>
<td>56</td>
</tr>
<tr>
<td>3.3.4.1. Numbered Headings</td>
<td>56</td>
</tr>
<tr>
<td>3.3.4.2. Unnumbered Headings</td>
<td>57</td>
</tr>
<tr>
<td>3.3.4.3. Changing the Numbering</td>
<td>58</td>
</tr>
<tr>
<td>3.3.4.4. Short Titles</td>
<td>58</td>
</tr>
<tr>
<td>3.3.4.5. Special Information</td>
<td>58</td>
</tr>
<tr>
<td>3.3.5. Quotes and Poetry line spacing</td>
<td>58</td>
</tr>
<tr>
<td>3.3.5.1. Quote and Quotation</td>
<td>59</td>
</tr>
<tr>
<td>3.3.5.2. Verse</td>
<td>59</td>
</tr>
<tr>
<td>3.3.6. Lists</td>
<td>60</td>
</tr>
<tr>
<td>3.3.6.1. General Features</td>
<td>60</td>
</tr>
<tr>
<td>3.3.6.2. Itemize</td>
<td>60</td>
</tr>
<tr>
<td>3.3.6.3. Enumerate</td>
<td>61</td>
</tr>
<tr>
<td>3.3.6.4. Description</td>
<td>62</td>
</tr>
<tr>
<td>3.3.6.5. The \LaTeX\ List</td>
<td>63</td>
</tr>
<tr>
<td>3.3.7. Letters</td>
<td>64</td>
</tr>
<tr>
<td>3.3.7.1. Address and Right Address: An Overview</td>
<td>64</td>
</tr>
<tr>
<td>3.3.7.2. Usage</td>
<td>64</td>
</tr>
<tr>
<td>3.3.8. Academic Writing</td>
<td>65</td>
</tr>
<tr>
<td>3.3.8.1. Abstract</td>
<td>65</td>
</tr>
<tr>
<td>3.3.8.2. Bibliography</td>
<td>66</td>
</tr>
<tr>
<td>3.3.9. \LaTeX-Code</td>
<td>66</td>
</tr>
<tr>
<td>3.4. Nesting Environments</td>
<td>67</td>
</tr>
<tr>
<td>3.4.1. Introduction</td>
<td>67</td>
</tr>
<tr>
<td>3.4.2. What You Can and Can’t Nest</td>
<td>68</td>
</tr>
<tr>
<td>3.4.3. Nesting Other Things: Tables, Math, Floats, etc.</td>
<td>69</td>
</tr>
<tr>
<td>3.4.4. Usage and General Features</td>
<td>71</td>
</tr>
<tr>
<td>3.4.5. Some Examples</td>
<td>71</td>
</tr>
<tr>
<td>3.4.5.1. Example 1: The Six-fold Way and Mixed Nesting</td>
<td>71</td>
</tr>
<tr>
<td>3.4.5.2. Example 2: Inheritance</td>
<td>72</td>
</tr>
<tr>
<td>3.4.5.3. Example #3: Labels, Levels and other list environments</td>
<td>72</td>
</tr>
<tr>
<td>3.4.5.4. Example 4: Going Bonkers</td>
<td>74</td>
</tr>
<tr>
<td>3.5. Spacing, pagination and line breaks</td>
<td>75</td>
</tr>
<tr>
<td>3.5.1. Protected Space</td>
<td>75</td>
</tr>
<tr>
<td>3.5.2. Horizontal Space</td>
<td>75</td>
</tr>
<tr>
<td>3.5.2.1. Inter-word Space</td>
<td>76</td>
</tr>
<tr>
<td>3.5.2.2. Thin Space</td>
<td>76</td>
</tr>
<tr>
<td>3.5.2.3. More Spaces</td>
<td>76</td>
</tr>
<tr>
<td>3.5.2.4. Horizontal Fills</td>
<td>76</td>
</tr>
<tr>
<td>3.5.2.5. Phantom Space</td>
<td>77</td>
</tr>
<tr>
<td>3.5.3. Vertical Space</td>
<td>78</td>
</tr>
<tr>
<td>3.5.4. Paragraph Alignment</td>
<td>78</td>
</tr>
<tr>
<td>3.5.5. Forced Page Breaks</td>
<td>79</td>
</tr>
<tr>
<td>3.5.5.1. Clear Page Breaks</td>
<td>79</td>
</tr>
</tbody>
</table>
3.5.6. Forced Line Breaks ........................................ 79
3.5.7. Horizontal Lines ......................................... 80
3.6. Characters and Symbols .................................... 80
3.7. Fonts and Text Styles ...................................... 80
  3.7.1. Font Types .............................................. 80
  3.7.2. Document Font and Font size ......................... 81
  3.7.3. Using Different Character Styles ..................... 82
  3.7.4. Fine-Tuning with the Text Style dialog .............. 83
3.8. Printing and Previewing .................................... 86
  3.8.1. Overview ................................................ 86
  3.8.2. Output file formats .................................... 86
    3.8.2.1. ASCII ........................................... 86
    3.8.2.2. \LaTeX ......................................... 86
    3.8.2.3. DVI ........................................... 87
    3.8.2.4. PostScript ...................................... 87
    3.8.2.5. PDF ........................................... 87
  3.8.3. Previewing ............................................. 88
  3.8.4. Printing the File from within \LaTeX ................ 88
3.9. A few Words about Typography .......................... 89
  3.9.1. Hyphens ............................................... 89
  3.9.2. Hyphenation .......................................... 89
  3.9.3. Punctuation Marks .................................... 90
    3.9.3.1. Abbreviations and End of Sentence .............. 90
    3.9.3.2. Quotes ......................................... 91
  3.9.4. Ligatures ............................................. 91
  3.9.5. \LaTeX's Proper Names ............................... 92
  3.9.6. Units ................................................ 92
  3.9.7. Widows and Orphans .................................. 93

4. Notes, Graphics, Tables, and Floats ...................... 95
  4.1. Notes ..................................................... 95
  4.2. Footnotes ................................................ 95
  4.3. Marginal Notes .......................................... 96
  4.4. Graphics and Images .................................... 96
    4.4.1. Image Formats ..................................... 97
    4.4.2. Grouping of Image Settings ....................... 98
  4.5. Tables ................................................... 98
    4.5.1. The Table dialog .................................. 98
    4.5.2. Longtables ......................................... 99
    4.5.3. Table Cells ........................................ 101
  4.6. Floats ................................................... 102
    4.6.1. Float Types ....................................... 103
      4.6.1.1. Figure Floats ................................ 103
      4.6.1.2. Table Floats ................................ 104
5. Mathematical Formulas 109
   5.1. Basic Math Editing .................................... 109
       5.1.1. Navigating in Formulas .......................... 109
       5.1.2. Selecting Text .................................... 110
       5.1.3. Exponents and Subscripts ........................ 110
       5.1.4. Fractions .......................................... 110
       5.1.5. Roots .............................................. 111
       5.1.6. Operators with Limits ............................. 111
       5.1.7. Math Symbols ...................................... 111
       5.1.8. Altering Spacing ................................... 111
       5.1.9. Functions .......................................... 112
       5.1.10. Accents ........................................... 112
   5.2. Brackets and Delimiters ................................ 113
   5.3. Arrays and Multi-line Equations .......................... 113
   5.4. Formula Numbering and Referencing .......................... 115
   5.5. User defined math macros ................................ 116
   5.6. Fine-Tuning ........................................... 116
       5.6.1. Typefaces .......................................... 116
       5.6.2. Math Text .......................................... 117
       5.6.3. Font Sizes ......................................... 117
   5.7. Theorem Modules ......................................... 117
   5.8. AMS-\LaTeX ............................................. 118
       5.8.1. Enabling AMS-Support .............................. 118
       5.8.2. AMS-Formula Types ................................ 118
6. More Tools 119
   6.1. Cross-References ........................................ 119
   6.2. Table of Contents and other Listings ...................... 120
       6.2.1. Table of Contents .................................. 120
       6.2.2. List of Figures, Tables, and Algorithms ............ 121
   6.3. URLs and Hyperlinks ..................................... 121
       6.3.1. URLs .............................................. 121
       6.3.2. Hyperlinks ........................................ 121
   6.4. Appendices ............................................. 122
   6.5. Bibliography ........................................... 122
       6.5.1. The Bibliography Environment ....................... 122
       6.5.2. Bibliography databases (\BibTeX) .................... 123
       6.5.3. Bibliography layout ................................ 124
Contents

6.6. Index ................................................................. 124
  6.6.1. Grouping Index Entries ................................. 124
  6.6.2. Page Ranges .............................................. 125
  6.6.3. Cross referencing ...................................... 125
  6.6.4. Index Entry Order .................................... 125
  6.6.5. Index Entry Layout .................................. 126
  6.6.6. Index Program ......................................... 126
6.7. Nomenclature / Glossary ....................................... 127
  6.7.1. Nomenclature Definition and Layout .................. 127
  6.7.2. Sort Order of Nomenclature Entries .................. 127
  6.7.3. Nomenclature Options ................................ 128
  6.7.4. Printing the Nomenclature ............................ 128
  6.7.5. Nomenclature Program ................................ 129
6.8. Branches .......................................................... 129
6.9. PDF Properties .................................................. 130
6.10. \TeX\ Code and the \LaTeX\ Syntax .............................. 131
  6.10.1. \TeX\ Code Boxes ..................................... 131
  6.10.2. The \LaTeX\ Syntax .................................... 131
6.11. Previewing Snippets of your Document ....................... 132
6.12. Spell Checking ................................................ 133
6.13. Thesaurus ...................................................... 134
6.15. International Support ....................................... 136
  6.15.1. Language Options .................................... 136
  6.15.2. Keyboard mapping configuration ...................... 136
  6.15.3. Character Tables .................................... 136

A. The User Interface ............................................. 139
A.1. The File Menu ................................................ 139
  A.1.1. New ...................................................... 139
  A.1.2. New from Template .................................... 139
  A.1.3. Open .................................................... 139
  A.1.4. Open Recent .......................................... 139
  A.1.5. Close ................................................... 139
  A.1.6. Save .................................................... 139
  A.1.7. Save As ................................................ 140
  A.1.8. Save All ............................................... 140
  A.1.9. Revert to saved ...................................... 140
  A.1.10. Version Control ..................................... 140
  A.1.11. Import ................................................ 140
  A.1.12. Export ................................................ 140
  A.1.13. Print .................................................. 141
  A.1.14. Fax .................................................... 142
  A.1.15. New and Close Window ............................... 142
xlvi
Contents

A.1.6. Exit .............................................. 142
A.2. The Edit Menu ............................... 142
    A.2.1. Undo and Redo ......................... 142
    A.2.2. Cut, Copy, Paste, Paste Recent, Paste Special .......... 142
    A.2.3. Select All .......................... 142
    A.2.4. Find & Replace .................... 142
    A.2.5. Move Paragraph Up/Down .............. 142
    A.2.6. Text Style .......................... 142
    A.2.7. Paragraph Settings .................. 143
    A.2.8. Table Settings and Math .......... 143
    A.2.9. Increase / Decrease List Depth ........ 143
A.3. The View Menu ................................. 143
    A.3.1. Open/Close all Insets .......... 143
    A.3.2. Unfold/Fold Math Macros .......... 143
    A.3.3. View Source ...................... 144
    A.3.4. Update ................................ 144
    A.3.5. Split View ......................... 144
    A.3.6. Close Tab Group .................... 144
    A.3.7. Fullscreen .......................... 144
    A.3.8. Toolbars ............................. 144
A.4. The Insert Menu .............................. 145
    A.4.1. Math .................................. 145
    A.4.2. Special Character .................. 145
    A.4.3. Formatting ........................... 145
    A.4.4. List / TOC ........................... 146
    A.4.5. Float .................................. 146
    A.4.6. Note .................................. 146
    A.4.7. Branch ................................ 146
    A.4.8. Custom Insets ...................... 147
    A.4.9. File .................................. 147
    A.4.10. Box .................................. 147
    A.4.11. Citation ............................ 147
    A.4.12. Cross-Reference ..................... 147
    A.4.13. Label ................................. 147
    A.4.15. Index Entry ......................... 147
    A.4.16. Nomenclature Entry ................. 147
    A.4.17. Table .................................. 148
    A.4.18. Graphics ............................. 148
    A.4.19. URL ................................... 148
    A.4.20. Hyperlinks .......................... 148
    A.4.21. Footnote ............................. 148
    A.4.22. Marginal Note ....................... 148
    A.4.23. Short Title ......................... 148
1. Getting Started

1.1. What is LyX?

LyX is a document preparation system. It is a tool for producing beautiful manuscripts, publishable books, business letters and proposals, and even poetry. It is unlike most other “word processors” in the sense that it uses the paradigm of a markup language as its core editing style. That means that when you type a section header, you mark it as a “Section”, not “Bold, 17 pt type, left justified, 5 mm space below”. LyX takes care of the typesetting for you, so you deal only with concepts, not the mechanics.

This philosophy is explained in much greater detail in the “Introduction”. If you haven’t read it yet, you need to. Yes, we mean now.

The “Introduction” describes several things in addition to LyX’s philosophy: most importantly, the format of all of the manuals. If you don’t read it, you’ll have a bear of a time navigating this manual. You might also be better served looking in one of the other manuals instead of this one. “Introduction” describes that, too.

1.2. How LyX Looks

Like most applications, LyX has the familiar menu bar across the top of its window. Below it is a toolbar with a pulldown box and various buttons. There is, of course, a vertical scrollbar and a main work area for editing documents.

Note that there is no horizontal scroll bar. This is not a bug or an oversight, but intentional. When you read a book, you expect the end of a line to wrap around to the next line. Text overflows onto new pages in a vertical fashion, hence the need for only a vertical scrollbar. There are three cases where you might want a horizontal scrollbar. The first case is large figures, displayed WYSIWYG. This, however, is due to a flaw in the routine that displays graphics on the LyX screen in a WYSIWYG fashion; it should rescale the graphics to fit in the window, just as you’d need to rescale graphics to fit on a page. The second and third cases are tables and equations which are wider than the LyX window. You can use the arrow keys to scroll horizontally through the table, but this doesn’t work for equations yet.

For a brief description of all LyX menus and toolbar buttons, have a look at Appendix A. Most of them are self-explanatory and you’ll find them listed in the corresponding sections of this documentation.
1.3. HELP

The help system consists of the LyX manuals. You can read all of the manuals from inside LyX. Just select the manual you want read from the Help menu.

1.4. Basic LyX Setup

There are several features of LyX that can be configured from inside LyX, without resorting to configuration files. First, LyX is able to inspect your system to see what programs, \LaTeX\ document classes and \LaTeX\ packages are available. It uses this knowledge to give reasonable defaults to several Preferences settings. Although this configuration has already been done when LyX was installed on your system, you might have some items that you installed locally, e.g. new \LaTeX\ classes, and which are not seen by LyX. To force LyX to re-inspect your system, you should use Tools\textgreater Reconfigure. You should then restart LyX to ensure that the changes are taken into account.

1.5. \LaTeX\ Setup

LyX needs several \LaTeX\ packages to work properly. The packages found on the system by LyX are listed in the file LaTeX Configuration that will be created when using the menu Help\textgreater LaTeX Configuration. You should install the required missing packages and then reconfigure LyX.
1. Getting Started
2. How to work with \textsc{LyX}

2.1. Basic File Operations

Under the \texttt{File} menu and in the standard toolbar are basic operations for any word processor in addition to some more advanced operations:

- New
- New from Template
- Open
- Close
- Save
- Save As
- Revert to saved
- Version Control
- Import
- Export
- Print
- Exit

They all do pretty much the same thing as in other word processors, with a few minor differences. The \texttt{File} $\rightarrow$ \texttt{New from Template} command not only prompts you for a name for the new file, but also prompts you for a template to use. Selecting a template will automatically set certain layout features for the document, features you would otherwise need to change manually. They can be of use for certain classes, especially those for writing letters (see section 3.1.2).

Note: There is no “default file” or document named “Untitled” or “scratch.” Unless you tell \textsc{LyX} to open a file or create a new one, that big, blank space is just that — a big, blank space.
2. How to work with LyX

Revert to saved and Version Control are useful if more people work on the same
document at the same time. Revert to saved will reload the document from disk.
You can of course also use it if you regret that you changed a document and want to
restore it to the last save. With Version Control you can there register the changes
you made to a document so that others can identify them as your changes.

2.2. Basic Editing Features

Like most modern word processors, LyX can perform cut and paste operations on
blocks of text, can move by character, word, or page of text, and can delete whole
words as well as individual characters. The next four sections cover the basic LyX
editing features and how to access them. We'll start with cut and paste.

As you might expect, the Edit menu and the standard toolbar has the cut and paste
commands, along with various other editing features. Some of these are special and
covered in later sections. The basic ones are:

- Cut
- Copy
- Paste
- Paste Recent
- Paste Special
- Find & Replace

The first three are self-explanatory. One thing to note: whenever you delete a block
of text that you’ve selected, it’s automatically placed in the clipboard. That is, the
Delete and Backspace keys also functions as the Cut command. Also, if you’ve selected
text, be careful: If you hit a key, LyX will completely delete the selected text and
replace it with what you just typed. You’ll have to do an Undo to get back the lost
text.

You can also copy text between LyX and other programs by cut, copy and paste.
The submenu of Paste Recent shows you a list with the last strings you have pasted.

The menu Edit>Paste Special>Plain Text, Join Lines will insert the text in the
clipboard so that the whole text is inserted as one paragraph. A new paragraph is
started when there is a blank line in the file. Without Join Lines, the text is inserted
as Paragraphs, where the line breaks of the text will start a new paragraph.

The Edit>Find & Replace item opens the Find and Replace dialog. Once you have
found a word or expression, LyX selects it. Hitting the Replace button replaces the

\[1\]If you plan to do this, you should check out the Version Control feature in LyX also. Read
Additional Features.
selected text with the contents of the Replace with field. You can click the Find Next button to skip the current word. Hit Replace All to replace all occurrences of the text in the document automatically. The Case sensitive option can be used if you want the search to consider the case of the search word. If the toggle is set, searching for “Test” will not match the word “test”. The Match whole words only option can be used to force \LaTeX{} to only find complete words, i.e., searching for “star” will not match “starlet”.

Things like notes, floats, etc., the so called insets can be dissolved. This means that the inset is deleted and its content is left as normal text. Dissolving an inset is done by setting the cursor to the beginning of an inset and pressing Backspace or by setting the cursor to the end and pressing Delete.

2.3. Undo and Redo

If you make a mistake, you can easily recover from it. \LaTeX{} has a large-capacity undo/redo buffer. Select Edit $\rightarrow$ Undo or the toolbar button \includegraphics[width=8em]{undo} to undo some mistake. If you accidentally undo too much, use Edit $\rightarrow$ Redo or the toolbar button \includegraphics[width=8em]{redo} to “undo the undo.” The undo mechanism is currently limited to 100 steps to minimize memory overhead.

Note that if you revert back all changes to arrive to the document as it was last saved, the “changed” status of the document is unfortunately not reset. This is a consequence of the 100-step undo limit, above.

Undo and Redo work on almost everything in \LaTeX{}. But they won’t undo or redo text character by character, but by blocks of text.

2.4. Mouse Operations

These are the most basic mouse operations.

1. Motion
   - Click the left mouse button once anywhere in the edit window. The cursor moves to the text under the mouse.

2. Selecting Text
   - Hold down the left mouse button and drag the mouse. \LaTeX{} marks the text between the old and new mouse positions. Use Edit $\rightarrow$ Copy to create a copy of the text in \LaTeX{}’s buffer (and the clipboard).
   - Re-position the cursor and then paste the text back into \LaTeX{} using Edit $\rightarrow$ Paste.

3. Insets (Footnotes, Notes, Floats, etc.)
2. How to work with LYX

Right-click on them to set their properties. Also check the appropriate section of this manual for more details.

4. Tables

*Single click* the right mouse button to open a dialog that will allow you to manipulate the table.

2.5. Navigating

LYX offers you two ways to navigate in documents:

- The menu **Navigate** lists all sections of the document as submenu entries that you can click to jump to the corresponding document part.

- The menu **Document** → **Outline** or the toolbar button 📚.

The outline window shows you the content of the document’s table of contents (TOC) that is described in section 6.2.1. You can click there on entries to jump to the corresponding document part. In the pull-down box at the top of the outline window, you can choose between different lists of document objects, like the list of footnotes. Some of them, the list of tables, figures, and algorithms can also be added to the document, see section 6.2.2. The **Sort** option sorts the current list, and the **Keep** option keeps it in the current view state. Keeping means that when you have e.g. the subsections of section 2 and 4 shown and click on section 3, the subsections of section 2 and 4 will still be shown. Without the **Keep** option they will be hidden to highlight the clicked section 3.

With the buttons 🔽 and 🔼 at the bottom of the outline window you can change the position of sections in your document. So you can for example move section 2.5 before section 2.4. LYX will then automatically renumber the sections to the new order. With the buttons 🔽 and 🔼 or the corresponding key bindings **Tab** and **Shift-Tab** you can change the sectioning level of sections. So you can for example make section 2.5 to chapter 3 or to subsection 2.4.1.

The toolbar button 🔄 jumps to the position in the document where you recently changed something. This is for example useful when you have a large document and navigated or scrolled to another document part to look for something, and want to go back to your last editing position.

2.6. Input / Word Completion

LYX provides completion of words by scanning all documents that are currently opened. Every word that appears in these documents is added to a database that is used to propose completions.
By default \LaTeX{} shows a small triangle behind the cursor as indicator that there are completions available. You can then press the Tab key to use this completion. When several completions are possible, a popup is opened showing them. You can select a completion in the popup using the mouse or the arrow keys, and accept the chosen completion by pressing Return.

In the preferences dialog, that is opened with the menu Tools->Preferences, the cursor completion indicator can be turned off in the section Editing->Input Completion by deselecting the option Cursor indicator. With the option Automatic inline completion the proposed completion is shown directly behind the cursor position. To accept this proposal, use the Tab key. With the option Automatic popup the completions are always shown in a popup. \LaTeX{} offers some more completion settings for experts that are described in sec. C.2.4.

### 2.7. Basic Key Bindings

There are at least two different primary binding maps: CUA and Emacs. \LaTeX{}’s default is CUA.

Some keys, like Page Up, Page Down, Left, Right, Up, and Down, do exactly what you expect them to do. Other keys don’t:

- **Tab**: There is no such thing as a tab stop in \LaTeX{}. If you don’t understand this, go read sections 3.2.1 and 3.3, especially section 3.3.6, right now. Yes, right now. If you’re still confused, look in the Tutorial.

- **Esc**: This is the “cancel key.” It’s used, generically, to cancel operations. Other parts of the manual will go into greater detail about this.

- **Home** and **End**: These move the cursor, respectively, to the beginning and end of a line, unless you are using the Emacs bindings where they jump to the beginning or end of the file.

There are three modifier keys:

- **Control**: (Denoted by “Ctrl” in the documentation files) This has a couple of different uses, depending on which keys it’s used in combination with:
  - With Backspace or Delete, it deletes an entire word instead of a single character.
  - With Left and Right, it moves by words instead of characters.
  - With Home and End, it moves to the beginning and the end of the document, respectively.

- **Shift**: (Denoted by “Shift” in the documentation files) Use this with any of the motion keys to select the text between the old and new cursor positions.
2. How to work with LyX

Alt (Denoted by “Alt” in the documentation files) This is the Alt key on many keyboards, unless your keyboard has a distinct Meta key. If you have both keys, you will need to try out which one actually performs the Alt+ function. This key does many different things, but it also activates the menu accelerator keys. If you use this in combination with any of the underlined letters in a menu or menu item, it selects that menu item.

For example, the sequence “Alt e s c” brings up the “Text Style” menu. Typing “Alt f” opens the File menu.

The Shortcuts manual lists all other things bound to the Alt key.

You’ll learn more and more keybindings and short-cut keys as you use LyX, because most actions will prompt a small message in the status bar at the bottom of LyX’s main window which describe the name of the action, you’ve just triggered, and any existing keybindings for that action. The LyX menus also list the defined keybindings. The notation for the keybindings is very similar to the notation used in this documentation, so you should not have any problems understanding it. However, notice that Shift-modifiers are explicitly mentioned, so “Alt+P Shift-A” means Alt+P followed by a capital A.

You can list or change the keybindings in the menu Tools > Preferences under Editing > Shortcuts as explained in sec. C.2.2.2.
3. LyX Basics

3.1. Document Types

3.1.1. Introduction

Before you do anything else, before you ever start writing a document, you need to decide what type of document you want to edit. Different types of documents use different types of spacing, headings, numbering schemes, and so on. Additionally, different documents use different paragraph environments, and format the title of your document differently.

A document class describes a group of properties common to a particular set of documents. By setting the document class, you automatically select these properties, making it easier to create the type of document you want. If you don’t choose a document class, LyX picks one for you by default. So it is up to you to change the class of your document.

Read on for info about the document classes you can choose from LyX, and how to adjust their properties.

3.1.2. Document Classes

You can select a class using the Document ➤ Settings dialog. Select the class you want to use, and make any fine tunings of the options you may need.

3.1.2.1. Overview

There are four standard document classes in LyX. They are:

- **Article** for basic articles
- **Report** for basic reports
- **Book** for writing a book
- **Letter** for US-style letters

There are also some non-standard classes, which LyX only uses if you have installed them. Here are some of the classes, the full list with detailed explanations can be found in chapter *Special Document Classes* in the *Additional Features* manual:

- **A&A** Journal articles in the style and format used in Astronomy & Astrophysics
3. \LaTeX Basics

**AASTeX** For submissions to the journals published by the American Astronomical Society

**AMS** Layouts for articles and books in the style and format used by the American Mathematical Society (AMS). There are three article layouts available. The standard one uses a typical numbering scheme for theorems etc., that prepends the section number to the number of the result. All result-type statements (propositions, corollaries, and so on) are sequenced together, but definitions, examples, and the like have their own sequence. The “sequential numbering” scheme does not place the section number with each result, but numbers them throughout the article in a single sequence. Each type of result gets its own sequence. There is also a layout that dispenses with numbering of statements altogether.

**Beamer** Layout for presentations

**broadway** Layout for writing plays. It is not an existing \LaTeX document class, but a new one which is distributed with \LaTeX.

**curriculum vitae** classes to create curriculum vitae

**Dinbrief** Letters in format of the DIN (German industry norm)

**dtk** Layout for Die \TeXnische Komödie, the journal of the German \TeX user Group (Dante)

**Elsevier** Layout for journals of the Elsevier publishing group

**Foils** Used to make transparencies

**g-brief** Letters in format of the DIN (German industry norm)

**hollywood** Used to type spec scripts for the US film industry. It is not an existing \LaTeX document class, but a new one which is distributed with \LaTeX.

**IEEEtran** Layout for the journals published by the Institute of Electrical and Electronics Engineers (IEEE)

**IOP** Layout for journals of the Institute of Physics publishing group

**Kluwer** Layout for journals of the Kluwer publishing group

**koma-script** a replacement for the standard classes, offers many useful features like caption formatting, automatic print space calculation etc.

**Memoir** another replacement for the standard classes

**paper** Used with the paper \LaTeX document class
3.1. Document Types

**Powerdot**  Layout for presentations

**REVTeX**  is used to write articles for the publications of the American Physical Society (APS), American Institute of Physics (AIP), and Optical Society of America (OSA). This class is not completely compatible with all \LaTeX{} features.

**Slides**  Used to make transparencies

**SPIE Proceedings**  Layout for the journals published by The International Society for Optical Engineering (SPIE)

**Springer**  Layouts for journals of the Springer publishing group

**TUGboat**  Layout for *TUGboat*, the journal of the international \TeX{} user Group (TUG)

We won’t go into any detail about how to use these different document classes here. You can find all the details about the non-standard classes in the *Additional Features* manual. Here, we will settle with a list of some of the common properties of all of the document classes.

### 3.1.2.2. Modules

Modules load additional features to a document that are not by default available in the chosen document class. For example you might want to use write Braille (embossed printing) in a document. This is of course not available in any document class, so you have to load the corresponding module in the **Modules** section of the **Document > Settings** dialog. Highlighting a module in the dialog will bring up a description of the module.

**Note:** Some modules require \TeX{} packages that are not always installed by default. \LaTeX{} will warn you if you do not have the needed package.

**Note also:** Some modules require other modules, and some pairs of modules are incompatible.

### 3.1.2.3. Properties

Each class has a default set of options. Here’s a quick table describing them:

<table>
<thead>
<tr>
<th></th>
<th>Page style</th>
<th>Sides</th>
<th>Columns</th>
<th>Max. sectioning level</th>
</tr>
</thead>
<tbody>
<tr>
<td>article</td>
<td>Plain</td>
<td>One</td>
<td>One</td>
<td>Section</td>
</tr>
<tr>
<td>report</td>
<td>Plain</td>
<td>One</td>
<td>One</td>
<td>Chapter</td>
</tr>
<tr>
<td>book</td>
<td>Headings</td>
<td>Two</td>
<td>One</td>
<td>Chapter</td>
</tr>
<tr>
<td>letter</td>
<td>Plain</td>
<td>One</td>
<td>One</td>
<td>none</td>
</tr>
</tbody>
</table>
You’re probably also wondering what “Max. sectioning level” means. There are several paragraph environments used to create section headings. Different document classes allow different types of section headings. Only two use the Chapter heading; the rest do not and begin instead with the Section heading. Some document classes, such as the ones for letters, don’t use any section headings. In addition to Chapter and Section headings, there are also Subsection headings, Subsubsection headings, and so on. We’ll describe these headings fully in section 3.3.4.

### 3.1.3. Document Layout

The most important properties of documents classes are set in the menu Document\rightarrow Settings. There in the Options field under Documents Class, you can enter special options for your document class in a comma-separated list. This is only necessary if \LaTeX{} doesn’t support special options you want to use for your document. To learn more about your favorite \LaTeX-class and its options, you have to read its manual.

The drop box Headings style in the Document\rightarrow Settings dialog under Page Layout controls what sorts of headings and page numbers go on a page. You can choose between the following five options:

- **Default** Use default page style of current class.
- **Empty** No page numbers or headings.
- **Plain** Page numbers only.
- **Headings** Page numbers and either the current chapter or section title and number. Whether \LaTeX{} uses the current chapter or the current section depends on the maximum sectioning level of the class.
- **Fancy** This allows you to create fully customizable headers and footers if you have the fancyhdr package installed. At the moment, support in \LaTeX{} is limited to this setting. To use the full power of this package, you have to add code to your document preamble. Check the documentation for the fancyhdr package for more details, [12].

The Separation of paragraphs is described in section 3.2.1.

### 3.1.4. Paper Size and Orientation

You’ll find the following options in the menu Page Layout of the dialog of the Document\rightarrow Settings menu:

- **Paper Format** What size paper to print on. The choices are
  
  - Default
  - A3, A4, A5

52
3.2. Paragraph Indentation and Separation

- B3, B4, B5
- US letter
- US legal
- US executive
- Custom

Orientation Two toggle buttons choose whether to print the output as Landscape or as Portrait.

Two-sided document Adjusts the print space to print both sides of paper. That means that the print space for odd- and even-numbered pages is different.

3.1.5. Margins

Paper margins are set in the menu Document ➤ Settings.

If you use a koma-script document class, you can use the default settings. Because koma-script calculates then the printspace automatically by taking the paper format and the font size into account.

3.1.6. Important Note

If you change a document class, LYX has to convert everything into the new class. That includes the paragraph environments. Some paragraph environments are standard; all of the document classes have them; but some classes have special paragraph environments. If this is the case, and you change the document class, LYX sets the missing paragraph environments to Standard and places an error box at the beginning of the paragraph. Just click on them and you’ll get a message dialog that tells you about the conversion and why it failed.

3.2. Paragraph Indentation and Separation

3.2.1. Introduction

Before describing all of the various paragraph environments, we’d like to say a word or two about paragraph indentation.

Everyone seems to have their own convention for separating paragraphs. Most Americans indent the first line of a paragraph. Others don’t indent but put extra space between the paragraphs. If you choose indentation for paragraphs the first paragraph of a section, or after a figure, an equation, a table, a list, etc., is not indented. Only a paragraph following another paragraph gets indented. Note that
the indentation behavior is different when you use another document language than English. \LaTeX{} takes care that the indentation follows the rules of the used language.

The space between paragraphs, like the line spacing, the space between headings and text — in fact, all of the spacings for just about everything are pre-coded into \lyx. As we said, you don’t worry about how much space to add between what. \lyx{} takes care of that. In fact, these pre-coded vertical spacings aren’t a single number but a range. That way, \lyx can squish or stretch the space between lines to make sure figures fit on a page with text, so that sections don’t start at the bottom of a page, and so on.\footnote{\LaTeX{} does this when \lyx{} goes to produce a printable file.} However, pre-coded doesn’t mean you can’t change them. \lyx gives you the ability to globally change all of these pre-coded spacings. We’ll explain more later.

### 3.2.2. Paragraph Separation

To separate paragraphs, select **Indent** or **Skip** in the submenu **Text Layout** of the dialog **Document \rightarrow Settings** to indent paragraphs or add extra space between paragraphs, respectively. The size of the skips can be defined in the dialog, for the indentation you have to add this line to your document preamble:

\[
\setlength{\parindent}{Length}
\]

where length is a value in one of the units listed in Appendix D.1. The default length is 30 pt.

### 3.2.3. Fine-Tuning

You can also change the separation method of a single paragraph. Open the **Edit \rightarrow Paragraph Settings** dialog and toggle the **Indent Paragraph** option to change the state of the current paragraph (shortcut Alt+A I). If paragraphs have no indentation but use extra space for separation, this button will be ignored (you can’t indent a single paragraph by toggling this).

You should only need to change the indentation method for a single paragraph if you need to do some fine-tuning.

### 3.2.4. Line Spacing

In the **Document \rightarrow Settings** dialog you can set the line spacing in the submenu **Text Layout**.\footnote{You need to have the \LaTeX-package *setspace* installed to use this feature.}
3.3. Paragraph Environments

3.3.1. Overview

Paragraph environments correspond to the \begin{environment} ... \end{environment} command sequence in \LaTeX files. If you don’t know \LaTeX, or the concept of a paragraph environment is totally alien to you, we urge you to read the Tutorial. The Tutorial also contains many more examples than this section does.

A paragraph environment is simply a “container” for a paragraph which gives that paragraph certain properties. This can include a particular style of font, different margins, a numbering scheme, labels, and so on. Additionally, you can “nest” the different environments inside one another, allowing one environment to inherit some of the properties of another. The different paragraph environments totally replace the need for messy tab stops, on the fly margin adjustment, and other hold-overs from the days of typewriters. There are several paragraph environments which are specific to a particular document type. We’ll only be covering the most common ones here.

To choose a new paragraph environment, use the pull-down box at the left end of the toolbar. LYX will change the environment of the entire paragraph in which the cursor sits. You can also change the environment of an entire group of paragraphs if you select them before choosing the new environment.

Note that hitting Return will typically create a new paragraph using the Standard paragraph environment. We say “typically” because if you are in one of these environments:

- Quote
- Quotation
- Verse
- Itemize
- Enumerate
- Description
- List

LYX keeps the old paragraph environment when you hit Return, rather than resetting it to Standard. LYX will still reset the nesting depth, however. Usually, starting a new paragraph resets both the paragraph environment and the nesting depth (for more on nesting see section 3.4). At the moment, all this is context-specific; you’re better off expecting Return to reset the paragraph environment and depth. If you want a new paragraph to keep the current environment and depth, use Alt+Return instead.
3. **LyX Basics**

### 3.3.2. Standard

The default paragraph environment is **Standard** for most classes. It creates a plain paragraph. If LyX resets the paragraph environment, this is the one it chooses. In fact, the paragraph you’re reading right now (and most of the ones in this manual) are in the **Standard** environment.

You can nest a paragraph using the **Standard** environment in just about anything else, but you can’t really nest anything in a **Standard** environment.

### 3.3.3. Document Title

A **\LaTeX** title page has three parts: the title itself, the name(s) of the author(s) and a “footnote” for thanks or contact information. For certain types of documents, **\LaTeX** places all of this on a separate page along with today’s date. For other types of documents, the title “page” goes at the top of the first page of the document.

**LyX** provides an interface to the title page commands through the paragraph environments **Title**, **Author**, and **Date**. Here’s how you use them:

- Put the title of your document in the **Title** environment.
- Put the author name in the **Author** environment.
- If you want the date to have a certain appearance, want to use a fixed date, or want other text to appear in place of today’s date, put that text in the **Date** environment. Note that using this environment is optional. If you don’t provide any, **\LaTeX** will automatically insert today’s date. If you don’t want any date, add the line \texttt{\textbackslash date{}} to the preamble of your document (menu **Document\rightarrow Settings**)

You can use footnotes to insert “thanks” or contact information.

### 3.3.4. Headings

There are several paragraph environments for producing section headings. **LyX** takes care of the numbering for you.

#### 3.3.4.1. Numbered Headings

There are 7 numbered types of section headings. They are:

1. Part
2. Chapter
3. Section
4. Subsection
5. Subsubsection
6. Paragraph
7. Subparagraph

LYX labels each heading with a series of numbers, separated by periods. The numbers describe where in the document you are. Unlike the other headings, parts are numbered with Latin letters.

Headings all subdivide your document into different pieces of text. For example, suppose you’re writing a book. You group the book into chapters. LYX does similar grouping:

- **Part** is divided in either **Chapters** or **Sections**
- **Chapters** are divided into **Sections**
- **Sections** are divided into **Subsections**
- **Subsections** are divided into **Subsubsections**
- **Subsubsections** are divided into **Paragraphs**
- **Paragraphs** are divided into **Subparagraphs**

**Note:** Not all document types use the **Chapter** heading as the maximum sectioning level. In that case the **Section** is the top-level heading.

So, if you use the **Subsubsection** environment to label a new sub-subsection, LYX labels it with its number, along with the number of the subsection, section, and, if applicable, chapter that it’s in. For example: the fifth section of the second chapter of this book has the label “2.5”.

### 3.3.4.2. Unnumbered Headings

There are 5 types of unnumbered section headings. They are:

1. **Part**
2. **Chapter**
3. **Section**
4. **Subsection**
5. **Subsubsection**

The “*” after each name means that these headings are not numbered. They work the same as their numbered counterparts but won’t appear in the table of contents, see section 6.2.
3. **LyX Basics**

### 3.3.4.3. Changing the Numbering

You can also alter which sectioning levels get numbered and which ones appear in the Table of Contents. Now, this doesn’t remove any of the levels; that’s preset in the document class. Certain classes start with **Chapter** and go down to the **Subparagraph** level. Others start at **Section**. Similarly, not all document classes number all sectioning levels. Most don’t number **Paragraph** or **Subparagraph**. This is something you can change.

Open the Document ➤ Settings dialog. Under Numbering & TOC you’ll see two counters. The one named **Numbering** controls how far down in the sectioning hierarchy LyX numbers a section heading. The other one controls the appearance of the section headings in the table of contents.

### 3.3.4.4. Short Titles of Headings

Some section or chapter titles, such as this one, can get quite long. This can cause trouble when there is limited horizontal space. For example, if the header of the page is set to show the current section title, a long title will protrude over the page margins and look awful.

**L****e** **T** **X** allows you to specify a short title for section headings. This short title is used in the header and in the actual table of contents, avoiding the problem mentioned. To specify a short title, use the menu **Insert** ➤ **Short Title**. This will insert a box labeled “opt” (stands for “optional”) which you can use to enter the short title text. This also works for captions inside floats.

The title of this section is a good example of using this feature.

### 3.3.4.5. Special Information

The following information applies to all section headings:

- You cannot do any nesting with these environments.
- You cannot use a margin note in any of these environments.
- You can only use inline math in these environments.
- You can use labels and cross-references to refer to their numbers.

### 3.3.5. Quotes and Poetry line spacing

**LyX** has three paragraph environments for writing poetry and quotations. They are **Quote**, **Quotation**, and **Verse**. Forget the days of changing line spacing and twiddling with margins. These three paragraph environments already have those changes built-in. They all widen the left margin and add a bit of extra space above and below the text they contain. They also allow nesting, so you can put a **Verse** in a **Quotation**, as well as in some other paragraph environments.
There is another feature of these three paragraph environments: they do not reset to Standard when you start a new paragraph. So, you can type in that poem and merrily hit Return without worrying about the paragraph environment changing on you. Of course, that means that, once you’re done typing in that poem, you have to change back to the Standard environment yourself.

3.3.5.1. Quote and Quotation

Now that we’ve described the similarities of these three environments, it’s time for the differences. Quote and Quotation are identical except for one difference: Quote uses extra spacing to separate paragraphs and never indents the first line. Quotation always indents the first line of a paragraph and uses the same line spacing throughout.

Here’s an example of the Quote environment:

This is in the Quote environment. I can keep writing, extending this line out further and further until it wraps. See – no indentation!

Here’s the second paragraph of this quote. Again, there’s no indentation, but there is extra space between me and the other paragraph.

Here’s another example, this time in the Quotation environment:

This is in the Quotation environment. If I keep writing, you’ll see the indentation. If your country uses a writing style that shows off new paragraphs by indenting the first line, then Quotation is the environment for you! Well, you’d use it if you were quoting other text.

Here’s a new paragraph. I could ramble on and on, like a politician at election time. If I did that, though, you’d get bored.

As the examples show, Quote is for those people who use extra space to separate paragraphs. They should put quotes in the Quote environment. Those who use indentation to mark a new paragraph should use the Quotation paragraph environment for quoted text.

3.3.5.2. Verse

Verse is a paragraph environment for poetry, rhymes, verses, and so on. Here’s an example:

This is in Verse
Which I did not rehearse!

It could be much worse. This line could be long, very long, oh so long, so very long that it wraps around. It looks okay on screen, but in the printed version, the extra lines are indented a bit more than the first. Okay, so it’s turned to prose and doesn’t rhyme anymore. So sue me.

To break a line
And make things look fine
Use Ctrl+Return.
As you can see, **Verse** does not indent both margins. Each stanza of the verse or poem is in its own paragraph. To separate the individual lines of a stanza, use the **break-line** function `Ctrl+Return`.

### 3.3.6. Lists

**LyX** has four different paragraph environments for creating different kinds of lists. In the **Itemize** and **Enumerate** environments, **LyX** labels your list items with bullets or numbers, respectively. In the **Description** and **List** environments, **LyX** lets you provide your own label. We’ll present the individual details of each type of list next after describing some general features of all four of them.

#### 3.3.6.1. General Features

The four paragraph environments for lists differ from the other environments in several ways. First, **LyX** treats each paragraph as a list item. Hitting `Return` does *not* reset the environment to **Standard** but keeps the current environment and creates a new list item. The nesting depth is hereby kept. If you want to keep the paragraph environment but reset the current nesting depth, you can use `Alt+Return` to break paragraphs.

You can nest lists of any type inside one another. In fact, **LyX** changes the labels on some list items depending on how it is nested. If you intend to use any of the list paragraph environments, we suggest you read all of section 3.4.

#### 3.3.6.2. Itemize

The first type of list we’ll describe in detail is the **Itemize** paragraph environment. It has the following properties:

- Each item has a particular bullet or symbol as its label.
  - **LyX** uses the same symbol for all of the items in a given nesting level.
  - The symbol appears at the beginning of the first line.

- The items can have any length. **LyX** automatically offsets the left margin of each item. The offset is always relative to whatever environment the **Itemize** list may be in.

- If you nest an **Itemize** environment inside another **Itemize** environment, the label changes to a new symbol.
  - There are four different symbols for up to a four-fold nesting.
  - **LyX** always shows the same symbol on screen.
  - See section 3.4 for a full explanation of nesting.
Of course, that explanation was also an example of an Itemize list. The Itemize environment is best suited for lists where the order doesn’t matter.

We said that different levels use different symbols as their label. Here’s an example of all four possible symbols.

- The label for the first level Itemize is a large black dot, or bullet.
  - The label for the second level is a dash.
    - The label for the third is an asterisk.
    - The label for the fourth is a centered dot.
    - Back out to the third level.
  - Back to the second level.
- Back to the outermost level.

These are the default labels for an Itemize list. You can customize these labels in the Document \textit{Settings} dialog in the submenu \textit{Bullets}.

Notice how the space between items decreases with increasing depth. We’ll explain nesting and all the tricks you can do with different depths in section 3.4.

### 3.3.6.3. Enumerate

The Enumerate environment is used to create numbered lists and outlines. It has these properties:

1. Each item has a numeral as its label.
   a) The label type depends on the nesting depth.
2. \LaTeX{} automatically counts the items for you and updates the label as appropriate.
3. Each new Enumerate environment resets the counter to one.
4. Like the Itemize environment, the Enumerate environment:
   a) Offsets the items relative to the left margin. Items can have any length.
   b) Reduces the space between items as the nesting depth increases.
   c) Uses different types of labels depending on the nesting depth.
   d) Allows up to a four-fold nesting.

Unlike the Itemize environment, Enumerate shows the different labels for each item in \LaTeX{}. Here is how \LaTeX{} labels the four different levels in an Enumerate:

1. The first level of an Enumerate uses Arabic numerals followed by a period.
3. \textit{\LaTeX} Basics

a) The second level uses lower case letters surrounded by parentheses.

i. The third level uses lower-case Roman numerals followed by a period.

A. The fourth level uses capital letters followed by a period.

B. Again, notice the decrease in the spacing between items as the nesting depth increases.

ii. Back to the third level

b) Back to the second level.

2. Back to the outermost level.

Once again, you can customize the type of numbering used in the \texttt{Enumerate} environment. It involves adding commands to the \LaTeX \texttt{preamble} (see the Additional Features manual). As stated earlier, such customization only shows up in the printed version, not in \LaTeX.

There is more to nesting \texttt{Enumerate} environments than we’ve stated here. You should read section 3.4 to learn more about nesting.

3.3.6.4. Description

Unlike the previous two environments, the \texttt{Description} list has no fixed label. Instead, \LaTeX uses the first “word” of the first line as the label. Here’s an example:

\textbf{Example:} This is an example of the \texttt{Description} environment.

\LaTeX typesets the label in boldface and puts extra space between it and the rest of the line.

With the first “word” it is meant that the first hit of the \texttt{Space} key ends the label if you are at the beginning of the first line of an item. If you need to use more than one word in the label use a \texttt{Protected Blank}. (Use either Ctrl+\texttt{Space} or the menu Insert $\Rightarrow$ Formatting $\Rightarrow$ Protected Space, see section 3.5.1 for more info.) Here is an example:

\textbf{Second Example:} This one shows how to use a \texttt{Protected Blank} in the label of a \texttt{Description} list item.

\textbf{Usage:} You should use the \texttt{Description} environment for things like definitions and theorems. Use it when you need to make one word in particular stand out in the text that describes it. It’s not a good idea to use a \texttt{Description} environment when you have an entire sentence that you want to describe. You’re better off using \texttt{Itemize} or \texttt{Enumerate} and nesting several \texttt{Standard} paragraphs into them.

\textbf{Nesting:} You can nest \texttt{Description} environments inside one another, nest them in other types of lists, and so on.

Notice that after the first line, \LaTeX indents subsequent lines, offsetting them from the first line.
3.3. Paragraph Environments

3.3.6.5. The LyX List

The List environment is a LyX extension to \LaTeX.

**Note:** When you are using a KOMA-Script document class, like in this document, the List environment is named Labeling.

Like the Description environment the List environment has user-defined labels for each list item. There are the following properties of this list environment:

- **item labels** LyX uses the first “word” of each line as the item label. The first Space after the beginning of the first line of an item marks the end of the label. If you need to use more than one word in an item label, use a protected blank as described above.

- **margins** As you can see, LyX uses different margins for the item label and the body of the item text. The body of the text has a larger left margin, which is equal to the default label width plus a little extra space.

- **label width** LyX uses the width of the label, or the default width, whatever is larger. If the label width is larger, the label “extends” into the first line. In other words, the text of the first line isn’t aligned with the left margin of the rest of the item text.

- **default width** You can set the default label width to ensure that the text of all items in a List environment have the same left margin. To change the default width, select all items in the list. Now open the Edit > Paragraph Settings dialog. The text in the box Longest label determines the default label width. You can use the text of your largest label here, but you can also use the letter “M” multiple times instead. The M is the widest character and is a standard unit of widths in \LaTeX. By using “M” as the unit of width you don’t need to keep changing the contents of Longest label every time you alter a label in a List environment. The predefined default width is the length of “00.00.0000” (equal to 6 M). **Note:** Setting the cursor into a list item to change only its label width will only change the width inside LyX but not in the output.

You should use the List environment the same way like the Description list: When you need one word to stand out from the text that describes it. The List environment gives you another way to do this, using a different overall layout.

You can nest List environments inside one another, nest them in other types of lists, and so on. They work just like the other list paragraph environments. Read section 3.4 to learn about nesting.

There is yet another feature of the List environment: As you can see in the examples, LyX left-justifies the item labels by default. You can use additional Hfills to change how LyX justifies the item label. Hfills are documented in section 3.5.2. Here are some examples:
3. \textbf{LyX Basics}

Left The default for List item labels.

Right One \texttt{HFill} at the beginning of the label right justifies it.

Center One \texttt{HFill} at the beginning of the label and one at the end centers it.

3.3.7. \textbf{Letters}

3.3.7.1. \textbf{Address and Right Address: An Overview}

Although \texttt{LyX} has document classes for letters, we’ve also created two paragraph environments called \texttt{Address} and \texttt{Right Address}. To use the letter class, you need to use specific paragraph environments in a specific order, otherwise \texttt{LaTeX} gags on the document. In contrast, you can use the \texttt{Address} and \texttt{Right Address} paragraph environments anywhere with no problem. You can even nest them inside other environments, though you can’t nest anything in them.

Of course, you’re not limited to using \texttt{Address} and \texttt{Right Address} for letters only. \texttt{Right Address}, in particular, is useful for creating article titles like those used in some European academic papers.

3.3.7.2. \textbf{Usage}

The \texttt{Address} environment formats text in the style of an address, which is also used for the opening and signature in some countries. Similarly, the \texttt{Right Address} environment formats text in the style of a right-justified address, which is used for the sender’s address and today’s date in some countries. Here’s an example of each:

Right Address
Who I am
Where I am
When is it? What is today?

That was \texttt{Right Address}. Notice that the lines all have the same left margin, which \texttt{LyX} sets to fit the largest block of text on a single line. Here’s an example of the \texttt{Address} environment:

Who are you
Where do I send this
Your post office and country

As you can see, both \texttt{Address} and \texttt{Right Address} add extra space between themselves and the next paragraph. If you hit \texttt{Return} in either of these environments, \texttt{LyX} resets the nesting depth and sets the environment to \texttt{Standard}. This makes sense, since \texttt{Return} is the \texttt{break-paragraph} function, and the individual lines of an address are not paragraphs. Thus, you have to use \texttt{break-line} (\texttt{Ctrl+Return} or Formatting Character> Line Break from the Insert menu) to start a new line in an \texttt{Address} or \texttt{Right Address} environment.
Abstract

This is an abstract. As you can see, it is printed in a smaller font size than the other paragraph types.

Also several paragraphs are possible in the abstract.

This is a “Standard” paragraph to visualize the differences in the font size.

Figure 3.1.: Paragraph in the Abstract environment

3.3.8. Academic Writing

Most academic writing begins with an abstract and ends with a bibliography or list of references. LYX contains paragraph environments for both of these.

3.3.8.1. Abstract

The Abstract environment is used for the abstract of an article. Technically, you could use this environment anywhere, but you really should only use it at the beginning of the document, after the title. Also, don’t bother trying to nest Abstract in anything else or vice versa. It won’t work. The Abstract environment is only useful in the article and report document classes. The book document classes ignores the Abstract completely, and it’s utterly silly to use Abstract in a letter document class.

The Abstract environment does several things for you. First, it puts the centered label “Abstract” above the text. The label and the text of the abstract are separated by some extra vertical space. Second, it typesets everything in a smaller font, just as you’d expect. Lastly, it adds a bit of extra vertical space between the abstract and the subsequent text. Well, that’s how it will appear on the LYX screen. The appearance in the output depends on the used article or report class.

Starting a new paragraph by hitting Return does not reset the paragraph environment. The new paragraph will still be in the Abstract environment. So, you will have to change the paragraph environment yourself when you finish entering the abstract of your document.

We’d love to give you directly an example of the Abstract environment, but since this document is in the “book” class, we can’t do this. We inserted it therefore as figure 3.1. If you’ve never heard of an “abstract” before, you can safely ignore this environment.
3. \LaTeX{} Basics

3.3.8.2. Bibliography

The Bibliography environment is used to list references. Technically, you could use this environment anywhere, but you really should only use it at the end of the document. Nesting Bibliography in anything else or vice versa won’t work.

When you first open a Bibliography environment, \LaTeX{} adds a large vertical space, followed by the heading “Bibliography” or “References,” depending on the document class. The heading is in a large boldface font. Each paragraph of the Bibliography environment is a bibliography entry. Thus, hitting Return does not reset the paragraph environment. Each new paragraph is still in the Bibliography environment.

There is another, usually better way to include references in your document by using a Bib\TeX{} database. For more information on that, and for a detailed description of \LaTeX{}‘s bibliography handling, have a look at in section 6.5.

3.3.9. \LaTeX{}-Code

The \LaTeX{}-Code environment is another \LaTeX{} extension. It typesets text in a typewriter-style font. It also treats the Space key as a fixed whitespace; this is the only case in which you can type multiple whitespaces in \LaTeX{}. If you need to insert blank lines, you’ll still need to use Ctrl+Return (the break-line function). Return breaks paragraphs. Note, however, that Return does not reset the paragraph environment. So, when you finish using the \LaTeX{}-Code environment, you’ll need to change the paragraph environment yourself. Also, you can nest the \LaTeX{}-Code environment inside of others.

There are a few quirks with this environment:

- You can’t use Ctrl+Return at the beginning of a new paragraph (i.e. you can’t follow Return with a Ctrl+Return).

- You can’t follow a Ctrl+Return with a Space.
  - Use a Return to begin a new paragraph, then you can use a Space.
  - Or: use Ctrl+Space instead.

- You can’t have an empty paragraph or an empty line. You must put at least one Space in any line you want blank. Otherwise, \LaTeX{} generates errors.

- You cannot get the typewriter double quotes by typing " since that will insert real quotes. You get the typewriter double quotes with Ctrl+".

Here’s an example:

```latex
#include <stdio.h>

int main(void)
```

\footnote{In the \LaTeX{}-Code environment, the Space key is treated as a Protected Blank instead of an end-of-word marker.}
3.4. Nesting Environments

3.4.1. Introduction

LyX treats text as a unified block with a particular context and specific properties. This allows you to create blocks that inherit some of the properties of another block. For example, you have three main points in an outline, but point #2 also has two subpoints. In other words, you have a list inside of another list, with the inner list “attached” to item #2:

1. one
2. two
   a) sublist – item #1
   b) sublist – item #2
3. three

You put a list inside a list by nesting one list inside the other. Nesting an environment is quite simple: Select Increase List Depth or Decrease List Depth from the Edit menu to change the nesting depth of the current paragraph (the status bar will tell you how far you are nested). Instead of the menu, you can also use the toolbar buttons \( \text{Increase List Depth} \) and \( \text{Decrease List Depth} \) or the convenient key bindings Tab and Shift-Tab or Alt+Shift+Right and Alt+Shift+Left to change the nesting level. The change will work on the current selection if you have made one (allowing you to change the nesting of several paragraphs at once), or the current paragraph.

Note that LyX only changes the nesting depth if it can. If it’s invalid to do so, nothing happens if you try to change the depth. Additionally, if you change the depth of one paragraph, it affects the depth of every paragraph nested inside of it.

Nesting isn’t limited to lists. In LyX, you can nest just about anything inside anything else, as you’re about to find out. This is the real power of nesting paragraph environments.
3. **LyX Basics**

### 3.4.2. What You Can and Can’t Nest

Before we fire a list of paragraph environments at you, we need to tell you a little bit more about how nesting works.

The question if nesting a paragraph environment is possible, is a bit more complicated than a simple yes or no. There are three types of paragraph environments:

- **Completely unnestable**
- **Fully nestable**, you can nest them inside things and you can also nest other things inside them.
- **A third type**, you can nest them into other environments, but you can’t nest anything into them.

Here’s a list of the three types of nesting behavior, and which paragraph environments have them:

**Unnestable**  Can’t nest them. Can’t nest into them.

- Bibliography
- Abstract
- Title
- Author
- Date

**Fully Nestable**  You can nest them. You can nest other things into them.

- Verse
- Quote
- Quotation
- Itemize
- Enumerate
- Description
- List
- LyX-Code

**Nestable-Inside**  You can nest them inside other things. You can’t nest anything into them.

- Standard
- Part
- Chapter
3.4. Nesting Environments

- Section
- Subsection
- Subsubsection
- Paragraph
- Subparagraph
- Part*
- Chapter*
- Section*
- Subsection*
- Subsubsection*
- Right Address
- Address

**Note:** Although it is possible to nest numbered section headings like Chapter, Section, etc. to e.g. lists, it is highly recommended not to do this because the aim is to create well structured documents following typesetting guidelines whereas nested section headings violate this.

### 3.4.3. Nesting Other Things: Tables, Math, Floats, etc.

There are several things that aren’t paragraph environments, but which are affected by nesting anyhow. They are:

- equations
- tables
- figures

(**Note:** Figures and tables in **Floats** are not affected by this. Have a look at section 4.6 for more information about **Floats**.)

LyX can treat these three objects as either a word or as a paragraph. If a figure, table, or an equation is inline, it goes wherever the paragraph it is in goes.

On the other hand, if you have an equation, figure or table in a “paragraph” of its own, it behaves just like a “nestable-inside” paragraph environment. You can nest it into any environment, but you obviously can’t nest anything into it.

Here’s an example with a table:

1. Item One
   
   a) This is (a) and it’s nested.
b) This is (b). The table is actually nested inside (a).

2. Back out again.

If we hadn’t nested the table at all, the list would look like this:

1. Item One
   a) This is (a) and it’s nested.

1. This is (b). The table is not nested inside (a). In fact, it’s not nested at all.

2. Back out again.

Notice how item (b) is not only no longer nested, but is also the first item of a new list!

There’s another trap you can fall into: Nesting the table, but not going deep enough. LyX then turns anything after the table into a new sublist.

1. Item One
   a) This is (a) and it’s nested.

   a) This is (b). The table is actually nested inside Item One, but not inside (a).

2. Back out again.

As you can see, item (b) turned into the first item of a new list, but a new list inside item 1. The same thing would have happened to a figure or an equation. So, if you nest tables, figures or equations, make sure you go to the right depth!
3.4.4. Usage and General Features

Speaking of levels, \LaTeX{} can perform up to a six-fold nesting. In other words, “level #6” is the innermost possible depth. Here’s an example to illustrate what we mean:

1. level #1 – outermost
   a) level #2
      i. level #3
         A. level #4
            • level #5
               – level #6

There are two exceptions to the six-fold nesting limit, and you can see both of them in the example. Unlike the other fully-nestable environments, you can only perform a four-fold nesting with the \texttt{Enumerate} and \texttt{Itemize} environments. For example, if we tried to nest another \texttt{Enumerate} list inside item “A.”, we would get errors.

3.4.5. Some Examples

The best way to explain just what you can do with nesting is by illustration. We have several examples of nested environments. In them, we explain how we created the example, so that you can reproduce them.

3.4.5.1. Example 1: The Six-fold Way and Mixed Nesting

#1-a This is the outermost level. It’s a \texttt{List} environment.

#2-a This is level #2. We created it by using Alt+Return followed by Alt+Shift+Right.

#3-a This is level #3. This time, we just hit Return, then used Alt+Shift+Right twice in a row. We could have also created it the same way as we did the previous level, by hitting Alt+Return followed by Alt+Shift+Right.

This is actually a \texttt{Standard} environment, nested inside of “#3-a”. So, it’s at level #4. We did this by hitting Alt+Return, then Alt+Shift+Right, then changing the paragraph environment to \texttt{Standard}. Do this to create list items with more than one paragraph — it also works for the \texttt{Description}, \texttt{Enumerate}, and \texttt{Itemize} environments!

Here’s another \texttt{Standard} paragraph, also at level #4, made with just a Alt+Return.
3. \textit{LyX Basics}

\#4-a This is level \#4. We hit Alt+Return and changed the paragraph environment back to List. Remember — we can’t nest anything inside a \textit{Standard} environment, which is why we’re still at level \#4. However, we \textit{can} keep nesting things inside \#3-a.

\#5-a This is level \#5...

\#6-a ...and this is level \#6. By now, you should know how we made these two.

\#5-b Back to level \#5. Just hit Alt+Return followed by a Alt+Shift+Left.

\#4-b After another Alt+Return followed by a Alt+Shift+Left, we’re back at level \#4.

\#3-b Back to level \#3. By now it should be obvious how we did this.

\#2-b Back to level \#2.

\#1-b And last, back to the outermost level, \#1. After this sentence, we’ll hit Return and change the paragraph environment back to \textit{Standard} to end the list.

We could have also used the \textit{Description}, \textit{Quote}, \textit{Quotation}, or even the \textit{Verse} environment in place of the List environment. The example would have worked exactly the same.

\textbf{3.4.5.2. Example 2: Inheritance}

This is the \textit{LyX-Code} environment, at level \#1, the outermost level. Now we’ll hit Return, then Alt+Shift+Right, after which, we’ll change to the Enumerate environment.

1. This is the Enumerate environment, at level \#2.

2. Notice how the nested Enumerate not only inherits its margins from its parent environment (\textit{LyX-Code}), but also inherits its font and spacing!

We ended this example by hitting Return. After that, we needed to reset the paragraph environment to \textit{Standard} and reset the nesting depth by using Alt+Shift+Left once.

\textbf{3.4.5.3. Example 3: Labels, Levels, and the Enumerate and Itemize Environments}

1. This is level \#1, in an Enumerate paragraph environment. We’re actually going to nest a bunch of these.
3.4. Nesting Environments

a) This is level #2. We used Alt+Return followed by Alt+Shift+Right. Now, what happens if we nest an Itemize environment inside of this one? It will be at level #3, but what will its label be? An asterisk?

• No! It’s a bullet. This is the first Itemize environment, even though it’s at level #3. So, its label is a bullet. (We got here by using Alt+Return, then Alt+Shift+Right, then changing the environment to Itemize.)

    - Here’s level #4, produced using Alt+Return, then Alt+Shift+Right. We’ll do that again...
      i. ...to get to level #5. This time, however, we also changed the paragraph environment back to Enumerate. Notice the type of numbering, it is lowercase Roman, because we are in the thirdfold Enumerate environment (i.e. it is an Enumerate inside an Enumerate inside an Enumerate).
      ii. What happens if we don’t change the paragraph environment, but decrease the nesting depth? What type of numbering does LyX use?
      iii. Oh, as if you couldn’t guess by now, we’re just using Alt+Return to keep the current environment and depth but create a new item.
      iv. Let’s use Alt+Shift+Left to decrease the depth after the next Alt+Return.

        i. This is level #4. Look what type of label LyX is using!
        ii. This is level #3. Even though we’ve changed levels, LyX is still using a lowercase Roman numeral as the label. Why?

    b) Another Alt+Return Alt+Shift+Left sequence, and we’re back to level #2. This time, we not only changed the nesting depth, but we also moved back into the twofold-nested Enumerate environment.

2. The same thing happens if we do another Alt+Return Alt+Shift+Left sequence and return to level #1, the outermost level.

Lastly, we reset the environment to Standard. As you can see, the level number doesn’t correspond to what type of labeling LyX uses for the Enumerate and Itemize environments. The number of other Enumerate environments surrounding it determines what kind of label LyX uses for an Enumerate item. The same rule applies for the Itemize environment, as well.
3. LyX Basics

3.4.5.4. Example 4: Going Bonkers

1. We’re going to go totally nuts now. We won’t nest as deep as in the other examples, nor will we go into the same detail with how we did it. (level #1: Enumerate)

(Return, Alt+Shift+Right, Standard: level #2) We’ll stick an encapsulated description of how we created the example in parentheses someplace. For example, the two keybindings are how we changed the depth. The environment name is the name of the current environment. Either before or after this, we’ll put in the level.

2. (Return, Enumerate: level #1) This is the next item in the list.

Now we’ll add verse.
It will get much worse.
(Return, Alt+Shift+Right, Verse: level #2)
Fiddle dee, Fiddle doo.
Bippitey boppitey boo!
(Alt+Return)
Here comes a table:

<table>
<thead>
<tr>
<th>one-fish</th>
<th>two-fish</th>
</tr>
</thead>
<tbody>
<tr>
<td>red-fish</td>
<td>blue-fish</td>
</tr>
</tbody>
</table>

(Alt+Return, Table, Alt+Shift+Right 3 times, Alt+Return, Verse, Alt+Shift+Left)

3. (Return, Enumerate: level #1) This is another item. Note that selecting a Table resets the nesting depth to level #1, so we increased the nesting depth 3 times to put the table inside the Verse environment.

We’re now ending the Enumerate list and changing to Quotation. We’re still at level #1. We want to show you some of the things you can do by mixing environments. The next set of paragraphs is a “quoted letter.” We’ll nest both the Address and Right Address environments inside of this one, then use another nested Quotation for the letter body. We’ll use Alt+Return to preserve the depth. Remember that you need to use Ctrl+Return to create multiple lines inside the Address and Right Address environments. Here it goes:

1234 Nowhere Rd.
Moosegroin, MT 00100
9-6-96

Dear Mr. Fizlewitz:
3.5. Spacing, pagination and line breaks

We regret to inform you that we cannot fill your order for 50 L of compressed methane gas due to circumstances beyond our control. Unfortunately, several of our cows have mysteriously exploded, creating a backlog in our orders for methane. We will place your name on the waiting list and try to fill your order as soon as possible. In the meantime, we thank you for your patience.

We do, however, now have a special on beef. If you are interested, please return the enclosed pricing and order form with your order, along with payment.

We thank you again for your patience.

Sincerely,
Bill Hick

That ends that example!

As you can see, nesting environments in LYX gives you a lot of power with just a few keystrokes. We could have easily nested an Itemize list inside of a Quotation or Quote, or put a Quote inside of an Itemize list. You have a huge variety of options at your disposal.

3.5. Spacing, pagination and line breaks

What is a space? While you might be used to pressing the space key anytime you want to separate two words in ordinary word processors, LYX offers you more spaces: Spaces of different width and spaces which can or cannot be broken at the end of a line. The following sections will show you some examples where those spaces are useful.

3.5.1. Protected Space

The protected space: It is used to tell LYX (and \LaTeX) not to break the line at that point. This may be necessary to avoid unlucky linebreaks, like in:

Further documentation is given in section 6.5.

Obviously, it would be a good thing to put a protected space between “section” and “6.5”. A protected space is set with Insert>Formatting>Protected Space (shortcut Ctrl+Space).

3.5.2. Horizontal Space

All horizontal spaces can be inserted with the menu Insert>Formatting>Horizontal Space. The length units are listed in Appendix D.
3.5.2.1. Inter-word Space

Some languages (e.g. English) have the typographical convention to add extra space after an end-of-sentence punctuation mark, and LyX honors those conventions (see section 3.9.3.1). Sometimes, you want a normal space nevertheless. In this case, insert an inter-word space (shortcut Ctrl+Alt+Space).

3.5.2.2. Thin Space

A “thin space” is a blank which has half the size of a normal space (and it is also “protected”). The typographical conventions in a lot of languages propose the use of thin spaces in cases where normal spaces would be too wide, for instance inside abbreviations:

D. E. Knuth has developed our beloved typesetting program.

or between values and units. Compare for example this:
10 kg (thin space)
10 kg (normal space)

You can insert thin spaces also with the menu Insert ➤ Formatting ➤ Thin Space (shortcut Ctrl+Shift+Space).

3.5.2.3. More Spaces

You can furthermore insert the following space types:

- **Negative thin space** A line with a $\rightarrow\leftarrow$ Negative thin space between the arrows.
- **Enspace (0.5 em)** A line with a $\rightarrow\leftarrow$ Enspace (0.5 em) space between the arrows.
- **Quad (1 em)** A line with a $\rightarrow\leftarrow$ Quad (1 em) space between the arrows.
- **QQuad (2 em)** A line with a $\rightarrow\leftarrow$ QQuad (2 em) space between the arrows.
- **Custom space** A line with $\rightarrow\leftarrow$ 2 cm space between the arrows.

Table 3.1 lists the different space sizes.

3.5.2.4. Horizontal Fills

Horizontal fills (HFills) are a special LyX feature for adding extra space in a uniform fashion. An HFill is actually a variable length space, whose length always equals the remaining space between the left and right margins. If there is more than one HFill on a line, they divide the available space equally between themselves.

Here a few examples what you can do with them:

<table>
<thead>
<tr>
<th>This is on the left side</th>
<th>This is on the right</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td>Middle</td>
</tr>
<tr>
<td>Left</td>
<td>1/3 Left</td>
</tr>
</tbody>
</table>
3.5. Spacing, pagination and line breaks

Table 3.1.: Width of the different horizontal spaces.

<table>
<thead>
<tr>
<th>Space</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>1/3\text{em}</td>
</tr>
<tr>
<td>Protected</td>
<td>1/3\text{em}</td>
</tr>
<tr>
<td>Thin</td>
<td>1/6\text{em}</td>
</tr>
<tr>
<td>Negative thin</td>
<td>-1/6\text{em}</td>
</tr>
<tr>
<td>Enspace (0.5\text{em})</td>
<td>0.5\text{em}</td>
</tr>
<tr>
<td>Quad (1\text{em})</td>
<td>1\text{em}</td>
</tr>
<tr>
<td>QQuad (2\text{em})</td>
<td>2\text{em}</td>
</tr>
</tbody>
</table>

That was an example in the \texttt{Quote} environment. Here→ ← is one in a standard paragraph. It may or may not be apparent in the printed text, but it is sitting in-between the two arrows.

HFills can be made visible when you choose one of the Fill Pattern in the space dialog: The following patterns are available:

- Dots: ............................................................
- Rule: ............................................................
- Left arrow: ←..................................................
- Right arrow: ..................................................
- Up brace: ....................................................
- Down brace: ...............................................  

\textbf{Note:} If an HFill is at the beginning of a line, and not in the first line in a paragraph, LYX ignores it. This prevents HFills from accidentally being wrapped onto a new line. If you need space in this case anyway, set the \texttt{Protect} option in the space dialog.

3.5.2.5. Phantom Space

Sometimes you want to insert space with exactly the length of a phrase. E.g. you want to create the following multiple choice question:

What is correct English?:

Mr. Edge would have been jumps the gun.

has to be jumped

jumps

So that the choices appear exactly after the phrase “Mr. Edge “. To get this, you can use the \texttt{\LaTeX}-command \texttt{\phantom} in \LaTeX\ Code\textsuperscript{4}. In our case write the command \texttt{\phantom{Mr. Edge }} (note the space after “Edge”) at the beginning of the last two lines. The command prints out the phrase within the the two braces, but invisible. That is why it is named “phantom”.\textsuperscript{5}

\textsuperscript{4}See sec. 6.10 for more information about \LaTeX-Code.

\textsuperscript{5}There exists also the commands \texttt{\hphantom} and \texttt{\vphantom}, but this too special for the LYX
3.5.3. Vertical Space

To add extra vertical space above or below a paragraph, use the Insert $\triangleright$ Formatting $\triangleright$ Vertical Space dialog. There you find the following sizes:

- **SmallSkip**, **MedSkip** and **BigSkip** are \LaTeX\ sizes which depend on the font size of the document. **DefSkip** is the skip adjusted in the dialog Document $\triangleright$ Settings for the paragraph separation. If you use indentation to separate paragraphs DefSkip is equal to MedSkip.

**VFill** is a variable space, set so that the space is maximal within one page. An example: You have only two short paragraphs on one page with a vfill between them. Then the first paragraph is placed at the top of the page and the second one at the bottom, because the space between them is then maximal. **VFills** work like **HFills**: they fill the remaining vertical space on a page with blank space.\(^ 6\) If there are several **VFills** on a page, they divide the remaining vertical space equally between themselves. You can therefore use **VFills** to center text on a page, or even place text 2/3 down a page.

**Custom** are custom spaces in units explained in Appendix D.

**Note:** When the extra vertical space would be in the output at the top/bottom of a page, the space is only added if you have also checked the option **Protect**.

3.5.4. Paragraph Alignment

You can change the paragraph alignment with the Edit $\triangleright$ Paragraph Settings dialog. There are five possibilities:

- **Justified** (shortcut Alt+A J)
- **Left** (Alt+A L)
- **Right** (Alt+A R)
- **Center** (Alt+A C)
- **Default** (Alt+A E)

The default in most cases is justified alignment, in which the inter-word spacing is variable and each line of a paragraph fills the region between the left and right margins. The other three alignments should be self-explanatory, and look like this:

```
This paragraph is right aligned,
```

```
this one is centered,
```

```
this one is left aligned.
```

\(^1\)HFills are described in section 3.5.2.
3.5. Forced Page Breaks

If you don’t like the way \LaTeX{} does the pagebreaks in your document, you can force a page break where you want one. Normally this will not be necessary, because \LaTeX{} is good at page breaking. Only if you use a lot of \texttt{Floats}, \LaTeX{}’s page breaking algorithm can fail.

We recommend not to use forced pagebreaks until the text is finished and until you have checked in the preview to see if you \textit{really} have to change the page breaking.

There are two types of pagebreaks: One that ends the page without any special action. This can be inserted above or below a paragraph via the menu \texttt{Insert} \texttt{\textbackslash\hspace{1pt} Formatting} \texttt{\textbackslash\hspace{1pt} New Page}. The second type, that is inserted via the menu \texttt{Insert} \texttt{\textbackslash\hspace{1pt} Formatting} \texttt{\textbackslash\hspace{1pt} Page Break}, ends a page but stretches the content of the page, so that it fills out the complete page. This type is useful to avoid whitespace when a page break produces a page on which only the last few lines are absent.

You might try to use a page break to ensure that a figure or table appears at the top of a page. This is, of course, the wrong way to do it. LyX gives you a way of automatically ensuring that your figures and tables appear at the top of a page (or the bottom, or on their own page) without having to worry about what precedes or follows your figure or table. See chapter 4 to learn more about \texttt{Floats}.

3.5.5.1. Clear Page Breaks

Rather than forced pagebreaks where the content behind the break is placed directly on the next page, you can also clear pages while breaking them. That means that the current paragraph is terminated and everything, including unprocessed floats, from the earlier part of the document are placed behind it, if necessary by adding pages.

You can insert a clear page break with the menu \texttt{Insert} \texttt{\textbackslash\hspace{1pt} Formatting} \texttt{\textbackslash\hspace{1pt} Clear Page}. When you have a two-sided document like a book, you can use the menu \texttt{Insert} \texttt{\textbackslash\hspace{1pt} Formatting} \texttt{\textbackslash\hspace{1pt} Clear Double Page} to insert a clear page break that assures that the next page is a right-hand page (odd-numbered), if necessary by adding a page.

3.5.6. Forced Line Breaks

Similar to page breaks there are two types of line breaks: One that simply breaks the line. You can force this line break within a paragraph by selecting \texttt{Insert} \texttt{\textbackslash\hspace{1pt} Formatting} \texttt{\textbackslash\hspace{1pt} Ragged Line Break} or with Ctrl+Return. Another type that is inserted via the menu \texttt{Insert} \texttt{\textbackslash\hspace{1pt} Formatting} \texttt{\textbackslash\hspace{1pt} Justified Line Break} breaks the line and stretches it so that it fills out the whole space between the page margins. This is necessary to avoid “fringes” in justified paragraphs due to whitespace introduced by line breaks.

You shouldn’t use forced line breaks to correct \LaTeX{}’s line breaking, as \LaTeX{} is very good at line breaking. There are, however, a number of situations where it is necessary to actively set a line break, e.g. in a poem or for an address (see sections 3.3.5.1, 3.3.5.2 and 3.3.7.2).
3.5.7. Horizontal Lines

In the dialog Insert $\rightarrow$ Formatting $\rightarrow$ Horizontal Line you can insert horizontal lines that span the whole width of the document or column.

3.6. Characters and Symbols

You can directly enter all characters that are available on your keyboard. You can also use special keyboard maps to be able to enter e.g. characters needed for French with an English keyboard. See section 6.15.2 for information on how this is done.

For the case where you need a character that is not on your keyboard, you can use the Symbols dialog via the menu Insert $\rightarrow$ Special Character $\rightarrow$ Symbols.

Note: Maybe not all symbols inserted with the symbols dialog can be displayed when you are using a special screen font in LyX’s preferences. But the inserted symbols will in every case be displayed in the output.

3.7. Fonts and Text Styles

3.7.1. Font Types

There are two types of fonts:

Vector fonts are fonts, built from outlines of the single glyphs (i.e. characters) in the font. This means that each glyph is defined using mathematical curves that are well suited for scaling to any requested size. This mathematical definition is interpreted by the font renderer and the curve is filled out with pixels according to the size and glyph. This means that outline fonts will look pretty good in all sizes. Only at very small sizes where each pixel has to be very carefully computed to provide a good image, it might be hard to provide a good rendering. That could mean that one only needs to define one font size and scale them. But to achieve a better quality, many fonts define several font sizes. That improves the appearance because you need more details at large font sizes than at small ones.

The font types TrueType, OpenType, and Type 1 are vector fonts.

Bitmap fonts on the other hand, are defined by bitmap graphics from the start, so they will look good at all the sizes they are meant for. However, they don’t scale well, because in order to scale a glyph, each pixel is enlarged into several pixels. It is the same effect that happens if you try to enlarge a picture in a picture manipulation program. In order to relieve this effect, bitmap fonts are typically provided in several fixed sizes typically from around 8 pixels high up to 34 pixels or so high in steps according to what is believed to be useful. The
advantage of bitmap fonts is that no complicated computations are necessary to display each glyph, so bitmap fonts are thus faster displayed than scalable fonts. The disadvantage is that sizes that don’t exist as fixed versions have to be scaled by doubling pixels, and thus look bad.

Bitmap fonts are named Type 3 in PostScript- and PDF-documents.

The result of all this, is that bitmap fonts are best for the size they are designed for, while scalable fonts are good for nearly all sizes. So one need fewer font size definitions for scalable fonts. That’s the reason why nearly all text rendering and typesetting programs use scalable fonts.

To test which fonts are used in a PDF-document, you can have a look into its document properties.

Many modern typesetting and markup languages have begun to move towards specifying character styles rather than specifying a particular font. For example, instead of changing to an italicized version of the current font to emphasize text, you use an “emphasized style” instead. This concept fits in perfectly with \LaTeX. In \LaTeX, you do things based on contexts, rather than focusing on typesetting details.

3.7.2. Document Font and Font size

You can set the document fonts in the Document > Settings dialog. There you can specify which font should be used for the three different font shapes roman (serif), sans serif, and typewriter and its scalings.

The possible options for the font include default and a list of fonts available on your system. default uses the standard \TeX fonts, known as “Computer Modern” (cm) or “European Computer Modern” (ec).

As cm and ec are bitmap fonts, they often looks pixeled in PDF output, especially when you read the PDF in a zoomed size.\(^7\) To get rid of pixeled fonts, you have to use a vector font. There are three ways to use one:

- One way is to use the AE fonts. AE is a virtual font. Virtual means that it “steals” outline cm-glyphs from other fonts. This has the disadvantage that some characters are missing, like the French guillemets (“«” and “»”)\(^8\) and that accented characters are not one glyph, they are build of two characters, the accent and the letter. Therefore you can’t search in documents using the AE fonts for words with accented characters. If you search for example for the French word “rêve” in a PDF, you won’t get any result, because the PDF-viewer searches for the glyph “ è ” and not for the glyph “ e + ` ”.

\(^7\)This problem doesn’t appear if you read PDFs in Adobe Reader version 6 or later, because this program includes a special bitmap font renderer.

\(^8\)Loading the \LaTeX-package aeguill with the document preamble line
\begin{verbatim}
\usepackage[ec]{aeguill}
\end{verbatim}
will fix the guillemet problem.
Another possibility is to use three different outline fonts:\footnote{Other fonts, like \texttt{Latin Modern} or \texttt{Computer Modern}, consist of these three main font types \texttt{sans serif}, typewriter, and serif.} \texttt{Times Roman} as roman font, \texttt{Helvetica} scaled to 92 or 95\% as sans serif font, and \texttt{courier} as typewriter font.

The differences between roman, \texttt{sans serif}, and \texttt{typewriter} fonts are explained in section 3.7.4.

The font \texttt{Times Roman} was originally designed for newspapers. That means its glyphs are smaller than the ones from other fonts to fit into the small newspaper columns. Therefore \texttt{Times Roman} is not the optimal choice for larger documents like books.

The best solution is to use the \texttt{Latin Modern (lm)} fonts. These fonts were developed for the \LaTeX{} community to replace \texttt{cm} as the default font. In most cases they look the same as \texttt{cm}\footnote{One difference is improved kerning for the \texttt{lm} fonts.}, but \texttt{lm} covers a vast number of characters over the other two families \texttt{cm} and \texttt{ec}.

For the font size there are four possible values: \texttt{default}, 10, 11, and 12. The size of \texttt{default} depends on your \LaTeX{}-system, normally it is equal to the font size 10.

The font sizes are the base size. That means that LyX scales all other possible font sizes (such as those used in footnotes, super-, and subscripts) by this value. You can fine-tune the font size of text parts via the Text Style dialog if needed. The possible font sizes for text parts are explained in section 3.7.4.

The field CJK allows users of the languages Chinese, Japanese, Korean (CJK) to specify a font to display the script characters.\footnote{The font will be the argument for the commands of the \LaTeX{}-package CJK. So this has no effect for the document language \texttt{Japanese} that doesn't use CJK.}

Note: When you choose a new font or font size, LyX does \textit{not} change the screen font! You will only see a difference in the printed output; this is part of the WYSIWYM concept. LyX’s screen fonts can be adjusted in the Tools >> Preferences dialog, see section C.1.2.

### 3.7.3. Using Different Character Styles

As we’ve already seen, LyX automatically changes the character style for certain paragraph environments. LyX supports two character styles, Emphasized and Noun. You can activate both of these styles via keybindings, the menus, and the toolbar.

To activate the Noun style, do one of the following:

- click on the toolbar button
- use the key binding Alt+C C
3.7. Fonts and Text Styles

These commands are all toggles. That is, if Noun style is already active, they deac-
tivate it.

One typically uses the Noun style for proper names. For example: “MATTHIAS
ETTRICH is the original author of L\TeX.”

A more widely used character style is the Emphasized style. You can activate (or
deactivate — it’s also a toggle) the Emphasized style by:

- clicking on the toolbar button $E$
- using the keybindings Ctrl+E

Normally the Emphasized style is equivalent to an italic font but some document
classes or E\TeX-packages use a different font.

We’ve been using the Emphasized style all over the place in this document. Here’s
one more example:

*Don’t overuse character styles!*

It’s also a warning in addition to an example. One’s writing should parallel ordinary
conversation. Since we don’t all constantly scream at each other, we should also avoid
the common tendency to overuse character style.

You can always reset to the default font using the key binding Alt+C Space or the
dialog Edit $\rightarrow$ Text Style.

3.7.4. Fine-Tuning with the Text Style dialog

There are always occasions when you will need to do some fine-tuning, so L\TeX gives
you a way to create a custom character style. For example, an academic journal
or a corporation may have a style sheet requiring a sans-serif font be used in certain
situations. Also, writers sometimes use a different font to offset a character’s thoughts
from ordinary dialog.

Before we document how to use custom character style, we want to issue a warning
yet again: Don’t overuse character styles!

Documents that overuse different fonts and sizes are not well readable and tend to
look as if someone has knocked huge holes in them.

To use custom character styles, open the Edit $\rightarrow$ Text Style dialog. There are several
boxes on this dialog, each corresponding to a different font property which you can
choose. You can choose an option for one of these properties, or select No change,
which keeps the current state of that property. The item Reset will reset the property
to whatever is the default. You can use this to reset attributes across a bunch of
different paragraph environments in a snap.

The font properties, and their options (in addition to No change and Reset) are:

Family The “overall look” of the font. The possible options are:
3. **LyX Basics**

- **Roman** This is the Roman font family. Normally a serif font. It’s also the default family. (key binding Alt+C R)
- **Sans Serif** This is the Sans Serif font family. (key binding Alt+C S)
- **Typewriter** This is the Typewriter font family. (key binding Ctrl+Shift+P)

**Series**
- **Medium** This is the Medium font series. It’s also the default series.
- **Bold** This is the Bold font series. (key binding Ctrl+B)

**Shape**
- **Upright** This is the Upright font shape. It’s also the default shape.
- **Italic** This is the Italic font shape.
- **Slanted** This is the Slanted font shape (although it might not be visible in LyX, this is different from italic).
- **Small Caps** This is the Small caps font shape.

**Size**
- **Tiny** This is the “Tiny” font size. (key bindings Alt+S T, Alt+S 1)
- **Smallest** This is the “Smallest” font size. (key binding Alt+S 2)
- **Smaller** This is the “Smaller” font size. (key bindings Alt+S Shift+S, Alt+S 3)
- **Small** This is the “Small” font size. (key bindings Alt+S S, Alt+S 4)
- **Normal** This is the “Normal” font size. It’s also the default size. (key bindings Alt+S N, Alt+S 5)
- **Large** This is the “Large” font size. (key bindings Alt+S L, Alt+S 6)
- **Larger** This is the “Larger” font size. (key bindings Alt+S Shift+L, Alt+S 7)
- **Largest** This is the “Largest” font size. (key binding Alt+S 8)
- **Huge** This is the “Huge” font size. (key bindings Alt+S H, Alt+S 9)
- **Huger** This is the “Huger” font size. (key bindings Alt+S Shift+H, Alt+S 0)

We’ll warn you yet again: don’t go crazy with this feature. You should almost never need to change the font size. LyX automatically changes the font size for different paragraph environments — use that instead. This is here for fine-tuning only!
3.7. Fonts and Text Styles

Misc
Here you can change a few other things at the character level. Options are:

Emph  This is text with emphasize on. This might seem like the same as Italic, but it is actually a bit different. Emphasized is a logical attribute. That means every document class can define its own font used for emphasized text. Normally this font is equal to italic.

Underbar  This is text with Underbar on. (key binding Ctrl+U, Alt+C U)
Avoid using underbar if you can! It’s a hangover from the typewriter days, when you couldn’t change fonts. We no longer need to emphasize text by underscoring characters. It’s only included in \LaTeX{} because some people may need it in order to follow style sheets for journal submissions.

Noun  This is text with Noun on. Like Emph, this is a logical attribute. Normally it’s equivalent to Small Caps.

Color
You can adjust the color of the text with this control. Notice that not all DVI-viewers are able to display colors. Besides No color, which is the default “color” and means normally black, you can choose between Black, White, Red, Green, Blue, Cyan, Magenta and Yellow text.

Language
This is used to mark regions of text as having a different language from the language of the document. Text marked in this way will be underlined in blue to indicate the change (only within \LaTeX{}).

So you have a huge number of combinations to choose from. Once you’ve chosen a new character style via the Edit $\rightarrow$ Text Style dialog, the settings are saved. You can activate them using the toolbar button $\downarrow$. The button lets you toggle the state of your custom character style even when the dialog isn’t visible.

To completely reset the character style to the default, use Alt+C Space. If you want to toggle only those properties that you have just changed (suppose you just set the shape to “slanted” and the series to “bold”), set the Toggle all switch and press Apply.

You should also know something about the differences between the three main font types serif, sans serif, and typewriter:

- **Typewriter** is a so called “monospaced” font, that means every character has the same width, the “i” is as wide as the “m”. Here is an example
  
  typewriter text
  no typewriter text

- **Serif** fonts use characters with serifs. These are the small “appendices” at the ends of the strokes that form the character. The following example will show the difference:
3. **LyX Basics**

- text with serifs
- text without serifs

Serifs facilitate quick and easy reading. These fonts are therefore used as default (named *roman*).

- **Sans serif** don’t use serifs. This font type is therefore often used for headings and short texts. We use it in this document to highlight menu names.

We conclude with the same warning once again: Don’t overuse the fonts. They are, more often than not, a kludge and a bad substitute for good writing.

### 3.8. Printing and Previewing

#### 3.8.1. **Overview**

Now that we have covered some of the basic features of document preparation using LyX, you probably want to know how to print out your masterpiece. Before we tell you that, we want to give you a quick explanation of what goes on behind-the-scenes. We cover this information in much greater detail in the *Additional Features* manual as well.

LyX uses the program *LATEX* as its backend. *LATEX* is just a macro package for the *TEX* typesetting system, but to prevent confusion, we’ll only refer to *LATEX*. LyX is what you use to do your actual writing. Then, LyX calls *LATEX* to turn your writing into printable output. This happens in two stages:

1. First, LyX converts your document to a series of text commands for *LATEX*, generating a file with the extension, “*.tex”.

2. Next, *LATEX* uses the commands in the *.tex* file to produce printable output.

#### 3.8.2. **Output file formats**

##### 3.8.2.1. **ASCII**

This file type has the extension “*.txt”. It contains your document as plain text following the rules of the “American Standard Code for Information Interchange” (ASCII).

You can export your document to ASCII by the menu **File** > **Export** > **ASCII**.

##### 3.8.2.2. **LATEX**

This file type has the extension “*.tex” and contains all commands that are necessary for the *LATEX* program to process your document. If you know *LATEX*, you can use it to find out *LATEX*-Errors or to process it manually with console commands. The *LATEX*-file is automatically created in LyX’s temporary directory whenever you view or export your document.

You can export your document as *LATEX*-file using the menu **File** > **Export** > **LaTeX**.
3.8.2.3. DVI

This file type has the extension "*.dvi". It is called "device-independent" (DVI), because it is completely portable; you can move them from one machine to another without needing to do any sort of conversion. DVIs are used for quick previews and as pre-stage for other output formats, like PostScript.

DVI files don’t contain images, they only link them. So don’t forget to deliver the images together with your DVIs. Because the DVI-viewer has to convert the image in the background to make it visible when you scroll the DVI, this can slow down your computer when you view the DVI. So we recommend to use PDF for files with many images.

You can export your document to DVI by the menu File $\rightarrow$ Export $\rightarrow$ DVI.

3.8.2.4. PostScript

This file type has the extension "*.ps". PostScript was developed by the company Adobe as a printer language. The file contains therefore commands that the printer uses to print the file. PostScript can be seen as a "programming language"; you can calculate with it and draw diagrams and images.\footnote{When you are interested to learn more about this, have a look at the $\LaTeX$-package pstricks.} Due to this ability, the files are often bigger than PDFs.

PostScript can only contain images in the format "Encapsulated PostScript" (EPS, file extension "*.eps"). As LyX allows you to use any known image format in your document, it has to convert them in the background to EPS. If you have e.g. 50 images in your document, LyX has to do 50 conversions whenever you view or export your document. This will slow down your workflow with LyX drastically. So if you plan to use PostScript, you can insert your images directly as EPS to avoid this problem.

You can export your document to PostScript using the menu File $\rightarrow$ Export $\rightarrow$ PostScript.

3.8.2.5. PDF

This file type has the extension "*.pdf". The “Portable Document Format” (PDF) developed by Adobe was derived from PostScript. It is more compressed and it uses fewer commands than PostScript. As the name “portable” implies, it can be processed at any computer system and the printed output looks exactly the same.

PDF can contain images in its own PDF format and in the formats “Joint Photographic Experts Group” (JPG, file extension "*.jpg" or "*.jpeg") and “Portable Network Graphics” (PNG, file extension "*.png"). You can although use any other image format, because LyX converts them in the background to one of these formats. But as described in the section about PostScript, the image conversion will slow down your workflow. So we recommend to use images in one of the three mentioned formats.

You can export your document to PDF via the menu File $\rightarrow$ Export in three different ways:
3. \textit{LYX Basics}

PDF This uses the program \texttt{ps2pdf} that creates a PDF from a PostScript-version of your file. The PostScript-version is produced by the program \texttt{dvips} which uses a DVI-version as intermediate step. So this export variant consist of three conversions.

PDF (\texttt{dvipdfm}) This uses the program \texttt{dvipdfm} that converts your file in the background to DVI and in a second step to PDF.

PDF (\texttt{pdflatex}) This uses the program \texttt{pdflatex} that converts your file directly to PDF.

We recommend to use PDF (\texttt{pdflatex}) because \texttt{pdflatex} supports all the features of actual PDF-versions, is quick, stable, and works without problems. The program \texttt{dvipdfm} is no longer under development and therefore a bit outdated.

3.8.3. Previewing

To get a look at the final version of your document, with all of the pagebreaks in place, the footnotes correctly numbered, and so on, use the menu \textit{View} and choose a file type. A viewing program will pop up showing the output. For \textit{View} \textit{DVI} you can use the toolbar button \texttt{[dvi]} (shortcut Ctrl+D), for \textit{View} \textit{PDF (pdflatex)} the toolbar button \texttt{[pdf]}, and for \textit{View} \textit{Postscript} the toolbar button \texttt{[ps]} (shortcut Ctrl+T).

If you have changed your document, you can refresh the output in the same viewer window using the menu \textit{View} \textit{Update}.

When you preview a file, the output file is only generated in LYX’s temporary directory. To have a real output, export your document.

3.8.4. Printing the File from within \textit{LYX}

Instead of exporting your file and then printing it, you can also print it directly from within \textit{LYX}. To print a file, select the menu \textit{File} \textit{Print} or click on the toolbar button \texttt{[print]. \textit{LYX} will internally call \texttt{TEX} to produce a DVI. This file is then processed by the program \texttt{dvips} to PostScript-file, which is finally printed using the program \texttt{Ghostscript}. Due to these steps in the background, this method is not the fastest.

You can choose to print only even-numbered or odd-numbered pages — this is useful for printing on two sides: You can re-insert the pages after printing one set to print on the other side. Some printers spit out pages face-up, others, face-down. By choosing a particular order to print in, you can take the entire stack of pages out of the printer without needing to reorder them.

You can set the parameters in the \textit{Print Destination} box as follows:
3.9. A few Words about Typography

3.9.1. Hyphens

In LYX, the “-” character comes in four lengths: the hyphen, the en dash, the em dash, and the minus sign:

<table>
<thead>
<tr>
<th>name</th>
<th>output</th>
<th>inserted with</th>
</tr>
</thead>
<tbody>
<tr>
<td>hyphen</td>
<td>-</td>
<td>“-” in text</td>
</tr>
<tr>
<td>en dash</td>
<td>–</td>
<td>Insert Special Character Symbols</td>
</tr>
<tr>
<td>em dash</td>
<td>—</td>
<td>Insert Special Character Symbols</td>
</tr>
<tr>
<td>minus sign</td>
<td>—</td>
<td>“-” in math mode</td>
</tr>
</tbody>
</table>

You can alternatively generate the en and em dash by inserting the “-” character multiple times in a row. They will automatically be converted to the appropriate length dash in the final output, but not in LYX. “- -” gives an en dash, “- - -” a em dash.

The three dash types are distinct from the minus sign, which appears in math mode and has a length of its own. Here are some examples of the “-” in use:

1. line- and page-breaks  \( (\text{hyphen}) \)
2. From A–Z \( (\text{en dash}) \)
3. Oh — there’s a dash. \( (\text{em dash}) \)
4. \( x^2 - y^2 = z^2 \) \( (\text{minus sign}) \)

3.9.2. Hyphenation

Words aren’t hyphenated within LYX but automatically in the output. Hyphenation is done by the \LaTeX\-package \texttt{babel} following the rules of the document language\textsuperscript{14}.

\LaTeX\ hyphenates almost perfectly, it only has problems with text in the \texttt{typewriter} font and with unusual constructs, like “h3knix/m0n0wall”. If \LaTeX\ can’t break

\textsuperscript{13}Note that this printer name is for the program \texttt{dvips}. That means \texttt{dvips} has to be configured for this printer name. The default printer can be set in LYX’s preferences dialog, see section C.6.1.

\textsuperscript{14}For German readers: That’s one of the main differences between the languages German and German (new spelling) in the Document \texttt{Settings} dialog.
3. **LyX Basics**

a word correctly, you can set hyphenation points manually. This is done with the menu `Insert ➤ Formatting ➤ Hyphenation Point`. These extra hyphenation points are only recommendations to \TeX. If no hyphenation is necessary, \TeX will ignore them.

Sometimes you want to prevent words or constructs to be hyphenated. Imagine that you are describing keybindings/shortcuts in your document in the form “A-b c”. \TeX would then see the hyphen “-” as hyphenation possibility. Hyphenating at this point would look ugly. To prevent the shortcut from being hyphenated, you can put it into the argument of the \TeX-box-command `\mbox`, because text within \TeX-boxes can’t be hyphenated. As LyX doesn’t support `\mbox`, you have to use \TeX Code. The result looks in LyX like:

\begin{verbatim}
\mbox{A-b c }
\end{verbatim}

To learn more about \TeX Code, have a look at section 6.10.

### 3.9.3. Punctuation Marks

#### 3.9.3.1. Abbreviations and End of Sentence

When LyX calls \TeX to generate the final version of your document, \TeX automatically distinguishes between words, sentences, and abbreviations. \TeX then adds the “appropriate amount of space”. That means sentences get a little bit more space between the period and the next word. Abbreviations get the same amount of space after the period as a word uses.

Unfortunately, the algorithm for figuring out what’s an abbreviation does not work in all cases. If a “.” is at the end of a lowercase letter, it’s the end of a sentence; if it’s at the end of a capitalized letter, it’s an abbreviation.

Here are some examples of correct abbreviations and the end of a sentence:

- M. Butterfly
- Don’t worry. Be happy.

And here’s an example of the algorithm going wrong:

- e.g. this is too much space!
- This is I. It’s okay.

You won’t see anything wrong until you view a final version of your document.

To fix this problem, use one of the following:

1. Use an **Inter-word Space** after lowercase abbreviations (see section 3.5.2.1).
2. Use a **Thin Space** between two tokens of an abbreviation (see section 3.5.2.2).
3. Use an **End of sentence period** found under the `Insert ➤ Special Character` menu to force the use of inter-sentence spacing. This function is also bound to Ctrl+ for easy access.
3.9. A few Words about Typography

With the corrections, our earlier examples look like this:

- e.g. this is too much space!
- This is I. It’s okay.

Some languages don’t use extra spacing between sentences. If your language is such a language, you don’t need to worry, because \LaTeX{} will take care of this.

For those that do need to bother, there is help to catch those sneaky errors: Check out the Check \TeX{} feature described in section Checking \TeX{} of the Additional Features manual.

3.9.3.2. Quotes

\LaTeX{} usually sets quotes correctly. Specifically, it will use an opening quote at the beginning of quoted text, and use a closing quote at the end. For example, “open close”. The keyboard character, ", generates this automatically.

You can change the behavior of the " key using the submenu Language of the dialog Document \rightarrow Settings dialog.

You can also select quotes for different languages in the box Type option. There are six choices:

- "Text" Use quotes like this “double” or ‘single’
- "Text" Use quotes like "this” or ’this’
- „Text“ Use quotes like „this“ or ′this’
- „Text“ Use quotes like „this” or ′this’
- «Text» Use quotes like «this» or ‹this›
- »Text« Use quotes like »this« or ›this‹

These settings affect what character the " key produces.

3.9.4. Ligatures

It is standard typesetting practice to group certain letters together and print them as single characters. These groups are known as ligatures. Since \LaTeX{} knows about ligatures, your documents will contain them too in the output. Here are the standard ligatures:

- ff
- fi
- fl
3. *LyX* Basics

- \texttt{fi}
- \texttt{fl}

Some languages uses other ligatures if the document font supports them.

Sometimes, you don’t want a ligature in a word. While a ligature may be okay in the word, “graffiti,” it looks really weird in compound words, such as “cufflink” or the German “Dorffest.” To break a ligature, use \texttt{Insert} \texttt{Formatting} \texttt{Ligature Break}. This changes “cufflinks” to “cufflinks” and “Dorffest” to “Dorffest”.

### 3.9.5. *LyX*’s Proper Names

You will certainly have noticed that the word “\LaTeX” always appears with characters in different sizes and heights. \LaTeX{} is the name of the program used by \textit{LyX} and is therefore recognized as a proper name when you type it in \textit{LyX} as “\LaTeX”. Note the order of the upper-and lowercase letters! \textit{LyX} recognizes the following proper names:

\begin{itemize}
\item \texttt{LyX} The name of the game, write “LyX” to produce it.
\item \texttt{TeX} The program used by \LaTeX{}, write “TeX” to produce it.
\item \texttt{LaTeX} The program used by \textit{LyX}, write “LaTeX” to produce it.
\item \texttt{LaTeX2ε} The actual version of \LaTeX{}, write “LaTeX2e” to produce it.
\end{itemize}

You might wonder why the \LaTeX{}-version is “2ε”. It’s an old tradition in the \TeX{}-world to give programs geek version numbers. For example the version number of \TeX{} converges to the number π: The actual version is “\TeX-3.141592”, the previous one was “\TeX-3.14159”.

If you don’t want to use proper names, e.g. in section headings, you can insert two empty braces in \TeX{} Code in the word. This will look in \textit{LyX} like: \texttt{LaT \{} \texttt{eX}

For more about \TeX{} Code, look at section 6.10.

### 3.9.6. Units

Generally the space between units and the number is smaller than the normal space between two words. As you can see in the example below, it looks better when the space is smaller. To get such a “half space” for units use the menu \texttt{Insert} \texttt{Formatting} \texttt{Thin Space} (shortcut \texttt{Ctrl}+\texttt{Shift}+\texttt{Space}).

Here’s an example to show the differences:

- 24 kW·h space between number and unit
- 24 kW·h half space between number and unit
3.9. A few Words about Typography

3.9.7. Widows and Orphans

In the early days of word processors, page breaks went wherever the page happened to end. There was no regard for what was actually going on in the text. You may remember once printing out a document, only to find the heading for a new section printed at the very bottom of the page, the first line of a new paragraph all alone at the bottom of a page, or the last line of a paragraph at the top of a new page. These bits of text became known as *widows* and *orphans*.

Clearly, LyX can avoid breaking pages after a section heading. That’s part of the advantage of paragraph environments. But what about widows and orphans, where the page breaks leave one line of a paragraph all alone at the top or bottom of a page? There are rules built into LATEX governing page breaks, and some of those rules are there to specifically prevent widows and orphans. This is the advantage LyX has in using LATEX as its backend.

There’s no way we can go into how TEX and LATEX decide to break a page, or how you can tweak that behavior. Some LATEX books listed in the bibliography [such as [1] or [2]] may have more information. You will almost never need to worry about this, however.
3. \LaTeX Basics
4. Notes, Graphics, Tables, and Floats

The issues of this chapter are described in detail in the *Embedded Objects* manual. There you’ll also find tips and tricks for special cases.

4.1. Notes

`LYX` offers you a few types of notes to add to your document:

**`LYX Note`**: This note type is for internal notes that won’t appear in the output.

**`Comment`**: This note also doesn’t appear in the output but it appears as *LATEX*-comment, when you export the document to *LATEX* via the menu File $\rightarrow$ Export $\rightarrow$ *LATEX* (pdflatex) / (plain).

**`Greyed Out`**: This note will appear in the output as grey text.

```plaintext
This is text\(^1\) of a comment that appears in the output as grey text.
```

As you can see in the example, the first line of greyed out notes is a bit indented and greyed out notes can have footnotes.

Notes are inserted with the toolbar button $\Rightarrow$ or the menu Insert $\rightarrow$ Note. Right-click on the appearing note box to select the note type.

4.2. Footnotes

`LYX` uses boxes to display footnotes: When you insert a footnote using the menu Insert $\rightarrow$ Footnote or the toolbar button $\Rightarrow$, you’ll see the following box: $\Rightarrow$ This box is `LYX`’s representation of your footnote. If you left-click on the “foot” label, the box will be opened and you can enter the footnote text into it. Clicking on the

\(^1\)This is an example footnote within a greyed out note. In this document the greyed out note color is redefined to blue. How this can be done is explained in the *Embedded Objects* manual.
4. Notes, Graphics, Tables, and Floats

box label again, will close the box. If you want to turn existing text into a footnote, simply mark it and click on the footnote toolbar button.

Here’s an example footnote:\[2\] The footnote will appear in the output as a superscript number at the text position where the footnote box is placed. The footnote text is placed at the bottom of the current page. The footnote number is calculated by \LaTeX, the numbers are consecutive, no matter in which chapter the footnote is in. \LaTeX doesn’t support other numbering schemes yet, but you can get another scheme using special \LaTeX-commands. They are described in the \textit{Embedded Objects} manual.

4.3. Marginal Notes

Marginal notes look and behave just like footnotes in \LaTeX. When you insert a margin note via the menu Insert $\rightarrow$ Marginal Note or the toolbar button $\text{\textit{Marginal Note}}$, you’ll see a grey box with a red label “margin” appearing within your text. This box is \LaTeX’s representation of your marginal note.

At the side is an example marginal note. Marginal notes appear at the right side in single-sided documents. In double-sided documents they appear in the outer margin – left on even pages, right on odd pages.

4.4. Graphics and Images

To insert an image in your document, place the cursor at the text position you want and click on the toolbar icon $\text{\textit{Insert}}$ or select Insert $\rightarrow$ Graphics from the menu. Then a dialog will appear to choose the file to load.

This dialog has numerous mostly self-explanatory parameters. The Graphics tab allows you to choose your image file. The image can be transformed by setting a rotation angle and a scaling factor. The scaling units are explained in Appendix D.

In the tab Clipping it is possible to set image coordinates to adjust the height and width of the image in the output. The coordinates can also be calculated automatically by pressing the button Get from file. The option Clip to bounding box will only print the image region within the given coordinates. Normally you don’t need to take care about image coordinates and can ignore the tab Clipping.

In the \LaTeX and \LaTeX options tab \LaTeX experts can specify additional \LaTeX options. You can also specify here the appearance of the image inside \LaTeX. The option Draft mode has the effect that the image doesn’t appear in the output, only a frame with the image size is printed. The option Don’t unzip on export is explained in the \textit{Embedded Objects} manual in section Graphics Dialog.

\[2\]To close a footnote, click on the footnote box label.
The graphics dialog can be called at any time by clicking on an image. The image will appear in the output exactly at the position where it is in the text. This is an example image within a separate, horizontally centered paragraph:

If you need image captions and want to reference images, you have to put the image into a float, see section 4.6.1.1.

### 4.4.1. Image Formats

You can insert images in any known file format. But as we explained in section 3.8.2, every output document format allows only a few image formats. LyX uses therefore the program *Imagemagick* in the background to convert the images to the right format. To increase your workflow by avoiding these conversions in the background, you can use only the image formats listed in the subsections of section 3.8.2.

Similar to fonts there are two types of image formats:

**Bitmap images** consist of pixel values, often in a compressed form. They are therefore not fully scalable and look pixeled in large zooms. Well-known bitmap image formats are “Graphics Interchange Format” (GIF, file extension “.gif”), “Portable Network Graphics” (PNG, file extension “.png”), and “Joint Photographic Experts Group” (JPG, file extension “.jpg” or “.jpeg”).

**Scalable images** consist of vectors and can therefore be scaled to any size without data loss. The scaling ability is necessary if you want to create presentations, because presentations are always scaled by the beamer. Scaling is also useful for online documents to let the user zoom into diagrams. Scalable image formats can be “Scalable Vector Graphics” (SVG, file extension “.svg”), “Encapsulated PostScript” (EPS, file extension “.eps”), and “Portable Document Format” (PDF, file extension “.pdf”). We say it can be, because you can convert any bitmap image format to PDF or EPS and the result won’t be scalable. In this cases only a header with the image properties is added to the original image.

Normally one can’t convert a bitmap image into a scalable one, only vice versa.

---

3 In the case of PDF, the original image is additionally compressed.
4.4.2. Grouping of Image Settings

Each image can define a new group of image settings or join an existing group. Images within such a group share their settings, so adjusting one image of the group automatically also adjusts all other images of the group in the same way. So you can for example change the size for a bunch of images without the need to manually change each of them.

A new group can be set by entering a name in the **Group Name** field in the Graphics dialog. Joining an existing group can be done using the context menu of the image by checking the name of the desired group.

4.5. Tables

You can insert a table using either the toolbar button 📊 or the menu **Insert > Table**. A dialog will appear, asking you for the number of rows and columns. The default table has lines around any cell and the first row appears separated from the rest of the table. This separation appears due to a double line: The cells of the first row have a line below them and the cells of the second row have a line above them. Here’s an example table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.5.1. The Table dialog

You can alter a table by clicking on it with the right mouse button, which brings up the table dialog. Here you can adjust the settings of the cell and row/column respectively where the cursor is placed currently. Most of the dialog options also work on selections. This means that if you select more cells, columns or rows the action is done on all of your selection.

Additionally to the table dialog, the **table toolbar** helps you in setting table properties. It appears when the cursor is inside a table.

In the tab **Table Settings** of the table dialog you can set the alignment for the current row. If you add a row or column, it will be inserted right beside or below the current cell respectively. The vertical alignment of a column can only be adjusted when a column width is given. A given width will allow the cell to have line breaks and multiple paragraphs of text, see section 4.5.3.

You can mark multiple cells of one row as a multicolumn cell using the check box **Multicolumn**. This will merge the cells to one cell, spread over more than one column. Multicolumn cells are treated as own rows, so that the alignment, width, and border settings affect only the multicolumn cell. Here’s an example table with a multicolumn cell in the first row and one in the last row without the upper border:
At the moment LYX doesn’t support multirow cells. Adept users can declare special \LaTeX-arguments for the table. They are necessary for special table formatting, like for multirow cells, explained in the tables section of the Embedded Objects manual. You can also rotate the current cell or the whole table 90 degrees counterclockwise. These rotations are not visible in LYX but in the output.

**Note:** Most DVI-viewers are not able to display rotations.

The Borders tab allows you to add and delete border lines for the current row/column. The button Default adds lines for all cell borders.

### 4.5.2. Longtables

If the table is too long to fit on one page, you can use the option Use long table in the tab Longtable of the table dialog to split the table automatically over more pages. Doing this enables some check boxes and you can now define:

**Header:** The current row and all rows above, that don’t have any special options defined, are defined to be the header rows of all pages of the longtable; except for the first page, if First header is defined.

**First header:** The current row and all rows above, that don’t have any special options defined, are defined to be the header rows of the first page of the longtable.

**Footer:** The current row and all rows below, that don’t have any special options defined, are defined to be the footer rows of all pages of the longtable; except for the last page, if Last footer is defined.

**Last footer:** The current row and all rows below, that don’t have any special options defined, are defined to be the footer rows of the last page of the longtable.

**Caption:** The first row is reset as single column. You can now insert there the table caption via the menu Insert>Caption. More about longtable captions can be found in the Embedded Objects manual.

You can also specify a row where the table is split. If you set more than one option in the same table row, you should be aware of the fact that only the first one is used in the given table row. The others will then be defined as empty. In this context, first means first in this order: Footer, Last footer, Header, First header. See the following longtable to see how it works:
4. Notes, Graphics, Tables, and Floats

<table>
<thead>
<tr>
<th>NAME</th>
<th>TEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annovi</td>
<td>Silvia 111</td>
</tr>
<tr>
<td>Bertoli</td>
<td>Stefano 111</td>
</tr>
<tr>
<td>Bozzi</td>
<td>Walter 111</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maria 111</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maurizio 111</td>
</tr>
<tr>
<td>Cinquemani</td>
<td>Giusi 111</td>
</tr>
<tr>
<td>Colin</td>
<td>Bernard 111</td>
</tr>
<tr>
<td>Concli</td>
<td>Gianfranco 111</td>
</tr>
<tr>
<td>Dal Bosco</td>
<td>Carolina 111</td>
</tr>
<tr>
<td>Dalpiaz</td>
<td>Annamaria 111</td>
</tr>
<tr>
<td>Feliciello</td>
<td>Domenico 111</td>
</tr>
<tr>
<td>Focarelli</td>
<td>Paola 111</td>
</tr>
<tr>
<td>Galletti</td>
<td>Orestes 111</td>
</tr>
<tr>
<td>Gasparini</td>
<td>Franca 111</td>
</tr>
<tr>
<td>Rizzardi</td>
<td>Paola 111</td>
</tr>
<tr>
<td>Lassini</td>
<td>Giancarlo 111</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Luciano 111</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Valeriano 111</td>
</tr>
<tr>
<td>Meneguzzo</td>
<td>Roberto 111</td>
</tr>
<tr>
<td>Mezzadra</td>
<td>Roberto 111</td>
</tr>
<tr>
<td>Pirpaner</td>
<td>Erich 111</td>
</tr>
<tr>
<td>Pochiesa</td>
<td>Paolo 111, 222</td>
</tr>
<tr>
<td>Radina</td>
<td>Claudio 111</td>
</tr>
<tr>
<td>Stuffer</td>
<td>Oskar 111</td>
</tr>
<tr>
<td>Tacchelli</td>
<td>Ugo 111</td>
</tr>
<tr>
<td>Tezzele</td>
<td>Margit 111</td>
</tr>
<tr>
<td>Unterkalmsteiner</td>
<td>Frieda 111</td>
</tr>
<tr>
<td>Vieider</td>
<td>Hilde 111</td>
</tr>
<tr>
<td>Vigna</td>
<td>Jürgen 111</td>
</tr>
<tr>
<td>Weber</td>
<td>Maurizio 111</td>
</tr>
<tr>
<td>Winkler</td>
<td>Franz 111</td>
</tr>
</tbody>
</table>

Annovi | Silvia | 555 |
Bertoli | Stefano | 555 |
Bozzi | Walter | 555 |
Cachia | Maria | 555 |
Cachia | Maurizio | 555 |
Cinquemani | Giusi | 555 |
Colin | Bernard | 555 |
continue ...
4.5.3. Table Cells

A table cell can contain text, inline equations, a figure, or another table. All these kinds of objects can be placed in the same cell. Font sizes and shapes can also be altered. But you can’t put a special environment in a cell (like Section*, etc.), nor set spacing options etc. for the cell’s paragraph.

To have multi-line entries in table cells, you have to declare a fixed width for the column in the table dialog. Your text is then automatically split into multiple lines and the cell is enlarged vertically when the length of the text exceeds the given width. An example:

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
NAME & TEL. \\
\hline
Concli & Gianfranco & 555 \\
Dal Bosco & Carolina & 555 \\
Dalpiaz & Annamaria & 555 \\
Feliciello & Domenico & 555 \\
Focarelli & Paola & 555 \\
Galletti & Oreste & 555 \\
Gasparini & Franca & 555 \\
Rizzardi & Paola & 555 \\
Lassini & Giancarlo & 555 \\
Malfatti & Luciano & 555 \\
Malfatti & Valeriano & 555 \\
Meneguzzo & Roberto & 555 \\
Mezzadra & Roberto & 555 \\
Pirpamer & Erich & 555 \\
Pochiesa & Paolo & 555, 222 \\
Radina & Claudio & 555 \\
Stuffer & Oskar & 555 \\
Tacchelli & Ugo & 555 \\
Tezzele & Margit & 555 \\
Unterkalmsteiner & Frieda & 555 \\
Vieider & Hilde & 555 \\
Vigna & Jürgen & 999 \\
Weber & Maurizio & 555 \\
Winkler & Franz & 555 \\
\hline
\end{tabular}
\end{table}

End
4. Notes, Graphics, Tables, and Floats

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>This is a multi-line entry in a table.</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>This is longer now.</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>This is a multi-line entry in a table. This is longer now.</td>
<td>9</td>
</tr>
</tbody>
</table>

Cutting and pasting between tables and table cells works reasonably well. You can cut and paste even more than one row.\(^4\) Selection with the mouse or with Shift plus the arrow keys works as usual. You can also copy and paste the entire table as a single unit by starting the selection from outside the table.

### 4.6. Floats

A float is a block of text associated with some sort of label, which doesn’t have a fixed location. It can “float” forward or backward a page or two, to wherever it fits best. Footnotes and Margin Notes are also floats, because they can float to the next page when there are too many notes on the page.

Floats make it possible to get a high quality layout. Images and tables can be spread evenly over the pages to avoid whitespace and pages without text. As the floating often destroys the context between the text and the image/table, every float can be referenced in the text. Floats are therefore numbered. Referencing is described in section 6.1.

To insert a float, use the menu Insert>Floats. A box with a caption that has e.g. the label “Figure #:” (# is the actual number) will be inserted into your document. The label will automatically be translated to the document language in the output. After the label you can insert the caption text. The image or table is inserted above or below the caption in a separate paragraph within the float. To keep your \LaTeX\document readable, you can open and close the float box by left-clicking on the box label. A closed float box looks like this: \[\text{float: Figure}\] – a gray button with a red label.

It is recommended to insert floats as a separate paragraph to avoid possible \LaTeX\-errors that can occur when the surrounding text is specially formatted.

\(^4\)Note, that you cannot paste into a multicell selection because it would not be clear what to do when pasting a single word in a selected 2×3.
4.6. Floats

4.6.1. Float Types

4.6.1.1. Figure Floats

The menu Insert $\triangleright$ Float $\triangleright$ Figure inserts a float with the label “Figure #:”. Set the cursor above this label (or before it and press enter) and insert the image as described above to get the caption printed below the image. This is what we did for Figure 4.1. If you want the caption to be above the image, set the cursor at the end of the caption, press enter and insert the image. This was done in Figure 4.2.

This figure float also shows how to set a label and create a cross-reference to it. As described in section 6.1, you can simply insert a label in the caption using the menu Insert $\triangleright$ Label and refer to it using the menu Insert $\triangleright$ Cross-Reference. It is important to use references to figure floats, rather than using vague references like “the figure above”, because as $\LaTeX$ will reposition the floats in the final document; it might not be “above” at all.

Normally only one image is inserted in a figure float, but sometimes you might want to use two images with separate subcaptions. This can be done by inserting
4. Notes, Graphics, Tables, and Floats

(a) Undefinable  
(b) Platypus

Figure 4.3.: Two distorted images.

Table 4.2.: A table float.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>Mary</td>
<td>Ted</td>
</tr>
<tr>
<td>$\int x^2 dx$</td>
<td>$\begin{bmatrix} a &amp; b \ c &amp; d \end{bmatrix}$</td>
<td>$1 + 1 = 2$</td>
</tr>
</tbody>
</table>

image floats into existing image floats. Note that only the main caption of the float is added to the List of Figures. Figure 4.3 is an example of a figure float with two images set side by side. You can also set the images one below the other. Figure 4.3a and 4.3b are the subfigures.

Note that the caption is added to the List of Figures as described in section 6.2.2.

4.6.1.2. Table Floats

Table floats can be inserted using the menu Insert $\rightarrow$ Float $\rightarrow$ Table. They have the same properties as figure floats except for the different label. Table 4.2 is an example of a table float.

4.6.1.3. Algorithm Floats

This float type is inserted with the menu Insert $\rightarrow$ Float $\rightarrow$ Algorithm. It is used for program codes and descriptions of algorithms. A possible environment for algorithms is the LyX-Code, described in section 3.3.9.

Note: that the float label is not automatically translated into the document language. You have to do this manually by adding the following line
\floatname{algorithm}{your name}

to the document preamble (menu Document>Settings). your name is the word “algorithm” in your language.

4.6.1.4. Wrap Floats

This float type is used if you want to “wrap” text around a figure so that it only occupies some fraction of the column width. It can be inserted using the menu Insert>Float.Wrap Float if the \LaTeX-package wrapfig is installed.\footnote{Installing a \LaTeX-package is explained it in the \LaTeX Configuration manual.} The width and placement of the float is adjusted by right-clicking on the float box. Figure 4.4 is for example a figure wrap float with a width of 40col\%.\footnote{Available units are explained in Appendix D.} Some space was added under the caption to separate it better from the surrounding text.

Note: Wrap floats might be fragile! E.g. having a figure too close to the bottom of the page can mess things up so that the float doesn’t appear in the output or that it is placed over some other text. In general:

- Wrap floats should not be placed in paragraphs that run over a page break. That means that wrap floats should preferably be inserted in the exact place when the document is nearly ready and you are able to estimate where page breaks will appear.

- Wrap floats should either be placed in their own paragraph before the paragraph where they should wrap into, or within a paragraph.

- Wrap floats in consecutive paragraphs may cause trouble, so make sure that there is a text paragraph between them as separator.

- Wrap floats are not allowed in section headings or tables.

4.6.2. Rotated Floats

Especially for wide tables you might have floats rotated. To rotate a whole float including the caption, right-click on the float-box and use the option Rotate sideways. Rotated floats are always placed on their own pages (or columns, when you have a multi-column document). You can let them span several columns using the float settings option Span columns. Floats are rotated so that you can read them from the outside margin. Forcing the rotation direction is explained in the Embedded Objects manual.
4. Notes, Graphics, Tables, and Floats

Referencing rotated floats is the same like for normal floats; the caption format is also the same: Table 4.3 is an example of a rotated table float.

Note: Not all DVI-viewers are able to display rotated floats.

4.6.3. Float Placement

Right-clicking on a float-box opens a dialog where you can alter the placement options that \LaTeX uses for positioning the float.

The option \texttt{Span columns} is only useful for two-column documents: If you select it, the float will span both columns on the page instead of being confined to just one.

The option \texttt{Rotate sideways} is used to rotate floats, see section 4.6.2.

You can use one or more of the following options in the float dialog to set the placement for a particular float when you uncheck the option \texttt{Use default placement}:

\textbf{Here if possible:} try to place the float at the position where it is inserted

\textbf{Top of page:} try to place the float at the top of the current page

\textbf{Bottom of page:} try to place the float at the bottom of the current page

\textbf{Page of floats:} try to place the float at an own page

The order of the above options is \textit{always} used by \LaTeX. That means, if you use the default placement, \LaTeX will first try out \texttt{Here if possible}, then \texttt{Top of page}, and then the others. If you don’t use the default, \LaTeX will try only the checked options but in the same order. If none of the 4 placements are possible the procedure is internally repeated but it is tried to put the float on the following page.

By default, each option has its own rules:

\texttt{Top of page} only floats occupying less than 70% of the page can be placed at the top of a page

\texttt{Bottom of page}: only floats occupying less than 30% of the page can be placed at the bottom of a page.

\texttt{Page of floats}: only if more than 50% of the page are occupied by floats, several floats can be set together on a page.

If you don't like these rules, you can ignore them by using the additional option \texttt{Ignore \LaTeX rules}.

Sometimes you might need, under all circumstances, a float to be placed exactly at the position where it is inserted. For this case you can use the option \texttt{Here definitely}. Use this option very rarely and only if the document is nearly ready to be printed. Because the float is then no longer able to “float” when you change your document, this will often destroy the page layout.

There are no placement options for text wrap floats, because they are always surrounded by the text of a certain paragraph.

For more details about float placements, have a look at \LaTeX books like [1, 2, 3].
Table 4.3.: Rotated table

<table>
<thead>
<tr>
<th>test</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Notes, Graphics, Tables, and Floats

4.7. Minipages

\LaTeX{} provides a mechanism to produce essentially a page within a page, called minipage. Within a minipage, all the usual rules of indentation, line wrapping, etc. apply.

Minipages in \LaTeX{} have their own collapsible box inserted via the menu Insert\textgreater Box. Right-clicking on the box allows you to alter the width of the minipage and its alignment within the page.

This is a minipage. The text is set in an italic style. Minipages are often used for text in another language or text that needs another formatting.

If you place two minipages side-by-side, you can use Horizontal Fills as described in section 3.5.2:

This is a minipage with some stupid dummy text. This dummy text is used to increase the size of the minipage.

This is a minipage with some stupid dummy text. This dummy text is used to increase the size of the minipage.

When you right-click on a minipage box, you can change the box from a minipage to other box types. All box types and their settings are explained in detail in chapter Boxes of the Embedded Objects manual.
5. Mathematical Formulas

The issues of this chapter are described in detail in the *Math* manual. There you’ll also find tips and tricks for special cases.

5.1. Basic Math Editing

To create a math formula, you can just click on the toolbar icon $\frac{a+b}{c}$. That will create a little blue rectangle, with purple markers around its corners. That blue rectangle is the formula itself; the purple markers indicate what level of nesting within the formula you are at. You can also choose a particular formula type to insert via the Insert $\rightarrow$ Math menu.

Editing the parameters of a formula and adding math constructs can be done with the math toolbar, that appears when the cursor is in a formula.

There are two main types of formulas: Inline formulas appear within a text line, like this one:

This is a line with an inline formula $A = B$ in it.

Displayed formulas appear outside the text like as if they were in an own paragraph, like this one:

$$A = B$$

You can only number and reference displayed formulas.

LYX supports also many *\LaTeX* math commands. E.g. typing "\textbackslash alpha", followed by a space, in a formula will create the Greek letter $\alpha$. So typing commands might sometimes be faster than using the Math Panel.

5.1.1. Navigating in Formulas

The best control over the cursor position within an existing formula is achieved with the arrow keys. LYX uses small rectangles to indicate places where something can be inserted. The arrow keys can be used to navigate between parts of a formula. Pressing Space will leave a formula construct (a square root $\sqrt{2}$, or parentheses $(f)$, or a matrix $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$). Pressing Escape will leave the formula, placing the cursor after the formula. Tab can be used to move horizontally in a formula; for example, through the cells of a matrix or the positions in a multi-line equation.

Space, printed in this document as “␣”, seems to do nothing in a formula, since it does not add a space between characters, but it does exit a nested structure. For this
reason, you have to be careful about using \textit{Space}. For example, if you want $\sqrt{2x+1}$, type \texttt{\textbackslash sqrt\_2x+1} and not \texttt{\textbackslash sqrt\_2x\_+1}, since in the latter case only the $2x$ will be under the square root sign: $\sqrt{2x+1}$.

You can leave many parts of a formula, like this matrix, partially filled in, such as:

$$
\begin{pmatrix}
\lambda_1 \\
\vdots \\
\lambda_n
\end{pmatrix}
$$

If you leave a fraction only partially filled in, or a subscript with nothing in it, the results will be unpredictable, but most constructs don’t mind.

### 5.1.2. Selecting Text

You can select text within a formula in two different ways. Place the cursor at one end of the string of text you want, and press \texttt{Shift} and a cursor movement key to select text. It will be highlighted as with regular text selection. Alternatively, you can select text with the mouse in the usual way. That text can then be cut or copied, and then pasted within any formula, but not in a normal text region in \LaTeX{}.

### 5.1.3. Exponents and Subscripts

You can use the math panel to add super- or subscripts, but the much easier way is to use a command. To get $x^2$, type in a formula \texttt{x\textasciicircum{}2}. The final \textit{Space} puts the cursor back down on the base line of the expression. If you type \texttt{x\textasciicircum{}2y}, you will get $x^2y$, to get $x^2y$, type \texttt{x\textasciicircum{}2\_y}. If you use characters in the superscript, that could be accented with the hat “\textasciicircum{}”, you have to use an extra \textit{Space} to separate the hat and the character. E.g. if you want $x^a$, type \texttt{x\textasciicircum{}a}. Subscripts are similar: To get $a_1$, type \texttt{a\_1}.

### 5.1.4. Fractions

Create a fraction with either the command \texttt{\textbackslash frac} or using the icon \texttt{\frac{a}{b}} in the Math Panel. You will be presented with an empty fraction. The cursor is above the fraction line. To move it to the bottom, simply press \texttt{Down}. To move back up, press \texttt{Up}. Any math structure can be placed in a fraction, as this example shows:

$$
\frac{
\begin{pmatrix}
1 \\
2 \\
3 \\
4 \\
5
\end{pmatrix}
}{
2 \\
3
}
$$
5.1. Basic Math Editing

5.1.5. Roots

Roots can be created using the Math Panel button $\sqrt{\phantom{a}}$ or the commands $\sqrt{}$ or $\sqrt{\phantom{a}}$. With the command $\sqrt{}$ you can produce roots of higher orders, like cube roots, while $\sqrt{}$ produces always a square root.

5.1.6. Operators with Limits

Sum ($\sum$) and integral ($\int$) operators are very often decorated with limits. These limits can be entered in LyX by entering them as you would enter a super- or subscript, directly after the symbol. The sum operator will automatically place its “limits” over and under the symbol in displayed formulas, and on the side in inline formulas. Such as $\sum_{n=0}^{\infty} \frac{1}{n!} = e$, versus

$$\sum_{n=0}^{\infty} \frac{1}{n!} = e$$

Integral signs, however, will place the limits on the side in both formula types.

All operators with limits will be automatically re-sized when placed in display mode. The placement of the limits can be changed by placing the cursor directly behind the operator and hitting Alt+M L or using the menu Edit $\rightarrow$ Math $\rightarrow$ Change Limits Type.

Certain other mathematical expressions have this “moving limits” feature as addition, such as

$$\lim_{x \to \infty} f(x),$$

which will place the $x \to \infty$ underneath the “lim” in display mode. In inline formulas it looks like this: $\lim_{x \to \infty} f(x)$.

Note that the lim-function was entered as the function macro $\lim$. Have a look at section 5.1.9 for an explanation of function macros.

5.1.7. Math Symbols

Most math symbols can be found in the Math Panel under one of several categories; including Greek, Operators, Relations, Arrows. There are also the additional symbols provided by the American Mathematical Society (AMS).

If you know the \LaTeX-command for a construct or symbol you wish to use, you don’t have to use the Math Panel, but can type the command directly into the formula. LyX will convert it to the corresponding symbol or construct.

5.1.8. Altering Spacing

You may want to create spaces that differ from the standard spacing that \LaTeX provides. To do this, type Ctrl+Space or to use the Math Panel button $. This generates a small space, and shows a small marker on the screen. For example, the
Table 5.1.: Accent names and the corresponding commands.

<table>
<thead>
<tr>
<th>Name</th>
<th>Command</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>circumflex</td>
<td>\hat</td>
<td>ą</td>
</tr>
<tr>
<td>grave</td>
<td>\grave</td>
<td>ą</td>
</tr>
<tr>
<td>acute</td>
<td>\acute</td>
<td>ą</td>
</tr>
<tr>
<td>umlaut</td>
<td>\ddot</td>
<td>ą</td>
</tr>
<tr>
<td>tilde</td>
<td>\tilde</td>
<td>ą</td>
</tr>
<tr>
<td>dot</td>
<td>\dot</td>
<td>ą</td>
</tr>
<tr>
<td>breve</td>
<td>\breve</td>
<td>ą</td>
</tr>
<tr>
<td>caron</td>
<td>\check</td>
<td>ą</td>
</tr>
<tr>
<td>macron</td>
<td>\bar</td>
<td>ą</td>
</tr>
<tr>
<td>vector</td>
<td>\vec</td>
<td>ą</td>
</tr>
</tbody>
</table>

sequence a Ctrl+Space b: ab appears in \LaTeX\ as a\_b. You can change the space to different sizes when you set the cursor behind the space marker and hit space again several times. With every space hit the size will be changed. Some markers for the space size appear red in \LaTeX, because they are negative spaces. Here are two examples:

\begin{itemize}
  \item a Ctrl+Space b and 3×Space: ab
  \item a Ctrl+Space b and 5×Space: ab
\end{itemize}

5.1.9. Functions

The Math Panel contains under the button \[\exp\] a number of function macros, such as sin, lim, etc. (you can also insert them in a formula by typing \[\sin\] etc.). Standard mathematical practice is, that functions are printed upright to avoid confusions, because sin does normally mean s·i·n.

Using the function macros will also produce correct spacing around the function: a\sin x is different from asinx

For some mathematical objects, like the limes, the macro changes where subscripts are placed, as described in section 5.1.6.

5.1.10. Accents

In a formula you can insert accented characters in the same way as in text mode. This may depend on your keyboard, or the bindings file you use. You can also use \LaTeX\ commands to e.g. enter ą even if your keyboard doesn’t have hat-accents enabled. Our example is entered by typing \[\hat{a}\] in a formula. Table 5.1 shows the equivalences between the accent names and the commands.

You can choose one of the accents by selecting an item from the Frame decorations symbol set button in the math panel; this will apply to any selection you have
made within a formula too.

5.2. Brackets and Delimiters

There are several brackets available through \LaTeX. For most purposes, using just the keys \[\{\}\langle\rangle|<> should suffice. But if you want to surround a large structure, like a matrix or a fraction, or if you have several layers of brackets, it is better to use the math toolbar delimiter icon \(\)[]. For example, that’s how you would construct the brackets around a standard matrix such as:

\[
\begin{pmatrix}
1 & 2 \\
3 & 4
\end{pmatrix}
\]

and to make it easier to see the layers of parentheses as in:

\[
\frac{1}{1+\left(\frac{1}{1+\left(\frac{1}{1+x}\right)}\right)}
\]

The parentheses, and other brackets from that menu will automatically re-size to accommodate the size of what is inside.

To construct brackets click on the button for the bracket you want on the left side and right side. If you use the option \textit{Keep matched}, the selected bracket type will be used for the left and the right side. The selection will be shown below the button field. If you want one side to not have a bracket, use the blank button. It will appear in \LaTeX with a dotted line, but nothing will be printed.

If you want to place brackets around math structures, like a square root, you can do that by highlighting (selecting) the structure that is to go inside the brackets. Then choose the appropriate brackets for left and right and click on \textit{Insert}. The parentheses will be drawn around the selected structure.

5.3. Arrays and Multi-line Equations

Matrices are entered in \LaTeX using the \textit{Math Panel} matrix button \(\), \(\). It will open a dialog for you to choose the number of rows/columns. Here is an example:

\[
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{pmatrix}
\]

The parentheses aren’t automatic, but you can add them as described in section 5.2. When you construct the matrix, you can decide whether the column entries will be left-, right-, or center-justified. This alignment is set in the box \textit{Horizontal} with the
5. Mathematical Formulas

letters “l”, “r”, and “c”. LyX proposes a “c” for every column as default. For example, the sequence “lrc” means that the first column will be left-justified, the second will be centered, and the third column will be right-justified, because each letter corresponds to the relevant column. The result will look like this:

<table>
<thead>
<tr>
<th>this column</th>
<th>this column</th>
</tr>
</thead>
<tbody>
<tr>
<td>has left alignment</td>
<td>has right alignment</td>
</tr>
</tbody>
</table>

You can add more rows to an existing matrix by hitting Ctrl+Return while the cursor is in the matrix. Adding or deleting columns can be done via the menu Edit>Math or the math toolbar.

There are other arrays used in formulas, such as distinctions of cases. It can be created with the menu Insert>Math>Cases Environment or the command \cases. Here is an example:

\[
f(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}
\]

Multi-line formulas are created when you press Ctrl+Return within a formula. In an empty formula you can see that three blue boxes appear, one for each column. When you press Ctrl+Return in a non-empty formula, the part before the relation sign (equal sign “=” etc.) will be inserted automatically to the first column, the relation sign is in the second column, and the rest in the third column. A new row is created by every further hit of Ctrl+Return. Multi-line formulas are always displayed formulas. Here is an example:

\[
a^2 = (b^2 + c^2)(b^2 - c^2) \\
a = \sqrt{b^2 - c^2}
\]

(5.1)

To change the column assignment of the formula parts, place the cursor where you want to start the shift and hit Ctrl-Tab. It shifts everything in the column which is right beside the current cursor position to the next column. Note that the middle column is designed for relation signs, structures within this column will be printed in a smaller size:

\[
\begin{array}{c|c|c} \\
\hline
A & B & A \\
\hline
\end{array}
\]

The multi-line formula type described here is called eqnarray. There are other multi-line types being more suitable for certain situations, for example if you want a better inter-line spacing than in formula (5.1). The other types are described in section 5.8.2.
5.4. Formula Numbering and Referencing

To number a formula, set the cursor in the formula and use the menu Edit ➤ Math ➤ Toggle Numbering or the shortcut Alt+M N. The formula number appears in LyX as “#” within parentheses. The “#” denotes, that the number will be calculated automatically when the output is generated. The placement and format of the formula number in the output depends on the document class. In this document the number is printed together with the chapter number, separated by a dot:

\[ 1 + 1 = 2 \] (5.2)

Using Alt+M N in a numbered formula will switch off the numbering. You can only number displayed formulas.

Multi-line formulas can be numbered line by line: Using the menu Edit ➤ Math ➤ Toggle Numbering of Line or the shortcut Alt+M Shift+N will only toggle the numbering of the line where the cursor is:

\[
\begin{align*}
1 & = 3 - 2 \\
2 & = 4 - 2 \\
4 & \leq 7
\end{align*}\] (5.3)

To number all lines use the shortcut Alt+M N.

Every displayed formula can be referenced by its number using a label. A label is inserted with the menu Insert ➤ Label when the cursor is in the formula. This opens a dialog to enter the label. It is recommended to use the proposed “eq:” as first part of the label, because this helps later to identify the label type when you have many labels in your document. We inserted in the following example the label “eq:tanhExp” in the second line:

\[
\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{2x} - 1}{e^{2x} + 1} \] (5.5)

Every labeled line is automatically numbered. Therefore the label is shown in LyX at the place of the formula number placeholder “#”. You can reference a labeled formula using the menu Insert ➤ Cross Reference. A dialog appears to choose a label you want refer to. The reference appears in LyX as a grey cross reference box and in the output as the formula number:

This is a cross-reference to equation (5.5).

The properties of LyX’s cross-reference box are described in section 6.1. To delete a label, set the cursor in the labeled formula, use the menu Insert ➤ Label and delete the label in the appearing dialog.\(^1\)

\(^1\)This is a unintuitive and will be fixed in the next version of LyX.
5. Mathematical Formulas

Table 5.2.: Typefaces and the corresponding commands.

<table>
<thead>
<tr>
<th>Font</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roman</td>
<td>\texttt{\mathrm}</td>
</tr>
<tr>
<td>Bold</td>
<td>\texttt{\mathbf}</td>
</tr>
<tr>
<td>Italic</td>
<td>\texttt{\mathit}</td>
</tr>
<tr>
<td>Typewriter</td>
<td>\texttt{\mathtt}</td>
</tr>
<tr>
<td>BLACKBOARD</td>
<td>\texttt{\mathbb}</td>
</tr>
<tr>
<td>Fraktur</td>
<td>\texttt{\mathfrak}</td>
</tr>
<tr>
<td>CALLIGRAPHIC</td>
<td>\texttt{\mathcal}</td>
</tr>
<tr>
<td>SansSerif</td>
<td>\texttt{\mathsf}</td>
</tr>
</tbody>
</table>

5.5. User defined math macros

\LaTeX allows you to define macros for formulas which is very useful when you have equations of the same form in a document several times. Math macros are explained in section Math Macros of the Math manual.

5.6. Fine-Tuning

5.6.1. Typefaces

The standard font for text is italic, for numbers the standard is roman. To set a font in a formula, use the Math Panel button, or enter its command, listed in table 5.2, directly.

**Note:** You can only print capital letters in the typefaces Blackboard and Calligraphic.

When you use a typeface, a blue box is inserted in the formula. Every character in this box will be printed in this typeface. Pressing Space within the box, will set the cursor outside, so that you have to use a protected space when you need a space in the box. Here an example where “N” in Blackboard denotes the set of numbers:

\[ f(x) = \sqrt{x}; \ x \in \mathbb{N} \]

The typefaces are nestable, which can cause confusion. You can e.g. put a character in Fraktur in a box for Typewriter: abc\texttt{\textfrak{de}}

So it is better not to use this feature.

The typefaces have no effect on Greek letters: abc\texttt{\textfrak{\textgreek{e}}}

You can only print them emboldened using the command \texttt{\textbf{\boldsymbol}}, which works like the other typeface commands: \texttt{\textbf{\textgreek{\alpha\beta\gamma}}}

\texttt{\textbf{\textgreek{\boldsymbol}}} works for all symbols, letters, and numbers.

A number of other font options are available as well, in the menu Edit>Math>Text Style.
5.6.2. Math Text

Typefaces are useful for entering some characters in some given font, but not for text. For typing longer pieces of text use the math text, which is obtained using the entry Normal text mode of the Math Panel button \textcopyright (shortcut Alt+C Space). Math text appears in LyX in black instead of blue. You can use spaces and accents in math text like in normal text. Here is an example:

$$f(x) = \begin{cases} x & \text{if I say so} \\ -x & \text{unter Umständen} \end{cases}$$

5.6.3. Font Sizes

There are four font styles (relative sizes) used in math-mode, which are automatically chosen in most situations. These are called \texttt{textstyle}, \texttt{displaystyle}, \texttt{scriptstyle}, and \texttt{scriptscriptstyle}. For most characters, \texttt{textstyle} and \texttt{displaystyle} are actually the same size, but fractions, superscripts and subscripts, and certain other structures, are set larger in \texttt{displaystyle}. Except for some operators, which resize themselves to accommodate various situations, all text will be set in the styles that \LaTeX{} thinks are appropriate. These choices can be overridden by using the math panel button \sum. A box for the size will be created in which you can insert the math structure. For example, you can set $\frac{1}{2}$, which is normally in \texttt{textstyle}, larger in \texttt{displaystyle}: $\frac{1}{2}$.

The four styles are used in the following example:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}$$

5.7. Theorem Modules

As of LyX 1.6, support for theorem-like environments has been moved out of the document classes and into layout modules. As a result, theorem-like environments can now easily be used with classes other than the AMS classes. See section 3.1.2.2 for more on layout modules.
5.8. AMS-\LaTeX

\LaTeX supports the packages provided by the American Mathematical Society (AMS) that are in common use.

5.8.1. Enabling AMS-Support

Selecting the checkbox Use AMS math package in the Document > Settings dialog under Math Options will include the AMS-packages in the document, and make the facilities available. AMS is needed for many math-constructs, so when you get \LaTeX-errors in formulas, assure that you have enabled AMS.

5.8.2. AMS-Formula Types

AMS-\LaTeX provides a selection of different formula types. \LaTeX allows you to choose between align, alignat, flalign, gather, and multiline. We refer to the AMS-documentation for an explanation of these formula types.
6. More Tools

6.1. Cross-References

One of LYX’s strengths are cross-references. You can reference every section, float, footnote, formula, and list in the document. To reference a document part, you have to insert a label into it. The label is used as anchor and name for the reference. We want for example to refer to the second item of the following list:

1. First item
2. Second item
3. Third item

First we insert a label into the second item with the menu Insert \ Insert Label or by pressing the toolbar button . A grey label box like this: is inserted and the label window pops up asking for the label text. LYX offers as text the first words of the item with a prefix, in our case the text “enu: Second-item”. The prefix “enu:” stands for “enumerate”. The prefix depends on the document part where the label is inserted, e.g. if you insert a label into a section heading, the prefix will be “sec.”.

To reference the item, we refer to its label using the menu Insert \ Insert Cross-Reference or the toolbar button . A grey cross-reference box like this: is inserted and the cross-reference window appears showing all the labels in the document. We can now sort the labels alphabetically and then choose the entry “enu:Second-item”. At the position of the cross-reference box the item number will appear in the output.

Alternatively to Insert \ Insert Cross-Reference, you can right-click on a label and use in the appearing context menu Copy as Reference. The cross-reference to this label is now in the clipboard and can be copied to the actual cursor position via the menu Edit \ Paste (shortcut Ctrl+V).

Here is our cross-reference: Item 2

It is recommended to use a protected space\footnote{described in section 3.5.1} between the cross-reference name and its number to avoid ugly line breaks between them.

There are six varieties of cross-references:

<reference>: prints the float number, this is the default: 4.3

\footnote{described in section 3.5.1}
6. More Tools

(<reference>): prints the float number within two parentheses, this is the style normally used to reference formulas, especially when the reference name “Equation” is omitted: (5.5)

<page>: prints the page number: Page 104

on page <page>: prints the text "on page" and the page number: on page 104

<reference> on page <page>: prints the float number, the text "on page", and the page number: 4.3 on page 104

Formatted reference: prints a self defined cross-reference format.

Note: This feature is only available when you have the \LaTeX-package preltref installed.

Note that the style <page> won’t print the page number if the label is on the previous, the same, or the next page. You will e.g. see the text “on this page” instead.

The number and current page of the referenced document part in the output, is automatically calculated by \LaTeX. The varieties are adjusted in the field Format of the cross-reference window, that appear when you click on the cross-reference box.

You can only use the style <reference> to reference numbered document parts, while the reference style <page> is always possible.

If you want to reference a section, put the label in the section heading, to reference a float, put the label in the caption. For footnotes you can put the label somewhere in it. Referencing formulas is explained in section 5.4.

The button Go to Label in the cross-reference window sets the the cursor before the referenced label. The button text changes then to Go Back and you can use it to set the cursor back to the cross-reference. Right-clicking on a cross-reference box also sets the cursor before the referred label and you can go back with the toolbar button.

You can change labels at any time by clicking on the label box. References to the changed label will automatically change its link to the new label text, so that you don’t need to take care about this.

If a cross-reference refers to a non-existent label, you’ll see two question marks in the output instead of the reference.

References are described in detail in the Embedded Objects manual.

6.2. Table of Contents and other Listings

6.2.1. Table of Contents

The Table of Contents (TOC) is inserted with the menu Insert\LaTeX List/TOC Table of Contents. Is is displayed in LyX as a gray box. If you click on it, the Outline window appears, showing you the TOC entries as outline to move and rearrange sections in
6.3. URLs and Hyperlinks

6.3.1. URLs

Links to web pages or email addresses can be inserted via the menu Insert\rightarrow URL.

Here is an example URL: LyX’s homepage: \url{http://www.lyx.org}

You cannot change the style of the link text, the URL text will always be in the style Typewriter. To be able to format the URL text, use hyperlinks as explained in the next subsection.

\textbf{Note:} URLs must not end with a backslash, otherwise you get \LaTeX errors.

6.3.2. Hyperlinks

Hyperlinks can be inserted with the menu Insert\rightarrow Hyperlink or with the toolbar button  

The appearing dialog has two fields: Target and Name. The name is the printed text for the hyperlink. The hyperlink type can be a weblink like this: LyX’s homepage, an Email address like this: lyx-docs mailing list, or a link to a file.

You can start applications via a hyperlink when you insert a weblink, but add the prefix \texttt{run:} to the link target.

Hyperlinks will automatically be hyphenated if necessary in the PDF output, and become clickable in the DVI and PDF-output. To set the format of the link text, highlight the hyperlink inset and use the text style dialog. This is for example a hyperlink with bold sans serif text: \textbf{LyX’s homepage}

The link text color can be changed, when the option Color links is set in the PDF Properties dialog (menu Document\rightarrow Settings\rightarrow PDF Properties). The link text is for example set in this document to blue by adding the option \texttt{urlcolor=blue} to the field Additional options in the PDF Properties dialog.
6. More Tools

6.4. Appendices

Appendices are created with the menu Document→Start Appendix Here. This menu sets the document from the current cursor position to the end as the appendix region. The region is marked with a red borderline.

Every chapter (or section) within the appendix region is treated as an appendix, numbered with a capital Latin letter. The appendix subsections are numbered with this letter followed by a dot and the subsection number. All appendix sections can be referenced as if they were normal sections, here two examples:

Appendix E; Appendix A.1.12

6.5. Bibliography

There are two ways of generating the bibliography in a L\LaTeX-document. You can include a bibliography database,\footnote{Known under the name “Bib\LaTeX-database”} which is explained in the next subsection, or you can insert the bibliography manually, using the paragraph environment Bibliography, which was described in section 3.3.8.2. If you want anything other than numerical citations that are used in this document, like author-year citations, then you must use a bibliography database.

6.5.1. The Bibliography Environment

Within the Bibliography environment, every paragraph begins with a gray bibliography box labeled with a number. If you click on it, you will get a dialog in which you can set a Key and a Label. The key is the symbolic name by which you will refer to this bibliography entry. For example, our second entry in the bibliography is a book about \LaTeX and we used “latexcompanion”, a short form of its title, as key.

You can refer to the key of a bibliography entry using the menu Insert→Citation or the toolbar button \(\text{\normalfont }\). A citation reference box is inserted and a citation window will appear in which you can select one or more keys in the available key list. The citation reference box will be labeled with the referenced key. When you click on the box, the citation window appears and you can change the reference.

Citation references appear in the output as the number of the bibliography entry with surrounding brackets. If you set a Label for the entry, the label will appear instead of the number. Here are two examples; the first without a label, the second with the label “Credits”:

Have a look at the \LaTeX Companion Second Edition: [1]

The \LaTeX-Team members are listed in the Credits: [Credits]
6.5.2. Bibliography databases (Bib\TeX)

Bibliography databases are useful if you use the same bibliography in different documents.\textsuperscript{3} It also makes it very easy to have a uniform layout for all bibliography entries. You can collect the bibliography of all relevant books and articles of your working field in a database. This database can be used for different documents, and only the entries cited in a particular document will appear in the bibliography list for that document. This relieves you of the need to keep track of which articles and books you have cited.

The database is a text file with the file extension “.bib”, containing the bibliography in a special format. The format is explained in \cite{bibformat} and in \LaTeX{} books ([MG04, KD03, Lam94]). The file can be created using any text editor, but normally one uses a special program to create and edit the entries in the database. A list of such programs is maintained on the \texttt{L\LaTeX}{} Wiki at \url{http://wiki.lyx.org/BibTeX/Programs}.

To use a database, use the menu Insert $\triangleright$ List/TOC $\triangleright$ \texttt{BibTeX Bibliography}. A grey box will be inserted and a window appears. In this window you can load one or more databases and a style file. The option \textit{Add bibliography to TOC} adds a table of contents entry for the bibliography. In the \textit{Content} drop box you can select what part of the database should be output.

The style file is a text file with the file extension “.bst” that controls how the bibliography entries will appear. Your \LaTeX{} distribution should provide several of these, and many publishers provide their own style files, so that you don’t have to take care of the layout. It is of course possible to write your own style file, but this is something for experts.\textsuperscript{4}

Inserting a citation reference works as described in the previous section.

To generate the bibliography from a database, \texttt{L\LaTeX}{} uses the program \texttt{Bib\TeX}. This program can be controlled with options that you can add in \texttt{L\LaTeX}’s preferences dialog under Outputs $\triangleright$ \LaTeX{} in the \texttt{Bib\TeX} \texttt{command} field. Before adding options, it is strongly recommended to read the manual of \texttt{Bib\TeX} \cite{bibmanual}.

When you select the option \textit{Sectioned bibliography} in the Document $\triangleright$ Settings dialog, it is possible to have multiple and sectioned bibliographies. This and other options are explained in detail in section Customizing Bibliographies with \texttt{Bib\TeX} of the Additional Features manual.

We use two bibliographies in this document to show the difference between the two methods of creating them. As you can see, the bibliography that is created from a database lists only the database entries that are referenced in the document. We used the style file \texttt{alphadin.bst} to get the complicated German reference key scheme in the bibliography.

\textsuperscript{3}They are also useful simply for keeping a database of articles and notes concerning them. Most of the database programs mentioned below allow you to store annotations and reviews along with bibliographical information.

\textsuperscript{4}For information how this is done, have a look at \url{http://www.ctan.org/get/biblio/bibtex/contrib/doc/btxhak.pdf}. 

123
6.5.3. Bibliography layout

In the citation reference dialog you can set a special citation format. For this feature you need to enable the option Natbib in the Document>Settings dialog under Bibliography. Setting a citation style for a reference will overwrite the default. For the global citation format use the Bib\TeX style files as explained in the previous section.

You can also set text, that should appear before or after a citation reference, in the citation reference window. Here an example where we set the text “Chapter 3” to appear after the reference:

Have a look at [1, Chapter 3].

6.6. Index

An index entry is created if you use the menu Insert>Index Entry or the toolbar button . A gray box labeled “Idx” is inserted containing the text that appears in the index. The word where the cursor is in or the currently highlighted text is proposed by \LaTeX as the index entry.

We give a short overview of the index commands in the next subsections. For a detailed description of \LaTeX’s index mechanism, have a look at one of the \LaTeX books [1, 2, 3].

You can change index entries by clicking on the index box.

The index list is inserted in the document with the menu Insert>List / TOC>Index List. A light blue box labeled “Index” will show the place where the index is printed in the output. The index list box is not clickable like other \LaTeX-boxes.

6.6.1. Grouping Index Entries

Index entries are often grouped to offer the reader a fast search in the index. We want to group for example the index entries for itemized and enumerated lists under the entry “Lists”. First we create the entry “Lists” in section 3.3.6. In the text field for the itemized list index entry in section 3.3.6.2, we insert the command

Lists ! Itemize

and the command

Lists ! Enumerate

for the enumerated list in section 3.3.6.3.

The exclamation mark “!” marks the grouping levels. You can have three levels; every index level is indented a bit more. An index entry for the higher levels is not required. If we don’t have an index entry for “Lists”, it will be printed anyway, but without a page number.
6.6.2. Page Ranges

Normally an index entry will appear with the page number of the indexed section. But sometimes you want to index more pages under the same entry. E.g. if we want to index the paragraph environments, we create an index entry in section 3.3 with the command

```
Paragraph environments|
```

and another entry at the end of section 3.3.9 with the command

```
Paragraph environments|
```

The commands “|” and “|)” respectively start and end the index range. You can also add the same index entry at different places in the document. They appear in the output under one entry with a comma separated list of the pages of the indexed document parts. An example is the index entry “Document ! Settings”.

6.6.3. Cross referencing

It is also possible to refer to another index entry. We referred for example in the index entry “GIF” (in section 4.4.1) to the index entry “Image formats” in the same section using the entry

```
GIF|see{Image formats}
```

where the braces have to be inserted as \TeX Code. The text within the braces is the referenced entry. The reference will appear in the output without a page number.

6.6.4. Index Entry Order

You can use accented characters in the index entry, but the entries might then not follow the rules for the index order. The index entries are sorted alphabetically, but \TeX doesn’t know how to sort accents in different languages. We have created as an example the three dummy index entries “maison”, “maïs”, and “maître”. They will be sorted in the order maïs, maître, maison, but we want the order maïs, maison, maître. To achieve this, we use the command

```
previous entry@current entry
```

In our case we want to have “maison” after “maïs” and write therefore for the index entry of maison:

```
maïs@maison
```

The previous entry needn’t to be a real existing entry, you can also use another word to tell \TeX the entry order, see the next subsection for an example.

In some cases the index entry order is not correct when you are using the program `makeindex` to generate the index (see sec. 6.6.6). `makeindex` would for example print the index entry for the \TeX-package aeguill in sec. 3.7.2 after the index entries of

---

5The index generating is done in the background by an extra program, see section 6.6.6.
the other \LaTeX-packages although all these index commands start with "\LaTeX-packages!". The reason is that the index entry for aeguill is in a footnote. To fix this makeindex bug, add these commands to the preamble of your document:
\begin{verbatim}
\let\OrgIndex\index
\renewcommand*{\index}{\OrgIndex{#1}}
\end{verbatim}

6.6.5. Index Entry Layout

You can change the appearance of index entries via the text style dialog. You can also format the page number using the character "|" followed by a \LaTeX-command without a backslash. We can write for example

\textit{italic page number:|textit}

to get the page number in italic. Normally all \LaTeX-commands begin with a backslash, but in this special case "|command" means \texttt{\textbackslash command\{page number\}}. Have a look at section 6.10.2 to learn more about the \LaTeX-syntax.

\textbf{Note:} Formatting single index entries only works when you use the program makeindex to generate the index, see sec. 6.6.6. If you use xindy, however, this won’t work for anything other than bold or italic text. This is because xindy requires to define semantic elements before they can be used, see [1, p. 678 ff.] for details.

In general, we encourage you to not format page numbers directly as shown above. Instead of this, you should define a macro in the preamble and use that. Consider why you want some page numbers to be bold. Maybe you want all page references italic that refer to a definition of the indexed term, so that users can easily find definitions. If so, put the following in the preamble
\begin{verbatim}
\newcommand{\IndexDef}{1}{\textit{#1}}
\end{verbatim}
and write
\begin{verbatim}
my entry|IndexDef
\end{verbatim}
in the index entry. The advantage is that, if you change your mind later or if your publisher insists that definitions must not be italic but bold, you just need to change the macro in the preamble, not every single index entry.

You can also change the layout for the whole index. E.g. we marked the index list box of this document as bold to get a bold font for all index entries. For more advanced tasks you have to set up a so-called Index Style File, see the makeindex or xindy documentation for details, [9, 10].

6.6.6. Index Program

When the index entry program xindy, which is only available for Linux, is installed, LyX uses it for index generation; otherwise the program makeindex, that is part of every \LaTeX\ distribution, is used. Both programs can be controlled by options that can be set in LyX’s preferences dialog, see section C.6.4. The available options are listed and explained in [9, 10]. You can also specify there another program to generate the index.
makeindex is very old, no longer under development and has many pitfalls, notably that it was developed with only the English language in mind. So it fails to sort anything other than a monolingual English text correctly. We have shown above how to fix this sorting. However, if you are writing in another language and using Linux, consider to use xindy.

## 6.7. Nomenclature / Glossary

Sometimes you need to compile a list of symbols that are mentioned in your document with a brief explanation of them – a so called nomenclature or glossary.

To be able to create nomenclatures, you need the \LaTeX\ package nomencl installed. You find it in the \TeX\ Catalogue, \cite{5} or in the package manager of your \LaTeX\-system.

A nomenclature entry is created if you place the cursor after a symbol entry and then use the menu Insert $\rightarrow$ Nomenclature Entry or the toolbar button $\mathbb{E}$. A gray box labeled “Nom” is inserted and a window pops up asking for the nomenclature entry.

A nomenclature entry consists of two main entries. The first is the symbol that you want to refer to. The second is the description of the symbol.

Note: You have to enter valid \LaTeX\-code for all fields of the nomenclature dialog.

### 6.7.1. Nomenclature Definition and Layout

When you have symbols in formulas, you have to define them in the Symbol field as \LaTeX\-formulas. For example to get “$\sigma$”, insert this:

\[
\sigma
\]

The “\$” character starts/ends the formula. The \LaTeX\-command for the Greek letter is the name of the letter beginning with a backslash “\”. For capital Greek letters, start the command also with a capital letter, like \texttt{\Sigma}.

(A short introduction to the \LaTeX\-syntax is given in section 6.10.2.)

You cannot use the Text Style dialog to format the description text but have to use \LaTeX\-commands. For example the description of the nomenclature entry for the “$\sigma$” in this document is:

\texttt{dummy entry for the character \textsf{sigma}}

The command \texttt{\textsf} sets the fonts to sans serif. To get bold font use the command \texttt{\textbf}, for typewriter use \texttt{\texttt}, for emphasized use \texttt{\emph}.

### 6.7.2. Sort Order of Nomenclature Entries

The nomenclature entries are sorted alphabetically by the \LaTeX\-code of the symbol definition. This leads to undesired results when you for example have symbols in formulas. Suppose you have nomenclature entries for the symbols $a$ and $\sigma$. They will be sorted by “a” and “$\texttt{\sigma}$” – the $\sigma$ will be sorted before the $a$ since the character “$\$” is considered in sorting.
To control the sort order, you can edit the **Sort as** field of the nomenclature dialog. Then the nomenclature entry will be sorted by this entry and not the symbol definition. For the example given, you can insert `sigma` in this field for the `σ`, then `a` will be located before `σ`.

For subgrouping and tips for using sort entries see the **nomencl** documentation, [14].

### 6.7.3. Nomenclature Options

The **nomencl** package offers some options to adjust the appearance of the nomenclature. Here are some of its options, for more have a look at its documentation:

**refeq** Appends the phrase “,** see equation (eq)” to every nomenclature entry, where *eq* is the number of the last equation in front of the nomenclature entry

**refpage** Appends the phrase “,** page (page)” to every nomenclature entry, where *page* is the number of the page on which the nomenclature entry appeared

**intoc** Inserts the nomenclature in the Table of Contents

There are furthermore the options **croatian**, **danish**, **english**, **french**, **german**, **italian**, **polish**, **portuguese**, **russian**, **spanish**, and **ukrainian** to print the reference texts and the nomenclature title in the corresponding language.

To use one or more of the options, add them to the comma-separated document class options list in the **Document** > **Settings** dialog. In this document the option **intoc** is used.

You can also use the first two options above only for certain nomenclature entries when you add one of the following commands as last entry to the **Description** field in the nomenclature dialog:

\nomrefeq  Like the **refeq** option
\nomrefpage  Like the **refpage** option
\nomrefeqpage  Short notation of **\nomrefeq\nomrefpage**
\nomnorefeq, \nomnorefpage, \nomnorefeqpage  Turns off the corresponding options

### 6.7.4. Printing the Nomenclature

To print the nomenclature, use the menu **Insert** > **Lists** > **TOC** > **Nomenclature**. A light blue box labeled “**Nomenclature**” will show the place where the nomenclature is printed in the output. Like the index list box, the nomenclature list box is not clickable.

In the printed output the title of the nomenclature appears as “Nomenclature”. If you are not happy with the name, you can change it by redefining the command
\nomname in the preamble. For example, in order to change the name to *List of Symbols*, add the following line to the preamble:

\renewcommand{\nomname}{List of Symbols}

If you are unhappy with the amount of space for symbols, you can alter it by adding the following line to the preamble:

\renewcommand{\nomlabelwidth}{width}

where the width is a value with one of the units listed in Appendix D. The default value is 1 cm.

### 6.7.5. Nomenclature Program

*\LaTeX* uses the program *makeindex*, that is part of every *\LaTeX* distribution, to generate the nomenclature. *\LaTeX*’s preferences dialog allows you to specify another program or to control *makeindex* by adding options, see section C.6.4. The available options are listed and explained in [14, 9].

### 6.8. Branches

Sometimes it is useful to hide some document parts in the output. For example a teacher who is setting an exam obviously doesn’t want the pupils to see the answers, but having questions and answers in the same document will make the life of the markers of that exam much easier.

For these cases *\LaTeX* allows you to put text into branches. The text will then only appear in the output when its branch is activated. To create a branch, go in the Document $\rightarrow$ Settings dialog to Branches. The name of the branch, its activation state and the background color of the branches inside *\LaTeX* can be specified in this dialog.

Text that should be in a branch is set into branch inset boxes. These boxes are inserted via the menu Insert $\rightarrow$ Branch where you can choose a branch. You can later change the branch of the boxes by right-clicking on them.

Here is an example, where only the question text appears, the answer branch is deactivated and does therefore not appear in the output:

Question: Who was the first physics Nobel prize winner?

To use conditional output inside places where you cannot insert branch insets, like inside equations, you can code special *\LaTeX* definitions for each branch. For example you can define for the question branch

\newcommand{\question}[1]{\textcolor{#1}{#1}}
\newcommand{\answer}[1]{}

and for the answer branch

\newcommand{\question}[1]{}
\newcommand{\answer}[1]{\textcolor{#1}{#1}}

---

\footnote{For an introduction to the *\LaTeX*-syntax, see section 6.10.2.}
6. More Tools

Now it is possible to use the commands \question{...} and \answer{...} to obtain conditional output. Here is an example formula where only the \question part appears:

\[ x^2 - 2x - 2 \Rightarrow x_1 = 1 + \sqrt{3}. \]

Inside math, the same effect can be achieved using math macros, see the Math manual.

6.9. PDF Properties

The Document Settings dialog allows you in the PDF Properties to set up special options for the PDF output of your document. All options there are provided by the \LaTeX\-package hyperref.

Using hyperref will link all cross-references in the DVI- and PDF-output. This means that the reader of your document will be able to click on a table of contents entry or on a reference and he is shown the referenced document part. You can specify in the dialog tab Hyperlinks how the links will look like and if links for bibliographical backreferences are created. The backreferences will appear in the bibliography behind the different entries, showing the number of the section, slide, or page where the entry is referenced.

In the dialog tab Bookmarks you can set if PDF-bookmarks should be created for every section of your document to make it easier for readers to navigate through the document. You can decide if the bookmarks should be numbered like your document sections or not. With the open bookmarks level you can specify what sectioning level should be displayed in the bookmarks when opening the PDF. For example level 2 will display all sections and subsections, while level 1 will only display the sections.

The header information in the dialog tab General are saved together with the PDF as file properties. Many programs are able to extract this information to e.g. automatically recognize who the author is and what the PDF is about. This is very useful to sort, classify, or use PDFs for bibliography issues. When the option Automatic fill header is set, LyX tries to extract the header information from your document title and author settings.

The option Load in fullscreen mode will open the PDF in fullscreen mode, which is useful for presentations.

PDF properties are also used in this document. When you look in its document settings, you can see that some additional hyperref options are used. For an explanation of them we refer you to the hyperref manual [13].
6.10. \TeX\ Code and the \LaTeX\ Syntax

6.10.1. \TeX\ Code Boxes

As \LaTeX\ is used in the background, it supports many \LaTeX\ commands and constructs, but not all. \LaTeX\ contains of hundreds of packages which provide different commands. All the time packages are being updated and new ones added. This has the advantage that you can typeset nearly everything as there is for every problem a \LaTeX\-package. But \LaTeX\ can of course not be up to date and support all packages and their commands.

But don’t worry, you can use any \LaTeX\-command directly in \LaTeX\ inside the \TeX\ Code box. A \TeX\ Code box is created by the menu \textit{Insert} \textarrow{\TeX\ Code} or by the toolbar button \TeX\. The box can be opened by left-clicking and closed by right-clicking on it.

You can insert complete or incomplete commands as \TeX\ Code. Incomplete means the command argument can be Standard \LaTeX\ text. For example, if you want to draw a frame around a word are therefore using the \LaTeX\-command \texttt{\fbox}, you can write the command part \texttt{\fbox{}} in a \TeX\ Code box before the word and the closing brace \texttt{}} in a second \TeX\ Code box behind the word. The word between the two \TeX\ Code boxes is then the argument as it is in the following example:

\begin{verbatim}
This is a line with a \fbox{framed} word.
\end{verbatim}

gives

This is a line with a \fbox{framed} word.

\textbf{Note:} At the end of \LaTeX\-commands without parameters, you have to insert a space to let \LaTeX\ know that the command is finished.

6.10.2. Short Introduction to the \LaTeX\ Syntax

When you write larger documents or books, you will need to know something about the \LaTeX\-commands that \LaTeX\ uses in the background. Because \LaTeX\ is based on commands, you can “program” your text. This has the advantage that the layout of the document can be changed at any time if you know the right commands. E.g. imagine you have to write a manual for a product and the deadline is the end of the day. Your boss just has complimented you for your good work but wants to have all caption labels bold. But you have over hundred figure and table captions with non-bold labels in your manual. Of course it’s impossible to change all caption labels manually in one day.

Now \LaTeX\ comes into play. As mentioned above, for every problem there exists a \LaTeX\-package. First you have to find out which and therefore look in the \LaTeX\ package database, [FW].

As result you know that the package \texttt{caption} is what you need. To use a package, you have to load it in the document preamble (menu \textit{Document} \textarrow{Settings}) with the command
\usepackage[options]{package name}

All \LaTeX{} commands begin with a backslash, the command argument is set within two braces, and the options are set within two brackets. Note that not all commands have an argument and options.

In your case the package name is \texttt{caption}. After a look in the documentation of the package, you know that the option \texttt{labelfont=bf} will change the font of all caption labels to bold. So you add the command
\begin{verbatim}
\usepackage[labelfont=bf]{caption}
\end{verbatim}
to the preamble and the problem is solved.\footnote{For more commands provided by the \texttt{caption} package, have a look at its documentation, \cite{caption}.}

Note that some document classes have built-in solutions for well known problems like your case. For example if you use a \texttt{koma-script} class, you don’t need the package \texttt{caption}, you can instead write
\begin{verbatim}
\setkomafont{captionlabel}{\bfseries}
\end{verbatim}
in the preamble and the problem is solved. So if you plan to write a large document, you should have a look at the documentation of the document class you want to use. (\texttt{\setkomafont} is an example of a command with more than one argument.)

Commands in the preamble affect the whole document, while commands in the text affect only the text after the command or only the text used as command argument. To insert a \LaTeX-command in text, use the \TeX{} Code box as described in the previous section.

If you want to learn more about \LaTeX{} and its syntax, have a look at the \LaTeX{}-books \cite{ref1,ref2}.

\section{Previewing Snippets of your Document}

\LY{} allows you to generate previews of sections of your document on the fly so you can see how they’ll look in the final document without having to break your train of thought with View\:>\>DVI.

If you would for example like to see in \LY{} your math formulas typeset by \LaTeX{}, install the \LaTeX{}-package \texttt{preview-latex} as explained below, and turn on Instant preview in the Tools\:>\>Preferences dialog under Look and feel\:>\>Graphics.

Previews are generated when you load a document into \LY{} and when you finish editing an inset. Previews of an already loaded document are not generated just by selecting the Instant preview check box, you have to reopen the documents to activate the previews.

\LY{} will generate previews of math insets. It will also generate previews of included insets if you select the Show preview check box in the insert dialog. This is useful if you wish to generate a preview of a \LaTeX{} figure, for example.

To get previews working, you need the \LaTeX{} package \texttt{preview-latex} (on some systems named simply \texttt{preview}) installed. If it is not already installed, you will find it in the \TeX{} Catalogue, \cite{ref5} or in the package manager of your \LaTeX{}-system. You
obtain prettier results if you install the program `pnmcrop` from the `netpbm` package; for LyX on Windows this program is automatically installed together with LyX.

You can furthermore preview the \LaTeX{} source of the whole document or parts of it. Use the menu `View \rightarrow View Source` and a window will be shown where you can see the \LaTeX{}-source code. The window shows the source of the whole paragraph where the cursor is currently in. You can also select document parts in LyX’s main window, then only this selection (when it is more than one paragraph) is shown as source code. To view the whole document as source, enable the corresponding option in the source view window.

## 6.12. Spell Checking

LyX itself has no built-in spell checker. Rather it uses one of the external programs `aspell`, `ispell`, `hspell`, or `pspell` as backend. This section assumes you have already installed and set up one of these programs. `aspell` can be seen as the successor to `ispell`, so that it is recommended to use `aspell`. `hspell` is a Hebrew spell-checker. The used spell checker and its settings are specified in LyX’s preferences under `Language Settings`.

For LyX on Windows, the selection box for the spell checking program is greyed out in the preferences dialog because only `aspell` can be used.

The menu `Tools \rightarrow Spellchecker` or the toolbar button `_spell_` starts the spell checking beginning from the current cursor position. A dialog window will appear showing any incorrect (or unknown) word found, allowing you to edit and replace it in a second line. Whenever an unknown word is found, the word is highlighted and the text scrolled so that it is visible. In the spell checker dialog, there is also a box showing suggestions for a correction, if any could be found. Clicking on one of the corrections will copy to the `Replace` field, double-click invokes directly the replacement. Unknown but correctly typed words can be added to the personal dictionary.

By default, the dictionary file used is determined by the document language that is set in the `Document \rightarrow Settings` dialog. If you do not have a dictionary for the document language, spell checking will bring an error message. In this case, you can specify another dictionary file in the dialog by specifying a different `Alternative language` in preferences dialog.

After a spell check you will be informed about the number of checked words.

### Limitations

It is not possible to change the spelling of a particular word globally, rather than having to change the spelling separately for each occurrence of the word. But you can use the `Find & Replace` dialog for that.

LyX cannot correctly spell check documents containing multiple languages. This does work with `pspell`, assuming you have marked the different languages appropriately.
6. More Tools

Further Settings

The Spellchecker section in the preferences dialog has some additional options:

**Escape characters** Allows you to add non-standard characters that the spell checker should consider, e.g. German umlauts although you are spell checking an English document. This should not normally be needed.

**Personal dictionary** to use a different file as your personal dictionary instead of the spell checker’s default choice

**Accept compound words** Prevent the spell checker from complaining about compounded words like “passthrough”.

**Use input encoding** Uses the document encoding that is set in the Document>Settings dialog under Language also for the spellchecker.

There have been reports that this does not work with all dictionaries, so this is disabled by default.

6.13. Thesaurus

Thesaurus currently only works when you use the document language English.

To start the thesaurus, highlight one word or set the cursor behind it, and use the menu Tools>Thesaurus or the toolbar button A dialog pops up showing you possibly related words that you can use as replacement.

The related words shown may not really be related to the word you are currently checking; scrolling in the proposal list might help in some cases to find related words.

The thesaurus only works for single words, and also only when it is in the singular form. For example starting the thesaurus with the word “reports” leads to no results, while results are shown for the word “report”. To avoid this, you can highlight only the part “report” of the word “reports” to get results.


When you work on a document collaboratively it is extremely useful to be able to see changes that others have made highlighted in the document. You can then decide if you want to accept a change or not. This can be achieved by turning on change tracking in the menu Document>Change Tracking>Track Changes.

Changes made in the document will then be highlighted by strokes and colors: **underlined text** is an addition, **canceled text** is a deletion. The color depends on the author that made the change. You can change the color in LyX’s preferences dialog.

---

8The encodings are explained in section B.
under Look and feel, Colors. The author and the date of the change are shown in \LaTeX’s status bar when the cursor is in changed text. The same information is shown when you use the toolbar button

When change tracking is activated, you will see the review toolbar in \LaTeX:

The review toolbar as shown above contains from left to right the following buttons:

- Document ➤ Change Tracking ➤ Track Changes
- Document ➤ Change Tracking ➤ Show Changes in Output
- ➤ Jumps to the next change
- Document ➤ Change Tracking ➤ Accept Change
- Document ➤ Change Tracking ➤ Reject Change
- Document ➤ Change Tracking ➤ Merge Changes
- Document ➤ Change Tracking ➤ Accept All Changes
- Document ➤ Change Tracking ➤ Reject All Changes
- ➤ Insert ➤ Note ➤ \LaTeX Note
- ➤ Navigate ➤ Next Note

The review toolbar helps you to accept, reject, or merge changes – highlight the change and press one of the desired toolbar buttons. When you merge changes, a window pops up showing you information about the next change behind the current cursor position. So you don’t need to highlight a certain change. Within the merge window you can decide to accept or reject changes and step to the next change. This way you can jump through all the changes in the document.

The toolbar has two buttons to handle notes because notes are often important to describe a change.

To show made changes in the output you need the \LaTeX package dvipost installed. You will find it in the \TeX Catalogue, [5] or in the package manager of your \LaTeX-system.
6.15. International Support

This section describes how to use LyX with any language you want. For some non-western languages there are special Wiki-pages that explain how to set up LyX to use them: [18, 19, 20, 21]

Besides languages, LyX also supports phonetic symbols, see section A.4.2.

6.15.1. Language Options

The Document\textgreater Settings dialog lets you set the language and character encoding for your language.

Choose your language in the Language section of this dialog. The default is English.

Under Encoding you can choose the character encoding map you want to use for \LaTeX{} export. The option Language Default is the preferred choice and works well in most cases. For details about the different encoding options see section B.

6.15.2. Keyboard mapping configuration

If you have for example a U.S.-style keyboard and want to write in a language other than English, you can use an alternate keymap. For example, if you have a U.S.-style keyboard but want to write in Italian, you can configure LyX to use an Italian keymap. The preferences dialog allows you to choose up to two keyboard mappings, see section C.2.3. You can choose primary and secondary keyboard languages and then select which one you want to use.

Finally, you may just want to change a few key mappings or create an entirely different keymap (for Vulcan, for instance). You may, for example, normally write in Italian on a U.S. keyboard but want to include an occasional quotation in German. In such a case, you can write your own keyboard mapping or modify an existing one to support the characters you want. This and many other customizations are explained in the Customization manual.

6.15.3. Character Tables

Table 6.2 shows the Latin1 character set. You should be able to enter the characters in the first eight columns directly from the keyboard.

There are a few things you need to know about this table. Here are some of the details you’ll need to bear in mind when using characters from the Latin1 character set:

- Even if you have selected latin1 in the Document\textgreater Settings dialog, users who have only the T1-fonts for \LaTeX{} [or who have the T1-fonts but aren’t using them] will still miss a few characters: D0, F0, DE, FE, AB, and BB – the uppercase and lowercase eth and thorn, and the french quotes won’t show up.
### Table 6.2: The latin1 character set

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>0A</th>
<th>0B</th>
<th>0C</th>
<th>0D</th>
<th>0E</th>
<th>0F</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>@</td>
<td>P</td>
<td>'</td>
<td>p</td>
<td>˚</td>
<td>Å</td>
<td>D</td>
<td>à</td>
<td>ô</td>
<td>˚</td>
<td>Ö</td>
<td>d</td>
<td>à</td>
<td>Ô</td>
<td>ô</td>
</tr>
<tr>
<td>01</td>
<td>!</td>
<td>1</td>
<td>A</td>
<td>Q</td>
<td>a</td>
<td>q</td>
<td>i</td>
<td>±</td>
<td>Å</td>
<td>Ñ</td>
<td>á</td>
<td>ñ</td>
<td>˚</td>
<td>Ô</td>
<td>ô</td>
<td>˚</td>
</tr>
<tr>
<td>02</td>
<td>&quot;</td>
<td>2</td>
<td>B</td>
<td>R</td>
<td>b</td>
<td>r</td>
<td>c</td>
<td>˚</td>
<td>Å</td>
<td>O</td>
<td>à</td>
<td>à</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>03</td>
<td>#</td>
<td>3</td>
<td>C</td>
<td>S</td>
<td>c</td>
<td>s</td>
<td>£</td>
<td>˚</td>
<td>Å</td>
<td>Ö</td>
<td>á</td>
<td>ò</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>04</td>
<td>$</td>
<td>4</td>
<td>D</td>
<td>T</td>
<td>d</td>
<td>t</td>
<td>z</td>
<td>˚</td>
<td>Å</td>
<td>O</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>ò</td>
<td>˚</td>
</tr>
<tr>
<td>05</td>
<td>%</td>
<td>5</td>
<td>E</td>
<td>U</td>
<td>e</td>
<td>u</td>
<td>¥</td>
<td>µ</td>
<td>Å</td>
<td>Ö</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>06</td>
<td>&amp;</td>
<td>6</td>
<td>F</td>
<td>V</td>
<td>f</td>
<td>v</td>
<td>˚</td>
<td>µ</td>
<td>Å</td>
<td>Ö</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>07</td>
<td>'</td>
<td>7</td>
<td>G</td>
<td>W</td>
<td>g</td>
<td>w</td>
<td>§</td>
<td>ç</td>
<td>Ç</td>
<td>Ç</td>
<td>ç</td>
<td>ç</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>08</td>
<td>(</td>
<td>8</td>
<td>H</td>
<td>X</td>
<td>h</td>
<td>x</td>
<td>˚</td>
<td>µ</td>
<td>Å</td>
<td>Ö</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>09</td>
<td>)</td>
<td>9</td>
<td>I</td>
<td>Y</td>
<td>i</td>
<td>y</td>
<td>˚</td>
<td>µ</td>
<td>Å</td>
<td>Ö</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>0A</td>
<td>*</td>
<td>:</td>
<td>J</td>
<td>Z</td>
<td>j</td>
<td>z</td>
<td>a</td>
<td>o</td>
<td>Å</td>
<td>U</td>
<td>é</td>
<td>û</td>
<td>©</td>
<td>1</td>
<td>E</td>
<td>Ù</td>
</tr>
<tr>
<td>0B</td>
<td>+</td>
<td>;</td>
<td>K</td>
<td>l</td>
<td>k</td>
<td>l</td>
<td>˚</td>
<td>µ</td>
<td>Å</td>
<td>Ö</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>0C</td>
<td>,</td>
<td>&lt;</td>
<td>L</td>
<td>\</td>
<td>l</td>
<td>\</td>
<td>˚</td>
<td>µ</td>
<td>Å</td>
<td>Ö</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>0D</td>
<td>-</td>
<td>=</td>
<td>M</td>
<td>]</td>
<td>m</td>
<td>}</td>
<td>˚</td>
<td>µ</td>
<td>Å</td>
<td>Ö</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>0E</td>
<td>.</td>
<td>&gt;</td>
<td>N</td>
<td>^</td>
<td>n</td>
<td>~</td>
<td>˚</td>
<td>µ</td>
<td>Å</td>
<td>Ö</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>0F</td>
<td>/</td>
<td>?</td>
<td>O</td>
<td>_</td>
<td>o</td>
<td>_</td>
<td>˚</td>
<td>µ</td>
<td>Å</td>
<td>Ö</td>
<td>à</td>
<td>å</td>
<td>˚</td>
<td>O</td>
<td>à</td>
<td>à</td>
</tr>
</tbody>
</table>

The following is a full list of all of the accented characters LyX can display directly. It includes not only the accented characters from the previous table, but also the characters from ISO8859–2 through 4.

- **From ISO8859–1:**
  - Á É Í Ó Ù ä é í ó ù
  - ~Á É Í Ó Ù á é í ó ù
  - ‘Á É Í Ó Ù à é i ó ù
  - ’Á É Í Ó Ù Yá é í ó ù
  - ~Á N Ó â ñ ô
  - ÇÇ
  - –

- **From ISO8859–2 through 4:**
  - ÊÊÍÍÔÔÚúüý
diaeresis
  - circumflex
  - grave
  - acute
  - tilde
  - cedilla
  - macron
  - HhjĆGŠš
  - ŠŽšžŘRLČNřčn
  - Ířůů
  - ŠšťšřšlšgšNKuš
  - ĖeĂOŬarôů
  - circumflex
critical
  - acute
tilde
cedilla
  - macron

137
6. More Tools

ÖÖöö ŌÜōū Hungarian umlaut

All the characters above are actively supported by TeX fonts. In addition TeX allows diacritical marks on almost all characters. Also make sure you’re using the T1 font-encoding.
A. The User Interface

This appendix lists all the available menus and describes their functionality. It is designed as a quick reference if you are searching for a special topic inside the user’s guide.

A.1. The File Menu

Under the File menu are the basic operations in addition to some more advanced operations. At the end of the menu the four last opened documents are listed.

A.1.1. New

Creates a new document.

A.1.2. New from Template

This menu entry prompts you for a template to use. Selecting a template will automatically set certain layout features for the document, features you would otherwise need to change manually.

A.1.3. Open

Opens a document.

A.1.4. Open Recent

The submenu shows a list of the recently opened files. Click there on a file to open it.

A.1.5. Close

Closes the current document.

A.1.6. Save

Saves the actual document.
A. The User Interface

A.1.7. Save As
Saves the actual document under a new name to create a copy.

A.1.8. Save All
Saves all opened documents.

A.1.9. Revert to saved
Reloads the actual document from disk.

A.1.10. Version Control
This is used when more people are working on the same document. It is described in the section Version Control in LyX of the Additional Features manual.

A.1.11. Import
You can import there files from older LyX-versions, HTML-files, \LaTeX-files, NoWeb-files, plain text files, and comma separated, table like, text files (CSV) as a new LyX-document. The files will be imported as new LyX-document.

When using the menu entry Plain Text, line breaks in the text will start a new paragraph; when using the menu entry Plain Text, Join Lines, consecutive lines of text will be imported to one big paragraph. A new paragraph will begin when there is a blank line in the file.

A.1.12. Export
You can export your document to various file formats. The resulting files are placed in the directory of your LyX-file. The menu entries are not the same on all installations. They depend on the programs found by LyX while its configuration.

Here is a list of all available entries; they are explained in detail in section 3.8.2:

**CJK LyX** format of the special LyX 1.4.x versions for Chinese, Japanese, and Korean (CJK); (Since LyX 1.5.0 CJK support is fully integrated to LyX.)

**DVI** DVI-format

**HTML** HTML-format (the HTML-converter is a third-party product and doesn’t work in all cases)

**HTML (MS Word)** HTML-format specialized so that the result can be imported to MS Word, as consequence of this formulas will be embedded as bitmap fonts and not in the format MathML.
**LaTeX (pdflatex)** text file with the \LaTeX source, additionally all images used in the document will be converted to a format that is readable by the pdflatex program (GIF, JPG, PDF, PNG)

**LaTeX (plain)** text file with the \LaTeX source code, additionally all images used in the document will be converted to the EPS-format, only this format is readable by the latex program

**LyX 1.y.x** LyX-document in a format readable by the LyX versions 1.y.x (“y” is replaced by the version number)

**OpenDocument** OpenDocument-formatted text file, to be opened with OpenOffice, KOffice, Abiword, etc. (the OpenDocument-converter is a third-party product and doesn’t work in all cases)

**PDF (dvipdfm)** PDF-format using the program dvipdfm

**PDF (pdflatex)** PDF-format using the program pdflatex

**PDF (ps2pdf)** PDF-format using the program ps2pdf

**Plain text** text format

**Plain text (ps2ascii)** text format, the document will first be converted to Postscript format and then exported as text using the program ps2ascii

**Postscript** PostScript format using the program dvips

**Custom** custom format

The program dvipdfm produces internally a DVI-file which is then converted to a PDF-file. It is a bit out of date and therefore the output could look different from what you want. pdflatex produces PDF-files directly and supports the latest PDF-file formats.

If one of the menu entries DVI, PDF (pdflatex) or Postscript is missing, you need to update your \LaTeX installation. After updating you have to reconfigure LyX, see section 1.4.

The menu Custom allows you to export your file by using special command line options for the export program.

**A.1.13. Print**

With this menu entry you can print your document to a file in PostScript format or send it to a printer. The printer will also use the document in PostScript format. The conversion to PostScript is done in the background by LyX using the program dvips. For more information have a look at section 3.8.4.
A. The User Interface

A.1.14. Fax

This menu entry will only appear when you have a fax program installed (on Windows you additionally need to register its program path to \texttt{LYX}'s PATH prefix, see section C.3). With this menu entry you can send your document to a fax program like \texttt{hylapex} or \texttt{kdeprintfax}. The default format of the sent file is PostScript. The format can be changed in \texttt{LYX}'s preferences as described in section C.7.1.

A.1.15. New and Close Window

Opens or closes a new instance of \texttt{LYX}.

A.1.16. Exit

Prompts you to save all unsaved documents and then exits.

A.2. The Edit Menu

A.2.1. Undo and Redo

Described in section 2.3.

A.2.2. Cut, Copy, Paste, Paste Recent, Paste Special

Described in section 2.2.

A.2.3. Select All

Selects the whole document.

A.2.4. Find & Replace

Described in section 2.2.

A.2.5. Move Paragraph Up/Down

This shifts the paragraph where the cursor is currently in one paragraph up or down.

A.2.6. Text Style

Described in section 3.7.4.
A.2.7. Paragraph Settings

Enables you to set the paragraph alignment, line spacing and label width. These settings only affect the paragraph in which the cursor is currently.

You can also prevent the first line of the paragraph being indented if you have chosen to separate paragraphs with indents in the Document Settings dialog under Text Layout.

A.2.8. Table Settings and Math

These two menus are only fully active when the cursor is inside a table or a formula. Here you can change the properties of tables and formulas. The properties of tables are described in section 4.5, the properties of formulas in chapter 5.

A.2.9. Increase / Decrease List Depth

These menu entries are only active when the cursor is in an environment that can be nested. They increase/decrease the environment nesting level as explained in section 3.4 and 3.3.4.3.

A.3. The View Menu

The View menu contains a list of available formats in which you can view the actual document with an external program. The menu entries for the viewing formats are not the same on all installations — it depends on the \LaTeX programs that are found while LYX was configured. All possible formats are formats listed in section A.1.12. You should at least see the menu entries DVI and PDF (pdflatex). If one of the two is missing, you need to update your \LaTeX installation. After updating you have to reconfigure LYX, see section 1.4.

Invoking a menu will start a viewer program. The viewer can be set or changed in the preferences, see section C.7.2. The default viewers are set by LYX while it is first configured.

At the bottom of the View menu the opened documents are listed.

A.3.1. Open/Close all Insets

Opens/closes all insets in your document.

A.3.2. Unfold/Fold Math Macros

Unfolds/folds the current math macro.

Math macros are described in the Math manual.
A. The User Interface

A.3.3. View Source

Opens a window showing the source code of the actual document, as described in section 6.11.

A.3.4. Update

This menu entry allows you to update the view with your latest changes without opening a new view window.

A.3.5. Split View

This will split LyX’s main window vertically or horizontally. This allows you to view documents the same time to compare them, or to view the same document, but at different positions. You can even split the main window several times to view for example 3 or more documents at the same time. To return to an unsplit view, use the menu Close Tab Group.

A.3.6. Close Tab Group

Closes a split view.

A.3.7. Fullscreen

Using this menu entry or pressing F11 removes the menu bar and all toolbars so that you will see nothing but your text. It furthermore displays LyX’s main window fullscreen. To return from fullscreen to the normal view, press F11, or right-click and turn off the fullscreen mode in the the context menu.

A.3.8. Toolbars

In this menu entry you can set the appearance of the different toolbars. All toolbars and the Command Buffer can be turned on and off. The on state is denoted in the menu with a checkmark. The Review, Table, Math Panels, and Math toolbars can be additionally set to the state automatic that is denoted in the menu with the suffix (auto).

In the on state the toolbar is permanently shown, in the automatic state the toolbar is only shown when the cursor is in a certain environment or when a certain feature is enabled. That means that the review toolbar will only be shown when change tracking is activated, the math and table toolbars are only shown when the cursor is inside a formula or table, respectively.

LyX’s toolbars and their buttons are explained in section A.9.
A.4. The Insert Menu

A.4.1. Math
Inserts math constructs that are explained in chapter 5 and the Math manual.

A.4.2. Special Character
Here you can insert the following characters:

- **Symbols** Inserts any character that can be output by your \LaTeX{} system. Therefore the number of character categories in this dialog and the available characters depend on the \LaTeX{}-packages you have installed.
  
  **Note:** Not all characters will be visible in the Symbols dialog because none of the screen fonts that you can set in the preferences dialog (see sec. C.1.2) can display every character.

- **Ellipsis** Inserts an ellipsis: \ldots{}

- **End of Sentence** Inserts an end of sentence dot as described in section 3.9.3.1.

- **Ordinary Quote** Inserts this quote: ’, no matter what quote type you selected in the Document ➤ Settings dialog under Language.

- **Single Quote** Inserts this quote: ‘

- **Protected Hyphen** Inserts a hyphen that is protected from line breaks: -

- **Breakable Slash** Inserts a slash where also a line break can occur: /

- **Menu Separator** Inserts the menu separator sign: \textgreater{}

- **Phonetic Symbols** Creates a formula with a so called tipa inset. Here you can insert commands to create IPA phonetic symbols. For this feature you must have the \LaTeX{}-package tipa installed.
  
  For more information about this feature we refer to the documentation of tipa, [15] and this Wiki-page:
  
  http://wiki.lyx.org/LyX/LinguistLyX

A.4.3. Formatting
Here you can insert the following format constructs:

- **Superscript** Inserts a superscript: test\textsuperscript{a,b}

- **Subscript** Inserts a subscript: test\textsubscript{3x}

- **Protected Space** Inserts a protected space that is described in section 3.5.1.
A. The User Interface

**Inter-word Space** Inserts an inter-word space that is described in section 3.5.2.1.

**Thin Space** Inserts a thin space that is described in section 3.5.2.2.

**Horizontal Space** Inserts horizontal space, see section 3.5.2 for a description.

**Horizontal Line** Inserts a horizontal line, see section 3.5.7 for a description.

**Vertical Space** Inserts vertical space, see section 3.5.3 for a description.

**Hyphenation Point** Inserts a hyphenation point, see section 3.9.2.

**Ligature Break** Inserts a ligature break, see section 3.9.4.

**Ragged Line Break** Inserts a forced line break, see section 3.5.6.

**Justified Break** Inserts a forced line break that furthermore stretches the broken text line to the page border, see section 3.5.6.

**New Page** Inserts a forced page break, described in section 3.5.5.

**Page Break** Inserts a forced page break that furthermore stretches the broken text page to the page border, described in section 3.5.5.

**Clear Page** Inserts a clear page break, described in section 3.5.5.1.

**Clear Double Page** Inserts a clear doublepage break, described in section 3.5.5.1.

### A.4.4. List / TOC

Various lists can be inserted with this menu entry. The table of contents, the algorithm, figures, and tables list are described in section 6.2. The index list is described in section 6.6, the nomenclature in section 6.7, and the BibTeX bibliography in section 6.5.2.

### A.4.5. Float

To insert floats, described in section 4.6.

### A.4.6. Note

To insert notes, described in section 4.1.

### A.4.7. Branch

Inserts branch insets as described in section 6.8.
A.4.8. Custom Insets

Inserts document class-specific insets. Such insets only exist when they are defined in the layout file for a certain document class. An example is the document class “article (Elsevier)” with three custom insets. The section *Flex insets and InsetLayout* of the *Customization* manual explains how custom insets are defined.

A.4.9. File

This menu entry allows you to insert or include the contents of other \LaTeX{} files in your document. How you can do this is explained in detail in the chapter *External Stuff* of the *Embedded Objects* manual.

A.4.10. Box

Inserts a minipage box that is described section 4.7. All box types supported by \LaTeX{} are explained in detail in the chapter *Boxes* of the *Embedded Objects* manual.

A.4.11. Citation

Inserts a citation as described in section 6.5.

A.4.12. Cross-Reference

Inserts a cross-reference as described in section 6.1.

A.4.13. Label

Inserts a label as described in section 6.1.


Inserts a caption in a float or longtable. Floats are described in section 4.6, captions in longtables are described in the section *Longtable Captions* of the *Embedded Objects* manual.

A.4.15. Index Entry

Inserts an index entry as described in section 6.6.

A.4.16. Nomenclature Entry

Inserts a nomenclature entry as described in section 6.7.
A. The User Interface

A.4.17. Table
Inserts a table. Tables are described in section 4.5.

A.4.18. Graphics
Inserts a graphic. Graphics are described in section 4.4.

A.4.19. URL
Inserts an URL as described in section 6.3.1.

A.4.20. Hyperlinks
Inserts a hyperlink as described in section 6.3.2.

A.4.21. Footnote
Inserts a footnote, see section 4.2.

A.4.22. Marginal Note
Inserts a marginal note, see section 4.3.

A.4.23. Short Title
Inserts a short title, see section 3.3.4.4.

A.4.24. \TeX Code
Inserts a \TeX Code box, see section 6.10.1 for a description.

A.4.25. Program Listing
Inserts a program listings box. Program listings are explained in the chapter Program Code Listings of the Embedded Objects manual.

A.4.26. Date
Inserts the actual date. The format depends on the date format of the language that is used for \LaTeX’s menus. \LaTeX offers various ways to insert a date which are explained and also compared in the section External Material of the Embedded Objects manual.
A.5. The Navigate Menu

This menu lists the existing chapters, sections, figures, tables, etc. of the current document. This allows you to navigate easily through your document.

A.5.1. Bookmarks

With this menu entry you are able to define your own bookmarks. This is useful when you are working on a large document and often have to jump e.g. between section 2.5 and 6.3. To create bookmarks for this example, go to section 2.5 and use the menu Save Bookmark 1. Then go to section 6.3 and use Save Bookmark 2. Now you can jump easily between these sections by using the menu or by the key bindings Ctrl+1 and Ctrl+2.

You can also use bookmarks to jump between several opened documents. The saved bookmarks are valid till the document is closed.

A.5.2. Next Note, Change, Cross-reference

Jump to the next note, change, or cross-reference following the current cursor position.

A.5.3. Go to Label

Only active when the cursor is in front of a cross-reference. Sets the cursor before the referenced label, the same as if you right-click on a cross-reference box.

A.6. The Document Menu

A.6.1. Change Tracking

Change Tracking is described in section 6.14.

A.6.2. LaTeX Log

After running \LaTeX by viewing or exporting a document, this menu will be enabled. It shows the logfile of the used \LaTeX-program.

Here you can see how \LaTeX works in the background. Experts will find in it reasons for \LaTeX-errors.

A.6.3. Outline

Opens the TOC/Outline window as described in section 2.5 and 6.2.1.
A. The User Interface

A.6.4. Start Appendix Here

This menu will start the appendix of the document at the current cursor position as described in section 6.4.

A.6.5. Compressed

Uncompresses the current document.

A.6.6. Settings

The document settings are described in appendix B.

A.7. The Tools Menu

A.7.1. Spellchecker

Spell checking is explained in section 6.12.

A.7.2. Thesaurus

The thesaurus is described in section 6.13.

A.7.3. Statistics

Counts the number of words and characters in the actual document or the highlighted document part.

A.7.4. \TeX\ Information

Shows you a list of the classes and styles installed in your \TeX\-system.

A.7.5. Reconfigure

This menu entry reconfigures \LaTeX; that is, \LaTeX\ looks for \TeX\-packages and needed programs it needs; see also section 1.4.

A.7.6. Preferences

The preferences dialog is described in detail in appendix C.
A.8. The Help Menu

This menu opens the documentation files of LYX in the language of LYX’s menus. The menu \LaTeX Configuration shows a LYX-document with information about the \LaTeX-packages and classes found by LYX (see also section 1.5).

A.9. Toolbars

How to show or hide toolbars is explained in section A.3.8. It is also possible to define custom toolbars. This is described in the Additional Features manual.

A.9.1. Standard Toolbar

The standard toolbar as shown above contains from left to right the following buttons:

- File ▶ New
- File ▶ Open
- File ▶ Save
- File ▶ Print
- Tools ▶ Spellchecker
- Edit ▶ Undo
- Edit ▶ Redo
- Edit ▶ Cut
- Edit ▶ Copy
- Edit ▶ Paste
- Edit ▶ Find & Replace
- Navigate ▶ Bookmarks ▶ Navigate Back
A. The User Interface

Emphasize text, function of the Edit ▷ Text Style dialog

Set text to noun style, function of the Edit ▷ Text Style dialog

Formats text using the current settings in the Edit ▷ Text Style dialog

Insert ▷ Math ▷ Inline Formula

Insert ▷ Graphics

Insert ▷ Table

Toggle outline window on/off, Document ▷ Outline

Toggle math toolbar on/off

Toggle table toolbar on/off

A.9.2. Extra Toolbar

The extra toolbar as shown above contains from left to right the following buttons:

Default

Numbered list

Itemized list

List

Description list

Edit ▷ Increase List Depth

Edit ▷ Decrease List Depth

Insert ▷ Float ▷ Figure
A.9. Toolbars

Insert ▶ Float ▶ Table
Insert ▶ Label
Insert ▶ Cross-Reference
Insert ▶ Citation
Insert ▶ Index Entry
Insert ▶ Nomenclature Entry
Insert ▶ Footnote
Insert ▶ Marginal Note
Insert ▶ Note ▶ LyX Note
Insert ▶ Box
Insert ▶ URL
Insert ▶ TeX
Insert ▶ Math ▶ Macro
Insert ▶ File ▶ Child Document

Edit ▶ Text Style
Edit ▶ Paragraph Settings
Tools ▶ Thesaurus

A.9.3. View / Update Toolbar

The view / update toolbar as shown above contains from left to right the following buttons:

View ▶ DVI
A. The User Interface

A.9.4. Other Toolbars

The change tracking toolbar is explained in section 6.14, the table toolbar in the Embedded Objects manual, the math macro toolbar in the Math manual.
B. The Document Settings

The document settings dialog contains submenus to set properties for the whole document and is called with the menu Document ▶ Settings. You can save your document settings as default with the Save as Document Defaults button in the dialog. This will create a template named default.lyx which is automatically loaded by \textsc{lyx} when you create a new document without using a template.

The different submenus of the dialog are explained in the following.

B.1. Document Class

Here you set the document class, class options, a graphics driver, and a master document. Document classes are described in section 3.1.2. Some classes use some class options by default. If this is the case, they are listed in the field Predefined and you can decide to use them or not. If you don’t exactly know what the default class options are for, it is recommended not to touch them. The graphics driver is used for \textsc{latex}'s graphics, color and page layout packages. When using Default, the default driver for the \textsc{latex}-packages is used. It is recommended that you use the default unless you know what you are doing.\footnote{When you want one of the following drivers dvi2ps, dvialw, dvilaser, dvitops, psprint, pubps, ln you first have to activate them in your \textsc{latex} distribution, see sec.2 in http://tug.ctan.org/get/macros/latex/required/graphics/grfguide.pdf.}

Specifying a master document is necessary when the current document is a child or subdocument. This master document will be used by \textsc{lyx} when the child document is opened without its master. This way child documents are always compilable. More about master and child documents is explained in the section Child Documents of the Embedded Objects manual.

B.2. Modules

Modules are explained in section 3.1.2.2.

B.3. Fonts

The document font settings are described in section 3.7.
B. The Document Settings

B.4. Text Layout

You can specify if paragraphs should be separated by indentations or vertical skips. The line spacing and the number of text columns can here also be specified.

Note that LyX won’t show two columns or the set up line spacing on screen. That’s impractical, often unreadable, and not part of the WYSIWYM concept. However, it will be as you specified it in the output.

The listings settings are explained in the corresponding section in the Embedded Objects manual.

B.5. Page Layout

A description of this menu is given in section 3.1.4 and 3.1.3.

B.6. Page Margins

Here you can adjust the paper margins, see section 3.1.5.

B.7. Language

The document language and quote styles are set here. The encoding specifies how the document content is exported to \LaTeX\ (the LyX file is always encoded in utf8). All characters that cannot be encoded using the specified encoding will be exported as \LaTeX-commands (this can fail if a \LaTeX-command is not known for a particular character).\(^2\)

If you use the option Language Default, LyX determines the encoding of a text part from the language of this text. If the document contains text in more than one language you may get more than one encoding in the \LaTeX\ file. If you do not use this option then the complete document will always use exactly one encoding. Checking this option is the preferred setting unless you use Xe\TeX\ (see below).

LyX also supports Unicode output, which is particularly useful if you need lots of special symbols or non-alphabetic scripts, respectively. If you want to use this (and your \LaTeX\ installation supports Unicode, for that matter), choose one of the four utf8 variants from the list below. Unfortunately the Unicode support of standard \LaTeX\ is quite incomplete, so it is not uncommon that a file with lots of Unicode

\(^2\) The known commands are defined in a text file. You can add commands for unknown symbols to that file yourself, see the Customization manual for details.

\(^3\) Xe\TeX\ is a \TeX\ typesetting engine, an alternative for \LaTeX. It natively supports Unicode while its input file is assumed to be in UTF-8 encoding. More about using LyX with Xe\TeX\ can be found in [17].
symbols works fine with Language Default (when \LaTeX\ uses its list of known \LaTeX\-commands), but does not work with a fixed utf8 encoding (when the list of known \LaTeX\-commands is not used, because all Unicode symbols can be encoded in utf8).

Here is a list with the important encodings:

**Language Default (no inputenc)** Same as Language Default, but the \LaTeX\-package inputenc is not used. When using this, you probably need to load some additional packages manually in the preamble and specify the used encoding for text parts in foreign languages in \LaTeX\ code.

**ASCII** the ASCII encoding, covers only plain English (7-bit ASCII). \LaTeX\ converts all other characters into \LaTeX\ commands, which may result in a big file when lots of \LaTeX\-commands are needed.

**Arabic (CP 1256)** MS Windows code page for Arabic and Farsi

**Arabic (ISO 8859-6)** for Arabic and Farsi

**Armenian (ArmSCII8)** for Armenian

**Baltic (CP 1257)** MS Windows code page for Estonian, Latvian, and Lithuanian, the same as the ISO-8859-13 encoding

**Baltic (ISO 8859-13)** for Estonian, Latvian, and Lithuanian, a superset of the ISO-8859-4 encoding

**Baltic (ISO 8859-4)** (latin 4) for Estonian, Latvian, and Lithuanian, a subset of the ISO-8859-13 encoding

**Central European (CP 1250)** MS Windows code page for ISO 8859-2 (latin2)

**Central European (ISO 8859-2)** (latin 2) covers Albanian, Croatian, Czech, German, Hungarian, Polish, Romanian, Slovak, and Slovenian

**Chinese (simplified) (EUC-CN)** for simplified Chinese, used especially on UNIX OSes, since 2001 this encoding is officially replaced by the encoding GB18030, as GB18030 is not available for \LaTeX\ you should try to use the encoding Unicode (CJK) (utf8)

**Chinese (simplified) (GBK)** for simplified Chinese, is the same as the Windows code page CP 936 except for the Euro currency sign, since 2001 this encoding is officially replaced by the encoding GB18030, as GB18030 is not available for \LaTeX\ you should try to use the encoding Unicode (CJK) (utf8)

**Chinese (traditional) (EUC-TW)** for traditional Chinese

**Cyrillic (CP 1251)** MS Windows code page for Cyrillic

**Cyrillic (ISO 8859-5)** covers Belorussian, Bulgarian, Macedonian, Serbian, and Ukrainian
B. The Document Settings

**Cyrillic (KOI8-R)** standard Cyrillic especially for Russian

**Cyrillic (KOI8-U)** Cyrillic for Ukrainian

**Cyrillic (pt 154)** Cyrillic for Kazakh

**Greek (ISO 8859-7)** for Greek

**Hebrew (CP 1255)** MS Windows code page for Hebrew, a superset of the ISO-8859-8 encoding

**Hebrew (ISO 8859-8)** for Hebrew

**Japanese (CJK) (EUC-JP)** EUC-JP encoding for Japanese, uses the \texttt{\LaTeX}-package \texttt{CJK}, when using this, set the document language to Japanese (CJK)

**Japanese (CJK) (JIS)** JIS encoding for Japanese, uses the \texttt{\LaTeX}-package \texttt{CJK}, when using this, set the document language to Japanese (CJK)

**Japanese (non-CJK) (EUC-JP)** EUC-JP encoding for Japanese, uses the \texttt{\LaTeX}-package \texttt{japanese}, when using this, set the document language to Japanese

**Japanese (non-CJK) (JIS)** JIS encoding for Japanese, uses the \texttt{\LaTeX}-package \texttt{japanese}, when using this, set the document language to Japanese

**Japanese (non-CJK) (SJIS)** SJIS encoding for Japanese, uses the \texttt{\LaTeX}-package \texttt{japanese}, when using this, set the document language to Japanese

**Korean (EUC-KR)** for Korean

**Southern European (ISO 8859-3)** (latin 3) covers Esperanto, Galician, Maltese, and Turkish

**South-Eastern European (ISO 8859-16)** (latin 10) covers Albanian, Croatian, Finnish, French, German, Hungarian, Irish Gaelic, Italian, Polish, Romanian, Slovenian, is designed to cover many languages and characters with diacritics

**Thai (TIS 620-0)** for Thai

**Turkish (ISO 8859-9)** (latin 5) for Turkish, is like the ISO-8859-1 encoding where the Icelandic letters are replaced by Turkish ones

**Unicode (CJK) (utf8)** Unicode utf8 with the \texttt{\LaTeX}-package \texttt{CJK} (for the languages Chinese, Japanese and Korean)

**Unicode (Xe\LaTeX) (utf8)** Unicode utf8 to be used with Xe\LaTeX, which uses Unicode directly, without the help of the \texttt{\LaTeX}-package \texttt{inputenc} \footnote{More about using LyX with Xe\LaTeX can be found in [17].}
B.8. Numbering & TOC

You can adjust here the numbering depth of section headings and the section depth in the table of contents as described in section 3.3.4.3.

B.9. Bibliography

You can specify here a citation style using the \LaTeX-packages natbib or jurabib. For a further description see section 6.5.

B.10. PDF Properties

The PDF properties are explained in section 6.9.

B.11. Math Options

These options will force \LaX to use the \LaTeX-packages amsmath and esint or to use them automatically when they are needed.

amsmath is needed for many constructs, so when you get \LaTeX-errors in formulas, ensure that you have enabled AMS.
esint is used for special integral characters.

B.12. Float Placement

The float placement options are described in section 4.6.3.
B. The Document Settings

B.13. Bullets

Here you can adjust the characters for the itemize levels. The itemize environment is described in section 3.3.6.2.

B.14. Branches

Branches are described in section 6.8.

B.15. LaTeX Preamble

In this text field are entered commands to load special \LaTeX-packages or to define \LaTeX-commands. The preamble is a thing for \LaTeX-experts. You should not enter commands here until you exactly know what you are doing.

An introduction to the \LaTeX-syntax is given in section 6.10.2.
C. The Preferences Dialog

The preferences dialog is called with the menu Tools > Preferences. It has the following submenus.

C.1. Look and Feel

C.1.1. User Interface

C.1.1.1. User Interface File

Note: You have to restart LyX before the change of the user interface file take effect.

The appearance of the menus and toolbars can be changed by choosing a user interface (ui) file. An ui-file is a text file where the toolbars and menus are listed. The toolbar buttons and menu entries are specified in the files stdtoolbars.inc and stdmenus.inc, respectively. Both files are loaded by the default.ui file. To create your own menu and toolbar layout, start with a copy of these files and edit the entries.

The syntax of the .inc-files is straightforward: The Menubar, Menu and Toolbar entries must be ended with an explicit End. They may contain Submenus, Items, OptItems, Separators, Icons, and in the case of the “file” menus a Lastfiles entry. The syntax for the entries is:

Item “menu or button name” “LyX-function”

All LyX-functions are listed in [27].

An example: Assuming you use the menu Navigate > Bookmarks quite often and therefore want six available bookmarks, you can add the line

Item “Save Bookmark 6” “bookmark-save 6”

to the navigate menu section in the .inc-file to have the sixth bookmark.

C.1.1.2. Automatic help

The option Enable tool tips in main work area enables tool tips showing the content of closed insets like index entries or footnotes.

C.1.1.3. Session

With the option Allow saving / restoring of window layout and geometries LyX’s main window will be opened with the size and layout that was used in the last LyX session.
C. The Preferences Dialog

The option **Restore cursor positions** sets the cursor to the position in the file where it has been the last time.

The option **Load opened files from last session** opens all files that were opened in the last \LaTeX\ session.

C.1.1.4. Documents

When the option **Backup documents** is set, you can specify the time between backup saves.

**Maximum last files** is the number of last opened files that \LaTeX\ should display in the menu `File > Open Recent`.

When the option **Open documents in tabs** is not used, then every file will be opened in its own new instance of \LaTeX\.

C.1.2. Screen Fonts

These fonts are used to display your documents on the screen.

**Note:** This section only deals with the fonts inside the \LaTeX\ window. The fonts that appear on the output are independent from these fonts, and set in the menu `Document > Settings > Fonts`.

By default, \LaTeX\ uses **Times** as roman (serif) font, **Arial** or **Helvetica** (depends on the system) as **sans serif** font, and **Courier** as **typewriter** font.

You can change the font size with the **Zoom** setting. You can also change the font zoom outside the preferences dialog for the current \LaTeX\ session by pressing Ctrl and scrolling the mouse wheel.

**Screen DPI** is the screen resolution in dpi (dots per inch). The Font Sizes are calculated as letter height in units of points. 72 points have the size of 1 inch, see Appendix D.

The default **Font Sizes** are the same as if a document font size of 10 pt would be used. The sizes are explained in detail in section 3.7.2.

With the option **Use Pixmap Cache to speed up font rendering** enabled, \LaTeX\ needs to redraw the screen less often. This results in better performance, especially on slow systems. On the other hand, the characters might look more fuzzy on screen. So whether you enable this or not depends on whether you prefer speed over aesthetics. Note that the Pixmap Cache is only available and useful under Mac OS and Windows.

C.1.3. Colors

You can here change all the colors used by \LaTeX\. Choose an item in the list and use the **Alter** button.

C.1.4. Graphics

Here you can specify how graphics inside \LaTeX\ are displayed.
Instant Preview enables previewing snippets of your document. This feature is described in section 6.11.

C.2. Editing

C.2.1. Control

C.2.1.1. Editing

The option **Cursor follows scrollbar** sets the cursor to the top of the currently displayed document part when scrolling.

The option **Sort environments alphabetically** sorts the entries in the pull-down box for the paragraph environments.

The option **Group environments by their category** groups the entries in the pull-down box for the paragraph environments.

The math macro editing option determines the editing style, see the section *Math Macros* of the *Math* manual.

C.2.1.2. Fullscreen

Here you can specify what is hidden in the fullscreen mode. The option **Limit text width** specifies the width of the text in fullscreen mode. This way you can display the text smaller than the screen, the text appears then centered.

C.2.2. Shortcuts

C.2.2.1. Bind File

Bindings are used to bind a \LaTeXX-function to a key. Several binding files are available:

- **cua.bind** typical set of PC keyboard shortcuts
- **(x)emacs.bind** set of bindings like they are used in the editor programs *Emacs* (XEmacs)
- **mac.bind** set of bindings for MacOS systems.

There are also bind-files designed for special document classes, like *broadway.bind*, and bind files for special languages. The name of language bind-files begin with a language code, e.g. “pt” for Portuguese. If you use \LaTeXX in a certain language, \LaTeXX will try to use the appropriate bind-file.

Some bind-files, like *math.bind*, have only a small scope. When looking at the the end of the file *cua.bind*, you can see that they are included to keep the overview in the bind-file.
C. The Preferences Dialog

C.2.2.2. Editing Shortcuts

To add new or modify existing keybindings to your own taste you can use the table in the dialog that lists all \texttt{LYX}-functions and the bound shortcuts. To find functions easily, they are grouped by categories and the dialog provides the field \textbf{Show keybindings containing}. In this field you can insert a keyword for a function you want to edit. Insert there for example as keyword “paste” and you get the 4 different existing shortcuts for the 3 different functions that contain “paste” in their name. As you can see, one function can have more than one shortcut. All \texttt{LYX}-functions are also listed in the file \texttt{LYX Functions} that you will find in the \texttt{Help} menu.

To add e.g. the shortcut \texttt{Alt+Q} for the function \texttt{textstyle-apply}, select the function and press the \texttt{Modify} button. A dialog pops up where you can add the shortcut by using it. So press \texttt{Alt+Q} to define the shortcut. Modifying an existing shortcut is done the same way. You can also bind multiple functions to one shortcut by modifying an existing binding and adding the different function names as a semicolon separated list. \texttt{LYX} will then use the first function that is enabled in the current document part. The binding for the function \texttt{command alters} is an example of this.

Alternatively you can also edit shortcuts by modifying bind-files with a text editor. The syntax of the entries is:

\begin{verbatim}
\bind “key combination” “LYX-function”
\end{verbatim}

C.2.3. Keyboard / Mouse

Normally keyboard settings have to be done in a menu of your operating system. For the case that this is not possible, \texttt{LYX} provides keyboard maps. If you have e.g. a Czech keyboard but want to write with it like with a Romanian one, you can use the keyboard map file named \texttt{romanian.kmap}.

\textbf{Note:} Keyboard maps are only provided as makeshift and don’t work on all systems.

Besides this, you can specify here the \texttt{Wheel scrolling speed}. The standard value is 1.0, higher values speed up the scrolling, lower ones slow it down.

C.2.4. Input Completion

Input completion is described in sec.2.6. The completion options for math do the same as the corresponding options for text. With the general options you can define the delay time for the inline and popup completion and choose if long completions should be abbreviated or not.

C.3. Paths

\textbf{Working directory} This is \texttt{LYX}’s working directory. It is the default when you \texttt{Open}, 
\texttt{Save} or \texttt{Save as} files.
C.4. Identity

Document templates This directory will be opened when you use the menu File\-New from Template.

Example files This directory will be opened when you use the button Examples in the File\-Open dialog.

Note: The Examples button does not exist when using LyX on MacOS and Windows systems.

Backup directory Backup copies will be saved to this directory. When no directory is given but backups are enabled as described in section C.1.1.4, the Working directory will be used to save the backups.

The backup files have the ending “.lyx~”.

LyXServer-Pipe Here you can enter the name of a Unix-pipe. This pipe is used to send data from external programs to LyX.

Note: This feature doesn’t work on Windows systems.

Temporary directory Temporary files will be saved in this directory.

PATH prefix This field contains a list of paths to external programs. When LyX needs to use an external program, it looks in this list where to find it on the system. The path list is automatically set up on Windows and Mac systems when LyX is configured, so that you normally don’t have to modify it. On Unix/Linux systems, the path list will need to be set only if there are external programs you wish to use that are not in your normal system path ($PATH).

C.4. Identity

Here you can insert your name and email address. The identity will be used when you have enabled change tracking as described in section 6.14, to mark changes you make as yours.

C.5. Language Settings

C.5.1. Language

User Interface language Here you can select the language of LyX’s menus. Unfortunately this doesn’t work on Mac and Windows. It works on Linux for the languages into LyX’s menus and dialogs are translated. You find the actual translation status here: http://www.lyx.org/I18n

Default language is the language used in new documents

Language package is a \texttt{\LaTeX}-command to load a \texttt{\LaTeX}-package that handles language issues. The default is the \texttt{\LaTeX}-command \texttt{\LaTeXe{\texttt{babel}}} that loads
the package \texttt{babel}.\footnote{For an introduction to the \LaTeX-Syntax, have a look at section 6.10.}

\texttt{babel} automatically translates in the background the text labels of documents to the document language. A text label is, for instance, the word “Table” at the beginning of every table caption.

\textbf{Command start} When a special \LaTeX-package is needed to write in a certain document language, you can here specify the command to start the package. An example is the start command \texttt{\begin{arabtext}} that is needed to write Arabic using the package \texttt{ArabTEX}, see [18]. The default is the \texttt{babel} command \texttt{\selectlanguage{$$lang$}}.

\textbf{Command end} Counterpart to \textbf{Command start}. Some packages, like the default, don’t have an end command since the start command toggles the package on and off.

\textbf{Use babel} Whether \texttt{babel} is used or not.

\textbf{Global} When this option is set, the languages used in the document will be added as options to the document class options, so that they can be used by all \LaTeX-packages. Otherwise they will only be used as options for the \texttt{babel} package.

\textbf{Auto begin} When this option is set, the documents starts with the chosen document language. When this option is not set, the \textbf{Command start} is explicitly set to the beginning of the document in the \LaTeX-output. This assures that the correct language is used when you use another \textbf{Command start} than the default.

\textbf{Auto end} Counterpart to \textbf{Auto begin}. When it is not set, the \textbf{Command end} is set to the end of the document.

\textbf{Mark foreign languages} Text marked formatted in a language different from the document language will be underlined blue.

\textbf{Enable RTL support} Enables the use of languages, written from right to left (RTL), like Arabic, Hebrew, Farsi.

\textbf{Cursor movement} When writing RTL, you can define if the left and right arrow keys move the cursor visually to the left or right, respectively, or logically. Logical means that the cursor is moved to the left when pressing the right arrow key and the cursor is inside text in an RTL language.

\section*{C.5.2. Spellchecker}

The spellchecker settings are explained in section 6.12
C.6. Outputs

C.6.1. Printer

Default printer Here you can specify the name of your default printer. The name will be used when the Printer command is executed.

Note: You can leave this field empty on Windows systems because it has no effect.

Adapt output to printer This option works only for the Printer command “dvips”. It activates a configuration file for dvips. This is an option only for dvips experts.

Printer command is the command \LaTeX{} uses for printing. The default is on most systems dvips.

Printer Command Options Here you can specify printer options. A list of printer options with explanations can be found in the documentation of the program that provides the Printer command you are using.

C.6.2. Date Format

The date format can be one or a mixture of the formats listed here:

http://unixhelp.ed.ac.uk/CGI/man-cgi?date

For example the format

\%d/\%m/\%y

prints the date as day/month/year.

C.6.3. Plain Text

Output line length sets the maximum number of characters printed in one line when using the menu File > Export > Plain text. Setting the line length to 0 means all text is printed in one endless line.

roff command defines an additional command used to produce better ASCII tables with the groff/troff/nroff UNIX-commands (refer to their manuals for more information about them). Setting this as empty tells \LaTeX{} to use the internal formatter.

C.6.4. \LaTeX{}

TeX encoding This is the default encoding of the document font. T1 is the default and covers western languages and symbols. T2A, T2B, T2C, LCY, and X2 are for Cyrillic. Combinations of the encodings are possible, like "T1, T2B". The font encoding is normally automatically loaded by the language packages \LaTeX{} sets up in the background. So there is no need to change the default encoding.
C. The Preferences Dialog

**Default paper size** This is the paper size that is used for new documents. The Default value depends on your \LaTeX-system setup.

You can also specify here commands with parameters for the listed applications. But before you change something, it is strongly recommended to read the manuals of the applications. Currently the following commands can be set:

**CheckTeX command** Command for the program Check\TeX that is described in the section Checking \TeX of the Additional Features manual.

**BibTeX command** Command for the program Bib\TeX that generates the bibliography, see section 6.5.2.

**Index command** Command for the program that generates the index, see section 6.6.6.

**Nomenclature command** Command for the program that generates the nomenclature, see section 6.7.5.

**DVI viewer paper size options** They only have an effect when the program xdvi is used as DVI-viewer.

There are additionally the following options:

**Use Windows-style paths in \LaTeX files** Uses paths in the notation of Windows, that means that “\” is used instead of “/” to separate folders. This option is enabled by default when you use \LaTeX on Windows.

**Reset class options when document classes changes** Removes all manually set document class options in the Document\>Settings dialog when changing the document class.

C.7. File Handling

C.7.1. Converters

Here you find the list of defined converter commands to convert material from one format to another. You can modify them or create new ones. To modify a converter, select it, change the entry of the field Converter and/or Extra flag, and press the Modify button. To create a new converter, select an existing one, select a different format in the From format drop-down list, modify the Converter field, and press the Add button.

When the Converter File Cache is enabled, conversions will be cached as long as specified in the field Maximum Age (in days). This means that images don’t need to be converted again when you reopen a document; the converted images from the cache will be used instead.

More about converters, like the variables and flags that can be used in the converter definition, is described in the section Converters of the Customization manual.
C.7.2. File Formats

Here you find the list of defined file formats that \LaTeX{} can handle. You can modify the viewer and editor program that should be used for certain formats.

More about formats, like the options that can be used in the format definition, is described in the section *Formats* of the *Customization* manual.

Since all conversions from one format to another take place in \LaTeX{}'s temporary directory, it is sometimes necessary to modify a file before copying it to the temporary directory in order that the conversion may be performed. This is done by specifying a *Copier*. More about this is described in the section *Copiers* of the *Customization* manual.
C. The Preferences Dialog
D. Units available in LyX

To understand the units described in this documentation, D.1 explains all units available in LyX.

Table D.1.: Units

<table>
<thead>
<tr>
<th>unit</th>
<th>name/description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm</td>
<td>millimeter</td>
</tr>
<tr>
<td>cm</td>
<td>centimeter</td>
</tr>
<tr>
<td>in</td>
<td>inch</td>
</tr>
<tr>
<td>pt</td>
<td>point (72.27 pt = 1 in)</td>
</tr>
<tr>
<td>pc</td>
<td>pica (1 pc = 12 pt)</td>
</tr>
<tr>
<td>sp</td>
<td>scaled point (65536 sp = 1 pt)</td>
</tr>
<tr>
<td>bp</td>
<td>big point (72 bp = 1 in)</td>
</tr>
<tr>
<td>dd</td>
<td>didot (72 dd ≈ 37.6 mm)</td>
</tr>
<tr>
<td>cc</td>
<td>cicero (1 cc = 12 dd)</td>
</tr>
<tr>
<td>Scale%</td>
<td>% of original image width</td>
</tr>
<tr>
<td>text%</td>
<td>% of text width</td>
</tr>
<tr>
<td>col%</td>
<td>% of column width</td>
</tr>
<tr>
<td>page%</td>
<td>% of paper width</td>
</tr>
<tr>
<td>line%</td>
<td>% of line width</td>
</tr>
<tr>
<td>theight%</td>
<td>% of text height</td>
</tr>
<tr>
<td>pheight%</td>
<td>% of paper height</td>
</tr>
<tr>
<td>ex</td>
<td>height of letter $x$ in current font</td>
</tr>
<tr>
<td>em</td>
<td>width of letter $M$ in current font</td>
</tr>
<tr>
<td>mu</td>
<td>math unit (1 mu = 1/18 em)</td>
</tr>
</tbody>
</table>
D. Units available in LyX
E. Credits

The documentation is a collaborative effort between many different people (and we would encourage people to contribute!).

- **Alejandro Aguilar Sierra**
- **Amir Karger**
- **David Johnson**
- **Hartmut Haase**
- **Ignacio García**
- **Ivan Schreter**
- **John Raithel**
- **John Weiss**
- **Lars Gullik Bjønnes**
- **Matthias Ettrich**
- **Matthias Zenker**
- **Rich Fields**
- **Pascal André**
- **Paul Evans**
- **Paul Russel**
- **Robin Socha**
- **Uwe Stöhr**
- The \LaTeX Team: [Credits]
E. Credits

The bibliography on the following page was created with the Bibliography environment.
Bibliography


[5] The *T*\TeX* Catalogue: 

[6] The *L*\TeX* FAQ: 
http://www.tex.ac.uk/cgi-bin/texfaq2html

[7] Documentation of the program Bib* \TeX*: 

[8] Documentation how to use the program Bib* \TeX*: 

[9] Documentation of the program makeindex: 
http://tug.ctan.org/indexing/makeindex/doc/manpages.dvi

[10] Documentation of the program xindy: 
http://www.xindy.org/documentation.html

[11] Documentation of the *L*\TeX*-package caption: 

[12] Documentation of the *L*\TeX*-package fancyhdr: 
[13] Documentation of the \LaTeX-package hyperref:

[14] Documentation of the \LaTeX-package nomencl:

[15] Documentation of the \LaTeX-package tipa:

[16] Documentation of the \LaTeX-package wrapfig:
ftp://tug.ctan.org/pub/tex-archive/macros/latex/contrib/wrapfig/wrapfig.sty

[17] Wiki-page how to use LyX with Xe\TeX:
http://wiki.lyx.org/LyX/XeTeX

[18] Wiki-page how to set up LyX for Arabic:
http://wiki.lyx.org/Windows/Arabic

[19] Wiki-page how to set up LyX for Armenian:
http://wiki.lyx.org/Windows/Armenian

[20] Wiki-page how to set up LyX for Farsi:
http://wiki.lyx.org/Windows/Farsi

[21] Wiki-page how to set up LyX for Hebrew:
http://wiki.lyx.org/Windows/Hebrew

[22] Wiki-page how to set up LyX for Japanese:
http://wiki.lyx.org/Windows/Japanese

[23] Wiki-page how to set up LyX for Latvian:
http://wiki.lyx.org/Windows/Latvian

[24] Wiki-page how to set up LyX for Lithuanian:
http://wiki.lyx.org/Windows/Lithuanian

[25] Wiki-page how to set up LyX for Mongolian:
http://wiki.lyx.org/Windows/Mongolian

[26] Wiki-page how to set up LyX for Vietnamese:
http://wiki.lyx.org/Windows/Vietnamese

[27] Wiki-page with a list of all available LyX-functions:
http://wiki.lyx.org/LyX/LyxFunctionList

[28] Wiki-page about new features in LyX 1.6.0:
http://wiki.lyx.org/LyX/NewInLyX16
Bibliography 2


The above bibliography is created from a BibTeX-database.
Nomenclature

a  dummy entry for the character a
Alt  Alt or Meta key
Ctrl  Control key
Esc  Escape key
Shift  Shift key
Tab  Tabulator key
\sigma  dummy entry for the character \texttt{sigma}
Index

Abstracts, 65
AMS math, 118
Appendix, 122

Backup
  Directory, 165
  Documents, 162

Bibliography, 66, 122
  BibTeX, 123
  Databases, 123
  Layout, 124

Bindings, see Key Bindings

Boxes, 147
Branches, 129

Captions, 147
Change Tracking, 134
Character count, 150
Character Styles, 82

Color
  Change tracking, 135
  LyX screen, 162
  Text, 85

Converters, 168
Copiers, 169
Cross references, 119
Customization
  of menus, 161
  of toolbars, 161

Date Format, 167

Document
  Branches, 129
  Change Tracking, 134
  Classes, 49
  Font, 81
  Language, 136
  Layout, 52
  Margins, 53
  Modules, 51
  Paper size, 52
  Preview, 88, 132
  Settings, 49, 51–54, 58, 61, 78, 81, 91, 118, 124, 136, 155
  Title, 56
  Types, 49

Dummy entries
  italic page number:, 126
  maïs, 125
  maison, 125
  maître, 125
  my entry, 126
  This is an italic dummy entry, 126

Editing, 44
EPS, see Image formats
External Material, 147

File formats, 86, 169
  ASCII, 86
  DVI, 87
  LaTeX, 86
  PDF, 87
  PostScript, 87
File handling, 168
File Operations, 43
Find, 44

Floats, 102
  Algorithm floats, 104
  Captions, 102
  Figure floats, 103
  Placement, 106
Index

Rotating, 105
Table floats, 104
Text Wrap Floats, 105
Font
Screen, 162
Size, 81
Types, 80
Fonts
Bitmap-, 80
Vector-, 80
Footnotes, 95
Formulas, see Math
GIF, see Image formats
Glossary, see Nomenclature
Graphics, 96
Horizontal lines, 80
Hyperlinks, 121
Hyphenation, 89
Hyphens, 89
Images, 96
Formats, 97
Settings grouping, 98
Index
Cross referencing, 125
Entry layout, 126
Entry order, 125
Grouping, 124
Page ranges, 125
Program, 126
Index generation, 124
Input completion, 46, 164
Instant preview, 132
International support, 136
JPG, see Image formats
Key Bindings, 47, 163
Editing, 164
Keyboard Map, 164
Language
Encoding, 156
Options, 136
Phonetic symbols, 145
Settings, 165
\LaTeX Syntax, 131
\LaTeX-packages
amsmath, 159
babel, 89
caption, 131, 175
CJK, 158
dvipost, 135
esint, 159
fancyhdr, 52, 175
hyperref, 130, 176
inputenc, 157, 159
japanese, 158
jurabib, 159
natbib, 159
nomencl, 127, 176
prettyref, 120
preview-latex, 132
tipa, 145, 176
ucs, 159
wrapfig, 105, 176
\LaTeX-packages
aeguill, 81
pstricks, 87
setspace, 54
Layout Modules, 117
Letters, 64
Ligatures, see Typography
Line breaks, 79
Lists, 60
Description, 62
Enumerate, 61
Itemize, 60
\LyX list, 63
Longtables, 99
Caption, 147
\LyX
Basics, 49
Proper names, 92
Reconfigure, see Reconfiguration of \LyX
Marginal notes, 96
Index

Margins, 53
Math, 109
   Accents, 112
   AMS, 118
   Arrays, 113
   Basics, 109
   Brackets, 113
   Delimiters, 113
   Exponents, 110
   Font sizes, 117
   Formula numbering, 115
   Fractions, 110
   Functions, 112
   Integrals, 111
   Limits, 111
   Macros, 116
   Matrices, 113
   Multi-line Equations, 113, 118
   Navigating, 109
   Referencing formulas, 115
   Roots, 111
   Spaces, 111
   Subscripts, 110
   Sums, 111
   Symbols, 111
   Text, 117
   Typefaces, 116

Menu
   Document, 149
   Edit, 142
   File, 139
   Help, 151
   Insert, 145
   Navigate, 149
   Tools, 150
   View, 143

Minipages, 108
Mouse Operations, 45

Navigating, 46
Nesting
   Environments, 67
   Examples, 71
   Tables etc., 69

Nomenclature, 127
   Layout, 127
   Options, 128
   Printing, 128
   Program, 129
   Sort order, 127

Notes, 95

Outline, 120

Page breaks
   Clear, 79
   Forced, 79

Paragraph
   Environments, 55
   Indentation, 53
   Line spacing, 54
   LyX code, 66
   Separation, 54
   Settings, 143
   Verse, 59

Paragraph environments, 55–67

Paste, 44
Paths, 164, 168
PDF, 87, 97
PDF properties, 130
Phonetic symbols, 145
PNG, see Image formats
Poetry, 59
Preferences, 43, 161
Printer, 167
Program listings, 148
Punctuation marks, 90

Quotation, 59
Quotes, see Typography

Reconfiguration of LyX, 150
Reconfiguration of LyX, 43, 141, 143
Redo, 45
Replace, 44

Section headings, 56
   Numbered, 56

183
Index

Short titles, 58
Unnumbered, 57

Settings
Color, 162
Date format, 167
Editing, 163
Graphics, 162
Keyboard Map, 164
Language, 165
\LaTeX, 167
Paths, 164, 168
Printer, 167
Shortcuts, 163
Shortcuts, see Key Bindings

Spaces
Inter-word, 76
Inter-word, 90
Protected, 75
Thin, 76
Thin, 90
Spacing, 75
Fills, 76, 78
Horizontal, 75
Phantom, 77
Vertical, 78
Spell checking, 133
SVG, see Image formats

Table of contents, 120
Tables, 98
Cells, 101
Longtables, 99
\TeX Code, 131
\TeX Information, 150
Text Style, 82, 83
Thesaurus, 134
Toolbar, 144
Extra, 152
Macro, 154
Review, 135
Standard, 151
Table, 154
View / Update, 153
Typography, 89

Ligatures, 91
Quotes, 91
Units, 92
Widows and orphans, 93
Undo, 45
Units, 171
URLs, 121
Word completion, see Input completion
Word count, 150
Additional LyX Features

by the LyX Team*

August 17, 2009

*Principal maintainer of this file is Richard Heck. If you have comments or error corrections, please send them to the LyX Documentation mailing list, <lyx-docs@lists.lyx.org>.
## Contents

1 Introduction .................................................. 185

2 LyX and \LaTeX ........................................... 187
   2.1 How LyX Uses \LaTeX .................................. 187
   2.2 Translating \LaTeX files into LyX ..................... 188
   2.3 Inserting \TeX Code into LyX Documents ............... 188
   2.4 LyX and the \LaTeX Preamble ......................... 189
      2.4.1 About the \LaTeX Preamble ...................... 189
      2.4.2 Changing the Preamble ........................... 190
      2.4.3 Examples ........................................ 190
         2.4.3.1 Example #1: Offsets ...................... 190
         2.4.3.2 Example #2: Labels ...................... 191
         2.4.3.3 Example #3: Paragraph Indentation ....... 191
         2.4.3.4 Example #4: This Document ............... 192
   2.5 LyX and \LaTeX Errors ................................ 192

3 Supplemental Tools ........................................ 195
   3.1 Customizing Bibliographies with Bib\TeX ................ 195
      3.1.1 Alternative Citation Styles ................... 195
      3.1.2 Sectioned Bibliographies ...................... 195
      3.1.3 Multiple Bibliographies ....................... 196
   3.2 Multipart Documents ................................... 196
      3.2.1 General Operation ............................... 196
      3.2.2 Cross-References Between Files ................. 197
      3.2.3 Bibliography Lists in all Subdocuments ........ 197
   3.3 Fancy Headers and Footers ............................ 198
   3.4 Itemize Bullet Selection ............................. 199
      3.4.1 Introduction ................................... 199
      3.4.2 How it looks ................................... 199
      3.4.3 How to use it .................................. 200

4 The LyX Server ............................................... 201
   4.1 Introduction ........................................ 201
   4.2 Starting the LyX Server .............................. 201
   4.3 Normal communication ................................ 202
   4.4 Notification ........................................ 202
5.14.2 article (koma-script), report (koma-script), and book (koma-script) ........................................ 238
5.14.3 letter (koma-script) .................................................. 239
5.14.4 The new letter class: letter (koma-script v.2) .................. 242
5.14.5 Problems .............................................................. 242
5.15 LATEX8 (IEEE Conference Papers) ............................ 243
  5.15.1 Introduction ......................................................... 243
  5.15.2 Getting Started .................................................... 243
  5.15.3 Supported Environments ...................................... 243
  5.15.4 Differences Between Screen and Paper ...................... 244
5.16 Memoir ................................................................. 244
  5.16.1 Overview ............................................................ 244
  5.16.2 Basic features and restrictions .................................. 244
  5.16.3 Extra features ..................................................... 246
5.17 The mw Classes ....................................................... 246
5.18 Paper ........................................................................ 247
5.19 REVTEX4 ................................................................. 247
  5.19.1 Installation ........................................................... 247
  5.19.2 Preamble Matter ................................................... 248
  5.19.3 Layouts ............................................................... 248
  5.19.4 Important Notes ................................................... 248
  5.19.5 Drawbacks .......................................................... 248
5.20 Springer Journals (svjour) ............................................ 249
  5.20.1 Description .......................................................... 249
  5.20.2 New styles .......................................................... 249
  5.20.3 Supported journals ............................................... 249
  5.20.4 Credits .............................................................. 250
  5.20.5 Bugs .................................................................. 250
5.21 Slides [aka SLiTEX] ..................................................... 250
  5.21.1 Introduction .......................................................... 250
  5.21.2 Getting Started .................................................... 250
  5.21.3 Paragraph Environments ...................................... 251
    5.21.3.1 Supported Environments .................................. 251
    5.21.3.2 Quirks of the New Environments ....................... 252
  5.21.4 Making a Presentation with Slide, Overlay and Note ...... 253
    5.21.4.1 Using the Slide Environment ........................... 253
    5.21.4.2 Using Overlay with Slide ............................... 253
    5.21.4.3 Using Note with Slide .................................... 255
  5.21.5 The slides Class Template File ............................... 256

6 LyX Features needing Extra Software .......................... 257
6.1 Checking \TeX ............................................................ 257
  6.1.1 Introduction .......................................................... 257
  6.1.2 How to use it ........................................................ 258
Contents

7.1.3.1 Two columns ........................................ 276
7.1.3.2 Multiple columns .................................... 276
7.1.3.3 Columns inside columns ............................... 277
7.2 Numbering in Enumerate ................................... 277
7.3 Dropped Capitals ............................................ 278
7.4 Non-standard Paragraph Shapes ............................ 279
7.5 Summary ...................................................... 279
1 Introduction

This manual is essentially Part II of the User’s Guide. The reason for separating this document out is simple: the User’s Guide is already quite lengthy, and it contains information on all of the basic features one needs to know in order to prepare most documents. However, the LyX Team has worked to LyX extensible through various configuration files and external packages. That means that if you want to support the Fizzwizzle LATEX package, you can create a layout file (or module) for it without having to alter LyX itself. We’ve already had contributions of several new features this way. This is the place where all of those get documented.

This manual also documents some special features, like fax support, version control, and SGML support, which require additional software to work properly. Lastly, there’s a chapter of LATEX tools and tips, things you can use to spruce up your documents by directly using the powerful features of LATEX. After all, LyX is only WYSIWYM and will only ever interface to some, not all, LATEX features.

If you haven’t read the Introduction yet, you are definitely in the wrong manual. The Introduction is the first place to go, since it describes the notation and format of all of the manuals. You should also be thoroughly familiar with the User’s Guide and all of the basic features of LyX before attempting to read this one.

Since all the topics in this manual depend heavily on LyX’s interaction with LATEX, this first chapter covers the inner workings of LyX and how to direct LyX to generate exactly the LATEX code you want. It is obviously for more seasoned LyX users.
2 LyX and \LaTeX

2.1 How LyX Uses \LaTeX

This chapter is for both \TeX-nicians and the \LaTeX-curious. In it, we’ll explain how LyX and \LaTeX work together to produce printable output. This is the only place in any of the manuals where we assume you know something about \LaTeX.

At one time, LyX was called a “WYSIWYM frontend to \LaTeX,” but that’s no longer true. There are frontends to \LaTeX out there.\(^1\) These are basically text editors with the ability to run \LaTeX and mark any errors in the file you’re editing. Although LyX is an editor, and it does run \LaTeX, and it also indicates errors in the file, it also does much, much more. For one thing, you don’t need to know \LaTeX to use LyX effectively. And LyX has added its own extensions to \LaTeX. Try the following sometime: select Export\LaTeX from the File menu (or View\Source), then look at the preamble of the resulting .tex file. You’ll notice a variety of new macros defined specifically by LyX. These macros are defined automatically, according to the features you use in the document.

There are several commands that automatically invoke \LaTeX. They are:

- View \hfill Format
- View \hfill Update \hfill Format
- File \hfill Print
- File \hfill Fax

They will only invoke \LaTeX if the file has changed since the last time \LaTeX was run. When LyX runs \LaTeX on the file you’re editing, it performs these steps:

1. Convert the document to \LaTeX and save to a file with the extension .tex in place of .lyx.

2. Run \LaTeX on the .tex file (maybe several times), and run any other commands (such as bibtex or makeindex) needed to compile the \LaTeX file.

3. If there are any errors, show the error log.

If you’ve run \LaTeX using View\DVI, LyX then runs a DVI viewer to display the DVI-file. If you’ve used View\PostScript, LyX performs further steps:

\(^1\)Some familiar ones are \TeXmaker and kile, on Linux, and \TeXshop, OSX. There are also the \LaTeX modes for vi and emacs, of course.
Run `dvips` to convert the DVI file to PostScript.

Run a PostScript viewer, such as `ghostview`, to display the PostScript file.

LyX does similar things when viewing, or exporting, other formats.

## 2.2 Translating \LaTeX files into LyX

You can import a \LaTeX file into LyX by using the File > Import > \LaTeX command in LyX. This will call a program named `tex2lyx` which will create a file `foo.lyx` from the file `foo.tex`. LyX will then open that file.\(^2\)

`tex2lyx` will translate most legal \LaTeX, but not everything. It will put things it doesn’t understand into \TeX code, so after translating a file with `tex2lyx`, you can look for \TeX code and hand-edit it until it looks right.

If you don’t know what \TeX code is, read the next section.

## 2.3 Inserting \TeX Code into LyX Documents

Anything you can do in \TeX you can do in LyX, for a very simple reason: You can always insert \TeX code into any LyX document. LyX cannot, and will never be able to, display every possible \TeX construct. If ever you need to insert \TeX commands into your LyX document, you can use the \TeX Code box, which you can insert into your document with Insert > \TeX Code.

Here’s an example of inserting \TeX commands in a LyX document. The code looks like this:

\begin{verbatim}
\begin{tabular}{ll}
\begin{minipage}{5cm}
This is an example for a minipage environment. You can put nearly everything in it, even (non-floating) figures and tables.
\end{minipage} &
\begin{minipage}{5cm}
This ...
\end{minipage}
\end{tabular}
\end{verbatim}

\(^2\)`tex2lyx` can also be run from the command line, of course.
The \TeX Code box containing this text is directly after this paragraph. Those of you reading the manual in LyX will only see the \TeX code inset. Those reading a printed version of the manuals will see the actual results:

This is an example for a minipage environment. You can put nearly everything in it, even (non-floating) figures and tables.

In addition to using \TeX code, you can also create a separate file containing some complex \LaTeX structure and then use Insert>Child Document to include your file (you should select the type Input). We recommend that you only do this if you have a .tex file which you know works already. Otherwise, you’ll have a big job tracking down \LaTeX errors.

There are a few last points to emphasize:

- LyX does not check if your \LaTeX code is correct.
- Beware reinventing the wheel.

On that last point, LyX does have quite a few features tucked into it, and more are coming. Be sure to check the manuals to make sure that LyX doesn’t have such-and-such feature before you decide you have to do it by hand. Moreover, there are numerous \LaTeX packages out there to do all sorts of things, from labels to envelopes to fancy multipage tables. Check out CTAN for details, and see chapter 7.

If you do need to do some wild and fancy things within your document, be sure to check out a good \LaTeX book for assistance. There are a number of them listed in the bibliography of the User’s Guide.

### 2.4 LyX and the \LaTeX Preamble

#### 2.4.1 About the \LaTeX Preamble

If you already know \LaTeX, there is no need to explain here what the preamble is good for. If you don’t, the following will give you some ideas—we recommend again that you consult a \LaTeX book for further information. In any case, you should read the points below, because they explain what you can do and what you don’t need to do in the \LaTeX preamble of a LyX document.

The \LaTeX preamble comes at the very beginning of a document, before the text. It serves to:

- Declare the document class.
  
  LyX already does this for you. If you’re a seasoned \LaTeX-nician, and you have a custom document class you want to use, check out the Customization Manual for information on how to make LyX interface to it.
• Declare the usage of packages.

\LaTeX{} packages provide special commands, which are only available within a document when the package has been declared in the preamble. For example, the package \texttt{indentfirst} forces all paragraphs to be indented. There are other packages for labels, envelopes, margins, etc.

• Set counters, variables, lengths and widths.

There are several \LaTeX{} counters and variables which must be set globally from within the preamble in order to have the desired effect. (There are variables which you can set and reset inside the document, too.) Margins are a good example of something which must be set in the preamble. Another example is the label format for lists. You can actually set these just about anywhere, but it’s best to do it just once, inside the preamble.

• Declare user defined commands (with \texttt{\newcommand} or \texttt{\renewcommand}).

These are abbreviations for \LaTeX{} commands which appear very often inside a document. Although the preamble is a good place to declare such commands, they can be declared anywhere (before they are used for the first time, of course). This can be useful if there is a lot of raw \LaTeX{} code in your document, which normally should not be the case.

\LYX adds its own set of definitions to the preamble of the \texttt{.tex} file it produces. This makes \LaTeX{} files generated by \LYX{} portable.

### 2.4.2 Changing the Preamble

The commands which \LYX adds to the preamble of a \LaTeX{} file are fixed; you can’t change them without patching \LYX{} itself. You can, however, add your own stuff to the preamble by selecting \LaTeX{} Preamble in the Document Settings dialog. \LYX adds anything in the Preamble dialog to its own built-in preamble. Before adding your own declarations in the preamble, you should make sure that \LYX doesn’t already support what you want to do. (Remember what we said about reinventing the wheel?) Also, make sure your preamble code is correct. \LYX doesn’t check it for you. If there is an error, you’re likely to get an error like “\texttt{Missing \begin\{document\}}”. If you see this error, check your preamble.

### 2.4.3 Examples

Here are some examples of what you can add to a preamble, and what they do.

#### 2.4.3.1 Example #1: Offsets

There are two variables under \LaTeX{} that control page position: \texttt{\hoffset} and \texttt{\voffset}. Their names should be self-explanatory. These variables are useful if you think for a moment about computer labels. Sometimes, the size of a print medium
and the area of the medium that you can actually print on aren’t the same. This is where \hoffset and \voffset come in.

The default values for \hoffset and \voffset are both 0 points, i.e. the page isn’t shifted. Unfortunately, some DVI drivers always seem to shift the page. We have no idea why, or why the sysadmin hasn’t fixed such behavior. If you’re using LyX on a system that you don’t personally maintain, and your sysadmin is a doofus, \hoffset and \voffset can save the day. Suppose you’re left and top margins are always 0.5 inches too big. You can add this to the preamble:

\setlength{\hoffset}{-0.5 in}
\setlength{\voffset}{-0.5 in}

and your margins should now be correct.

2.4.3.2 Example #2: Labels

Speaking of labels, suppose you wanted to print out a bunch of address labels. There’s a rather nice package, available at your nearest CTAN archive, for printing sheets of labels: labels.sty. Now, your system may not have this package installed by default. We leave that up to you to check. You’ll also want to read the documentation for it; we’re not going to do that for you. Since this is an example, however, we’ll give you an example of how you use this package.

First, make sure you’re using the article document class. Next, you need to put the following in your preamble:

\usepackage{labels}
\LabelCols=3
\LabelRows=7
\LeftBorder=8mm
\RightBorder=8mm
\TopBorder=9mm
\BottomBorder=2mm

This sets things up for Avery label sheets, stock #5360. You’re now ready to print labels, but you’ll need to insert \TeX code, placing the commands \begin{labels} and \end{labels} around each label text. This and other special features of labels.sty are explained in its documentation.

Someday, someone may write a LyX layout file to support this package directly. Maybe that someone is you.

2.4.3.3 Example #3: Paragraph Indentation

Americans are trained to indent the first line of every paragraph. As with all of their other weird quirks, most Americans will whine and moan until they can have their
way and indent the first line of all paragraphs. (Yes, we’re joking. (We are?) Yeah, we are.)

Of course, this behavior isn’t standard typography. In books, you typically only indent the first line of a paragraph if it follows another one. The idea behind indenting the first line of a paragraph is to distinguish neighboring paragraphs from one another. If there is no previous paragraph—for example, if it follows a figure or is the first paragraph in a section—then there is no need for indentation.

If you’re a typical American (we’re still joking!), though, you don’t care about such esoteric things; you want your indentation! Add this to the preamble:

\usepackage{indentfirst}

If your \TeX{} distribution isn’t braindead, you’ll have this package, and all of your paragraphs will get the indentation the Founding Fathers intended they should have.

### 2.4.3.4 Example #4: This Document

You can also check out the preamble of this document to get an idea of some of the advanced things you can do. Also, there are more examples and an assortment of \TeX{} “dirty tricks” given in Chapter 7.

### 2.5 \LaTeX{} and \TeX{} Errors

When \LYX{} calls \TeX{}, it tells \TeX{} to blithely ignore any errors and keep going. It then uses the logfile from the \TeX{} run to do a post-mortem. After analyzing the logfile, \LYX{} displays a dialog listing the errors. Clicking on any one of them will take you to the position in your \LYX\ file where the error occurred.\footnote{Well, usually. Analyzing the logfile is a tough job, and \LYX{} doesn’t always go to the right line. There are also cases where \TeX{} reports the error on one line, but the actual error is earlier. This is not unlike forgetting a closing brace in a program: You’ll get an error, but only later.}

Some folks also like to look at the log file directly: It is available from DocumentLatex Log. There are some fairly common error messages and warnings. We’ll cover those here. You should look at a good \TeX{} book for a complete listing.

- \TeX{} Warning
  Anything beginning with these words is a warning message for the purpose of “debugging” the \TeX{} code itself. You’ll get messages like this if you added or changed cross-references or bibliography entries, in which case, \TeX{} is trying to tell you that you need to make another run. You can by-and-large ignore these.

- \TeX{} Font Warning
  Another warning message, this time about fonts which \TeX{} couldn’t find. The rest of the message will often say something about a replacement font that \TeX{} used. You can safely ignore these, too.
2.5 \textit{LyX} and \textit{\LaTeX} Errors

- **Overfull \texttt{hbox}\)**

\LaTeX{} absolutely \textit{loves} to spew these out. They are warnings about lines that were too long and run past the right margin. Almost always, this is unnoticeable in the final output. (It can be just a point or two.) Or, only one or two characters extend past the margin. \LaTeX{} seems to generate at least one of these messages for just about any document you write.

You can ignore these messages. Your eyes will tell you if there’s a problem with something that’s too wide; just look at the output.\footnote{You can also enable the ‘draft’ option in Document\texttt{\ Settings}, and then \LaTeX{} will draw a black box in the margin of lines that are overfull.}

- **Underfull \texttt{hbox}\)**

Not quite as common as its cousin. \LaTeX{} seems to like to print lines that are a bit too wide as opposed to ones that are a bit too narrow. We have no idea why.

- **Overfull \texttt{vbox} and Underfull \texttt{vbox}\)**

Warnings about troubles breaking the page. Once again, just look at the output. Your eyes will tell you where something has gone wrong.

- **\LaTeX{} Error: File \texttt{Xxxx} not found\)**

The file “\texttt{Xxxx}” isn’t installed on this system. This usually appears because some package your document needs isn’t installed. If you didn’t touch the preamble or didn’t use the \texttt{\usepackage{}} command, then one of the packages \texttt{LyX} tried to load is missing. Use \texttt{Help} $\rightarrow$ \LaTeX{} Configuration to get a list of packages that \texttt{LyX} knows about. This file is updated whenever you reconfigure \texttt{LyX} (using \texttt{Tools} $\rightarrow$ \texttt{Reconfigure}) and tells you which packages have been detected and what they do.

If you did use the \texttt{\usepackage{}} command and the package in question isn’t installed, then you’ll need to install it yourself.

- **\LaTeX{} Error: Unknown option\)**

Error messages beginning with this are trying to tell you that you specified a bad or undefined option to a package. Check the package’s documentation.

- **Undefined control sequence\)**

If you’ve inserted \LaTeX{} code into your document, but made a typo, you’ll get one of these. You may have forgotten to load a package. In any case, this error message usually means that you used an undefined command.

There are other error and warning messages. Some are self-explanatory. These are usually \LaTeX{} messages. Others are downright cryptic. These are usually \TeX{} error messages, and we really have no clue what they mean or how to decipher them. No-one does.

There’s a general sequence you should follow if you get error messages:
1. Look at the \LaTeX code you inserted for typos.

2. If there are no typos, check that you used the command(s) correctly.

3. If you get a bunch of error boxes piled up at the very top of the document—and especially if you see a “Missing \begin{document}” error—it means that there are errors in the preamble. Start debugging your preamble.

4. If you didn’t add anything to the preamble and didn’t add any \LaTeX code to the document, the first suspect is your \LaTeX distribution itself. Check for missing packages and install them.

5. Okay, so there are no missing packages. Did you use any of the fine-tuning options in \LaTeX? Specifically, did you misuse any of them, like trying to manually insert lots of Protected Blanks, Linebreaks, or Pagebreaks? Did you try to kludge something together with these instead of using the appropriate paragraph environment?

6. All right, you didn’t use any of the fine-tuning options, you played by the rules. Did you try to pull a fancy maneuver? Did you do something funky inside a table or an equation, like inserting a graphic into a table cell?

7. Do you have long sections of text where \LaTeX cannot find a place to break a line? By default, \LaTeX is rather strict about how much extra inter-word spacing it will add in order to break a line. Preferably, you should rework the paragraph to avoid the problem. If this isn’t an option, you can wrap your text in executable code blocks to make \LaTeX’s line breaking more, well, sloppy.

8. Did you go overboard with the nesting? \LaTeX (currently) doesn’t check to make sure you’re in the limits for nesting environments. If you nested a bunch of environments to the 17th level, that’s the problem. (The limit in \LaTeX is five.)

9. Okay, you didn’t get any error messages, but your output looks awful. If you have a table or figure that’s too wide or long for the page, you need to:
   a) rescale the figure so it fits.
   b) trim down the table so it fits.

10. If something else is wrong with the output, and you didn’t try to pull anything fancy or kludge the fine-tuning options, we’re not sure what’s wrong.

If all this doesn’t help—well, then perhaps you might have found a bug in \LaTeX....
3 Supplemental Tools

3.1 Customizing Bibliographies with Bib\TeX

The basics how to use Bib\TeX are explained in section Bibliography databases (Bib\TeX) of the User’s Guide. The following subsections explain special bibliography features supported by LyX.

3.1.1 Alternative Citation Styles

Standard Bib\TeX uses numbers (e.g. “[12]”) to refer to a cited work. However, in many scientific disciplines, other citation styles are in use. The most common one is the author-year style (e.g. “Knuth 1984a”). LyX supports two packages that provide this style, natbib and jurabib. Both packages have their pros and cons, which cannot be listed in detail. If you only want to have simple author-year (or author-numerical) style, or if you want to use one of the countless style files for natbib, than the established natbib package is probably your choice. If you need special features like short title references, ibidem etc., you might consider the jurabib package.

The handling of both packages in LyX is basically the same. Go to Document\>Settings and select under Bibliography the option Natbib or Jurabib. With both packages, you will get some extra features in the citation dialog and you can select the style of the reference (“Knuth 1984”, “Knuth (1984)”, “Knuth, 1984”, “1984” etc.). Note that both packages need specifically designed style files. They both ship their own, but there are lots of additional style files, and there is even an interactive style file builder\footnote{See ftp://ctan.tug.org/tex-archive/macros/latex/contrib/custom-bib/} for natbib.

3.1.2 Sectioned Bibliographies

Sometimes you might need to divide your bibliography into several sections. If you are for instance a historian, the possibility to separate sources and scientific works is most likely a “must have”. Unfortunately, Bib\TeX itself does not allow you to do this. But with the help of some \LaTeX packages, Bib\TeX can be extended to fit your needs.

LyX provides native support for one of these packages, bibtopic.\footnote{Available from ftp://ctan.tug.org/tex-archive/macros/latex/contrib/bibtopic/} The advantage of this package (compared to other packages like multibib) is that you don’t need to define new citation commands. Instead, you need to prepare different bibliographic...
databases which include the entries for the different sections of the bibliography. For example: If you want to divide your bibliography into the sections “Sources” and “Scientific works”, you first need to create two bibliographic databases, e.g. sources.bib and scientific.bib.

Go to Document> Settings and check under Bibliography the option Sectioned bibliography. Now you can insert multiple Bib\TeX\ bibliography databases, one for each section of your bibliography. Returning to our example: Insert the Bib\TeX\ bibliography sources.bib and a second one for the database scientific.bib. You are free to use the same or different styles for each section. Additionally, you can choose if the bibliography section should contain “all cited references” of the specified database(s) (which is the default), “all uncited references” or even “all references”. This might be useful if you would like to separate your bibliography into three sections: “Cited sources”, “Uncited sources”, and “Scientific works”. The titles for the sections can be added as ordinary sections or subsections. Since \bibtopic\ removes the bibliography title, you have manually re-add that, too (as a chapter* or section*, for instance).

3.1.3 Multiple Bibliographies

Multiple bibliographies, e.g. a bibliography for each section or chapter of the document, are not supported by Bib\TeX\ itself. But the \bibtopic\ package, which is used for the creation of sectioned bibliographies in \LaTeX\ (see the previous section), provides an easy way to solve this task, if you are willing to use some \TeX\ Code (see section 2.3).\footnote{An alternative approach is to use the \chapterbib\ or \bibunits\ package, respectively.}

First go to Document> Settings and under Bibliography check Sectioned bibliography. In the document, you have to enclose the sections, which shall contain their own bibliography (including the Bib\TeX\ bibliography itself), between $\begin{btUnit}$ and $\end{btUnit}$ (those commands have to be inserted as \TeX\ code). The bibliography will contain all references which have been cited in the current btUnit. \textbf{Note:} If you are using this approach, then every citation reference has to be inside some \btUnit. Also, the \btUnits\ cannot be nested.

3.2 Multipart Documents

3.2.1 General Operation

When you are working on a large file with many sections, it is often convenient to break up the document into several files, or perhaps you have something where a table may change from time to time, but the preceding text does not. In these cases, you should seriously consider using multipart documents. For example, scientific papers often have five major sections: the introduction, observations, results, discussion, and conclusion. Each of these could be its own separate \LaTeX\ file, with one “master” file which contains the title, authors, abstract, references, etc., plus the five included files.
3.2 Multipart Documents

It is important to note that each of these files is a full \LaTeX file which can be formatted and printed on its own, as well as included in a master file. Each of these files must have the same document class, however—don’t attempt to mix book classes with article classes. You may also include \LaTeX files; however, these files must not have their own preamble (i.e., everything up to and including the \texttt{\begin{document}} line as well as the \texttt{\end{document}} line must be deleted) or else errors will be generated when you try to make a DVI file.

\LaTeX allows you to include files quite easily with |Insert\>| Child Document. When you click on this selection a small box is inserted into the file at the current cursor location. Clicking on the box raises a dialog which allows you to select the file to be included, and the method of its inclusion.

The file selection box should by now be obvious. The three inclusion methods are “include”, “input”, and “verbatim”. The difference between “include” and “input” is really only meaningful to \LaTeX experts, but the practical difference is that files which are “included” are typeset beginning on a new page, while files which are “inputted” are typeset starting on the current page.

Generally, the master file is converted into a full \LaTeX file before typesetting, while the included files are converted to \LaTeX files which do not have all the preamble information.

A “verbatim” included file allows you to include a file typeset exactly as it appears in the file, i.e. in \texttt{verbatim} mode, with the characters set in a fixed-width typewriter font. Normally, spaces in this file are invisible, though two consecutive spaces are conserved, unlike \LaTeX’s normal treatment of spaces. However, setting the Mark spaces in output checkbox typesets a mark to unambiguously define the presence of a space.

### 3.2.2 Cross-References Between Files

This section is somewhat out of date. Need to describe default master documents and how children are opened when the master is. [[FIXME]]

It is possible to set up cross-references between the different files. First, open all the files in question: let’s call them A and B in a two file example, where B is included in A. Let’s say you insert a label in A, then want to reference it in B. Open the cross-reference dialog in whilst in document B, and you can select the “buffer” to use.

### 3.2.3 Bibliography Lists in all Subdocuments

This section also needs updating. There is now material about this on the wiki, and it could be copied here.

Copy the bibliography list with all entries to all subdocuments and transform them to a comment. This way \LaTeX will find the .bib-files and you can easily insert references without making the bibliography list visible.
3 Supplemental Tools

As the bibliography list is in a comment, \LaTeX won’t use it and the references will look like this: [?], instead of like this: [1]. One solution is to use the \LaTeX-package \texttt{comment} that will only include comments by processing the files separately. To do this, add in the \LaTeX preamble of every subdocument the following:

\begin{verbatim}
\usepackage{comment}
\includecomment{comment}
\end{verbatim}

See also http://wiki.lyx.org/FAQ/Unsorted#toc31.

3.3 Fancy Headers and Footers

The default page layout is rather plain; for an \texttt{article} document class, all you get is a centered page number at the bottom of the page. This document uses KOMA-script’s book class, so it appears to be a bit fancier. But to really put on a show, you need to set the document page style to “fancy”, as mentioned in the \textit{User Guide}. This section describes the \LaTeX code you need to insert in your \LaTeX preamble in order to get the desired effects.

The page header is divided into three fields, not surprisingly labeled “left”, “center”, and “right”. The footer is also divided into these three fields. The \LaTeX commands to set these fields in the simplest manner are \texttt{\lhead}, \texttt{\chead}, \texttt{\rhead}, \texttt{\lfoot}, etc. Suppose you wish to put your name in the upper left hand corner of each page. Simply insert the following command in the preamble:

\begin{verbatim}
\lhead{John Q. DocWriter}
\end{verbatim}

You will now see your name in the upper left. If a field has a default entry that you would like to get rid of (often the page number appears in the central footer, simply include a command with a blank argument, e.g.:

\begin{verbatim}
\lfoot{}
\end{verbatim}

Let’s get really fancy: let’s put the section number with the word “Section” (e.g. Section 3) in the upper left, the page number (e.g. Page 4) in the upper right, your name in the lower left, and the date in the lower right. The following commands should now appear in the preamble:

\begin{verbatim}
\lhead{Section \thesection}
\chead{}
\rhead{Page \thepage}
\lfoot{John Q. DocWriter}
\cfoot{}
\rfoot{\today}
\end{verbatim}

The commands \texttt{\thesection} and \texttt{\thepage} access \LaTeX’s section and page counters, and so print out the current section and page numbers. \texttt{\today} simply prints out today’s date.

The thicknesses of the horizontal rules drawn beneath the header and above the footer can also be modified. If you don’t want one of the rules, set its thickness to 0.

198
3.4 Itemize Bullet Selection

The header rule has a default thickness of 0.4pt, the footer rule is 0pt. Use commands like \renewcommand{\headrulewidth}{0.4pt} and \renewcommand{\footrulewidth}{0.4pt} to set the thicknesses.

You can switch the header/footer settings on and off for individual pages using commands like \thispagestyle{empty}, \thispagestyle{plain}, and \thispagestyle{fancy}. Simply insert them in the text on the page you want changed and mark them as \TeX code. In fact, title pages are marked as plain by default, while following pages are marked fancy when using the global fancy setting.

There are more complex commands which will let you insert things in the upper left on odd numbered pages, etc., but we will refer you to the fancyhdr package documentation for more information. (Find the file fancyhdr.dvi.)

As a final example, it is possible to include an image in the header or footer. Suppose you want to put a company logo in the upper lefthand corner. You might try something like

\lhead{\resizebox{1in}{!}{\includegraphics{logo.eps}}}

(you may need to preface this with \usepackage{graphics} if you don’t include graphics elsewhere in your document).

3.4 Itemize Bullet Selection

by Allan Rae

3.4.1 Introduction

\LaTeX provides 216 bullet shapes that can be accessed from a simple dialog. Using this dialog you can easily specify what bullet shape to use at each level of an itemized list. These settings are document-wide so you won’t be able to specify different sets of bullets for different paragraphs.\footnote{Well, actually you can but you’ll have to do it by hand.}

3.4.2 How it looks

Open the dialog by selecting the Document $\rightarrow$ Settings menu item and then select the Bullets tab.

The dialog provides you with a table of bullet shapes. A column of buttons on the left of the table provides access to the six different panels of bullet shapes. The row of buttons across the top is used to select which bullet depth you are changing. A text entry under the table shows the currently selected bullet shape’s \LaTeX equivalent and this can be edited if desired. If you do modify the text you will also need to specify any needed packages in the \LaTeX preamble.

The six panels are divided up by the packages they require. The following table shows the mappings from button name to \LaTeX packages.
LYX doesn’t stop you using bullets from packages you don’t have. If you get errors from LATEX when you try to view or print the file, then it is likely you are missing a package.  

3.4.3 How to use it

Select which bullet depth you want to change then select the bullet shape and size. Any changes will not be visible in LYX, but are visible when viewing the document. You can reset a bullet shape to the default simply by clicking your right mouse button on the appropriate bullet depth button.  

LyX doesn’t restrict your use since you may be editing locally and exporting elsewhere. If you really want to have multiple sets of paragraphs with different sets of bullets in each, then you’re going to have to get your hands dirty with TEX code. The bullet selection dialog can help though because it provides you with the LATEX code for a wide range of bullet shapes. To make your own custom paragraphs you have the following options:

♯ Use the LATEX command \renewcommand{}{} to specify a new bullet shape for a given depth. You’ll also need to save the current bullet shape so you can restore it again afterwards. In this itemized list the following LATEX code was used to change the bullet used for the first depth.

\let\savelabelitemi=\labelitemi
\renewcommand\labelitemi[0]{\small\sharp}

♯ Note that the itemize depth is specified in Roman numerals as part of the \labelitem command.

⋆ Specify each individual entry by starting each item with the bullet shape enclosed in square brackets and set as TEX Code. For example, this item was started with [\star]. You’ll also need to revert the labelitem back to its previous setting for the global bullet shape settings to remain in effect. The way used here was:

\renewcommand\labelitemi[0]{\savelabelitemi}
4 The LyX Server

4.1 Introduction

The ‘LyX server’ allows other programs to talk to LyX, invoke LyX commands, and retrieve information about the LyX internal state. This is only intended for advanced users, but they should find it useful. It is by writing to the LyX server, for example, that bibliography managers, such as JabRef, are able to “push” citations to LyX.

Please note that, at present, the server does not work natively on Windows\(^1\) but it does work with Cygwin versions of LyX.

4.2 Starting the LyX Server

The LyX server works through the use of a pair of named pipes. These are usually located in UserDir and have the names “lyxpipe.in” and “lyxpipe.out”. External programs write into .lyxpipe.in and read back data from .lyxpipe.out. The stem of the pipe names can be defined in the Tools\(\rightarrow\)Preferences dialog, for example "/home/myhome/lyxpipe”. You must configure this manually in order for the server to start.

LyX will add the ‘.in’ and ‘.out’ to create the pipes. If one of the pipes already exists, LyX will assume that another LyX process is already running and will not start the server. If for some other reason, an unused “stale” pipe is left in existence when LyX closes, then LyX will try to delete it. If this fails for some reason, you will need to delete the pipes manually and then restart LyX.

To have several LyX processes with servers at the same time, you have to use different configurations, perhaps by using separate user directories, each with its own preferences file, for each process.

If you are developing a client program, you might find it useful to enable debugging information from the LyX server. Do this by starting LyX as \texttt{lyx -dbg lyxserver}.

You can find a complete example client written in C in the source distribution as development/lyxserver/server_monitor.c.

Another useful tool is command-line based client you will find in src/client/lyxclient.

\(^1\)There is no reason it cannot do so. But none of the developers on Windows have yet implemented this functionality there.
4.3 Normal communication

To issue a \texttt{LYX} call, the client writes a line of ASCII text into the input pipe. This line has the following format:

\texttt{LYXCMD}:\texttt{clientname}:\texttt{function}:\texttt{argument}

\texttt{clientname} is a name that the client can choose arbitrarily. Its only use is that \texttt{LYX} will echo it if it sends an answer—so a client can dispatch results from different requesters.

\texttt{function} is the function you want \texttt{LYX} to perform. It is the same as the commands you’d use in the minibuffer.

\texttt{argument} is an optional argument which is meaningful only to some functions (for instance, the “self-insert” LFUN will insert the argument as text at the cursor position).

The answer from \texttt{LYX} will arrive in the output pipe and be of the form

\texttt{INFO}:\texttt{clientname}:\texttt{function}:\texttt{data}

where \texttt{clientname} and \texttt{function} are just echoed from the command request, while \texttt{data} is more or less useful information filled according to how the command execution worked out. Some commands, such as “font-state”, will return information about the internal state of \texttt{LYX}, while other will return an empty data-response. This means that the command execution went fine.

In case of errors, the response from \texttt{LYX} will have this form

\texttt{ERROR}:\texttt{clientname}:\texttt{function}:\texttt{error message}

where the \texttt{error message} should contain an explanation of why the command failed.

Examples:

\begin{verbatim}
  echo "LYXCMD:test:beginning-of-buffer:" >/.lyxpipe.in
  echo "LYXCMD:test:get-xy:" >/.lyxpipe.in
  read a <-.lyxpipe.out
  echo $a
\end{verbatim}

4.4 Notification

\texttt{LYX} can notify clients of events going on asynchronously. Currently it will only do this if the user binds a key sequence with the function “notify”. The format of the string \texttt{LYX} sends is as follows:

\texttt{NOTIFY}:\texttt{key-sequence}
where *key-sequence* is the printed representation of the key sequence that was actually typed by the user.

This mechanism can be used to extend LyX’s command set and implement macros. Bind some key sequence to “notify”. Then start a client that listens on the output pipe, dispatches the command according to the sequence, and starts a function that may use LyX calls and LyX requests to issue a command or a series of commands to LyX.

### 4.5 The simple LyX Server Protocol

LyX implements a simple protocol that can be used for session management. All messages are of the form

```
LYXSRV:clientname:protocol message
```

where *protocol message* can be “hello” or “bye”. If “hello” is received from a client, LyX will report back to inform the client that it’s listening to its messages, while “bye” sent from LyX will inform clients that LyX is closing.

### 4.6 Reverse DVI/PDF search

Some DVI/PDF viewers\(^2\) provide reverse search facility (also called inverse search). This means that you can tell LyX to put the cursor to a specific line in the document by clicking at the respective position in the DVI/PDF output. To achieve this, the viewer must be able to communicate with LyX. This is done via the LyX server either by using the named pipe (*lyxpipe*), or the UNIX domain socket (*lyxsocket*) that LyX creates in its temporary directory (this is the way the *lyxclient* program communicates with LyX). In some cases, you need a helper script that mediates between the viewer and LyX, in others, the viewer can communicate with LyX directly. This depends on the selected viewer and on your operating system. The same applies to the way viewers need to be configured and the way the reverse search is actually performed. In what follows, we will thus describe how to setup reverse search for specific viewers. Before we turn to this, though, we will explain what needs to be done generally to enable reverse search in the DVI/PDF output.

#### 4.6.1 Enabling reverse search

\LaTeX{} provides several different methods for reverse search. Some are built-in in the *latex/pdflatex* program, some are provided by external packages. Your choice depends on whether your \LaTeX{} distribution already provides a given method (the built-in methods are rather new) and whether your viewer can cope with it. The available methods are described in the following.

\(^2\)The following viewers offer the reverse PDF search feature: Okular on KDE/Linux, Skim on Mac OSX and SumatraPDF on Windows.
Built-in DVI-search via src-specials (DVI only)

This method provides the DVI file with the necessary information for reverse search. It is available in \LaTeX since quite some time (any somewhat recent \LaTeX distribution should include it), and it works reliably. To enable it, change the \LaTeX (plain) -> DVI or \LaTeX (plain) -> DraftDVI converter in Preferences > File Handling > Converters to latex -src-specials $$i. If this doesn’t work, check if your \TeX engine needs different options (the syntax might differ in some distributions).

External Packages (PDFSync and scrltx)

The packages pdfsync and scrltx provide reverse search facility for PDF output (via pdflatex) and DVI output, respectively. In order to enable it, load the packages in the \LaTeX preamble:

- \usepackage{pdfsync} for reverse PDF search,
- \usepackage[active]{srcltx} for reverse DVI search.

If you want to be able to perform both DVI and PDF reverse searches, you can also insert in the preamble the following lines

```
\usepackage{ifpdf}
\ifpdf
 \usepackage{pdfsync}
\else
 \usepackage[active]{srcltx}
\fi
```

This way, you can preview the file as either DVI or PDF (pdflatex) and the right package will be used.

Note that PDFSync might affect the output layout of your document. It is therefore advised to disable PDFsync for final documents.

Built-in reverse search via Sync\TeX (DVI and PDF)

Recent versions of (pdf)\LaTeX have built-in support for both PDF and DVI reverse search. This so-called Sync\TeX facility is basically the result of the integration of the PDFSync package to the pdftex program and its merge with the scr-specials approach. You need at least \TeX Live 2008 or a recent Mik\TeX distribution in order to use it. Also note that only a few PDF viewers (Skim on the Mac, SumatraPDF on Windows) already provide Sync\TeX support.

To enable Sync\TeX for DVI output, change the \LaTeX (plain) -> DVI or \LaTeX (plain) -> DraftDVI converter in Preferences > File Handling > Converters to latex
4.6 Reverse DVI/PDF search

-synctex=1 $$i, and for PDF output, change the LaTeX (pdflatex) -> PDF (pdflatex) or converter to pdflatex -synctex=1 $$i. Check the documentation whether the viewer needs to be configured for the use with SyncTeX.³

4.6.2 Configuring and using specific viewers

Xdvi (all platforms)

If you use xdvi, you don’t need to do anything else for performing a reverse DVI search, as LyX already provides the necessary hooks for automatically using the lyxclient program. Just setup your document as described above (reverse search is triggered by Ctrl-click or Alt-click on Mac OSX, respectively).

However, if for whatever reason you want to use the named pipe instead of the socket for communicating with LyX, simply change the DVI viewer in Preferences ➤ File Handling ➤ File formats to⁴ xdvi -editor 'lyxedtor.sh %f %l', where lyxedtor.sh is a suitable script. For example, a minimal shell script is the following one:

```sh
#!/bin/sh
LYXPIPE="/path/to/lyxpipe"
COMMAND="LYXCMD:revdvi:server-goto-file-row:$1 $2"
echo "$COMMAND" > "$LYXPIPE".in || exit
read < "$LYXPIPE".out || exit
```

where /path/to/lyxpipe is the LyXServer pipe path specified in Preferences ➤ Paths.⁵

MacDviX (Mac OSX)

At the end of /Applications/MacDviX_Folder/calleditor.script, add the following lines:

```sh
/Applications/LyX.app/Contents/MacOS/lyxedtor "$2" $1
exit 1
```

Modify the lines accordingly if you install LyX somewhere else than in the Applications folder.

Reverse search is triggered by Alt-click (OPTION-click).

Skim (Mac OSX)

Enter open -a Skim.app $$i to the viewer setting in Preferences ➤ File Handling ➤ File formats ➤ PDF (pdflatex), and then in Skim ➤ Preferences ➤ Sync select LyX.

Reverse search is triggered by COMMAND-SHIFT-click.

³The -synctex=1 option enables gzip compression. If your viewer does not support it, you should instead use -synctex=-1.

⁴On Mac OSX you have to use DISPLAY=:0.0 xdvi -editor 'lyxedtor.sh %f %l'

⁵In the development/tools folder of a source distribution you can find a lyxedtor script which is able to locate the lyxpipe based on your preferences.
Okular (KDE)

Go to Settings → Configure Okular… → Editor, select “Custom Text Editor” and add the command `lyxclient -g %f %l`.
Reverse search is triggered by SHIFT-click

YAP (Cygwin)

Launch yap, choose its View → Options menu and select the “Inverse DVI Search” tab. Click on the “New…” button and, in the window that opens, enter “LYX Editor” (or any other name you like) in the “Name:” field. Now click on the button labeled “… ” to open a file dialog and navigate to the directory containing the batch file `lyxeditor.bat` (see below). Select `lyxeditor.bat` and then specify the program arguments as `%f %l` if you want to use the shell script above, or as `-g %f %l` if you want to use the `lyxclient` program. Since yap is a native Windows application, the filename it provides should be converted to POSIX style before being passed to `lyxeditor.sh` or `lyxclient`, and this is the purpose of the `lyxeditor.bat` wrapper, which is as follows:

```bash
@echo off
if "%1" == "-g" goto lyxclient
bash -c './lyxeditor.sh $(cygpath -a "%1") %2'
exit
:lyxclient
bash -c './lyxeditor %1 $(cygpath -a "%2") %3'
```

You have to make sure that both `lyxeditor.sh` and `lyxclient.exe` are in the command PATH, otherwise you have to use their full posix path in the above batch file.
In yap, reverse search is triggered by double-click.

SumatraPDF (Cygwin)

In SumatraPDF, you can set the name of the program that communicates with LYX by simply launching SumatraPDF as `SumatraPDF -inverse-search "lyxeditor.bat -g %f %l"` and then quit. The program will remember the setting and using the `-inverse-search` option will not be needed from now on⁶ (in this way you will be using the lyxsocket; omit the `-g` option if you want to use the lyxpipe and be sure that the `lyxeditor.sh` script is in your command PATH). If SumatraPDF is not your default PDF viewer, you should enter `SumatraPDF.sh` in the viewer setting in Preferences → File Handling → File formats → PDF (pdflatex), where `SumatraPDF.sh` is the following script (to be placed in your command PATH, `/usr/local/bin` being the best choice):

---
⁶ It has been reported that SumatraPDF is not able to remember the settings if it is installed in the Program Files system folder. This problem can be avoided by installing it somewhere else, for example in `/usr/local/bin`.
This is needed because SumatraPDF is a native Windows application and does not understand the posix paths used by the Cygwin version of LyX.

Reverse search is triggered by double-click.
4 The \LaTeX{} Server
5 Special Document Classes

5.1 A&A Paper

by Peter Sütterlin

5.1.1 Introduction

This section describes how \LaTeX\ can be used to write articles for submission to the scientific journal *Astronomy and Astrophysics* (www.edpsciences.fr/aa/ http://www.edpsciences.fr/aa/) using Version 5.01 of the document class *aa.cls*. This package can be downloaded from the ftp site


A manual comes together with that package, and this text is not meant to replace the original manual but merely a short guide how to realize the correct form of your paper.

Please note that the publisher of the journal was changed from Springer to EDP Sciences starting January 1, 2001. That change implicated also some slight changes of the style files, namely the removal of the thesaurus command. The \LaTeX\ class *aa* supports the newest version of these style files, V 5.01. If you have an older version installed, please upgrade. For compatibility, the old (version 4) layout has been kept as article (A&A V4). Please refer to the comments in \LaTeXDir/layouts/aapaper.layout.

5.1.2 Getting started

It is recommended you start from the example template distributed with \LaTeX\X. If you are not using a template, note the following settings:

- Select article (A&A) in the Document>Settings dialog (OK, that one was obvious).

- Don’t change the option Page style: Leave it set to default. The whole layout is done by the macros, you shouldn’t change anything.
5 Special Document Classes

5.1.3 The header block

First thing to enter is the header information. It consists of seven entries, of which some are optional. They are

- **Title**: [required]
- **Subtitle**: [optional]
- **Author**: [required]
- **Address**: [required]
- **Offprints**: [optional] if more than one author: whom to contact for offprint requests.
- **Mail**: [optional] mail address for contacts.
- **Date**: [required]. Suggested format is Received: <date>; Accepted <date>

There is no need to issue the \maketitle command, this is done automatically by \LaTeX when the header is finished. Although the order of the single header entries doesn’t matter it is advised to keep the above sequence, just to get the best optics and meets the layout of the real document.

If you want to place footnotes in the header block, e.g. to state your present address, just use the standard footnote via the menu *Insert* > *Footnote*. \LaTeX will automagically use the term \thanks{} in that case.

In addition to these topics, the macros use three additional \LaTeX commands that have no counterpart in \LYX:

- \and to separate different names for more than one author and institute, respectively.
- \inst{<nr>} to mark corresponding author/institute pairs. The institutes are numbered sequentially as they appear in the **Address** field, so you have to put a marker to each author.
- \email{address} to supply an email address for fast contact.

In all cases, the appropriate command has to be entered in \LYX and marked as \LaTeX code. See the examples.

5.1.4 The abstract

The abstract should immediately follow the header block. With version 5 the abstract environment was changed to a command, and there is now a restriction to only one paragraph. In addition, it should contain an entry with the keywords. This is not yet implemented for \LYX, therefore you have to enter the \LaTeX command \keywords{} by hand and mark it as \LaTeX code. Refer to the example paper.
5.1.5 Supported environments

The A&A paper layout supports the following environments for structuring your text:

- Standard
- Section
- Subsection
- Subsubsection
- Itemize
- Enumerate
- Description
- Caption
- Abstract
- Acknowledgment
- Bibliography
- \LaTeX

5.1.6 Commands not supported by LyX

Some commands are not yet supported by the paper (A&A) layout for LyX. Some have already been mentioned. For the sake of completeness, they are listed all together here:

- \and
- \email
- \appendix
- \authorrunning
- \inst{}
- \keywords{}
- \object{}
- \titlerunning{}

If you want to use any of these commands, you have to enter them yourself. Do not forget to mark them as \LaTeX code!
5 Special Document Classes

5.1.7 Figure and Table Floats

LYX provides support for the necessary float environments figure, figure*, table and table*, therefore we won’t tell much about it here. Refer to the User’s Guide. Just remember that tables should be left-aligned. For that, select the table and change the alignment in Edit> Paragraph Settings.

There is only one special thing: the figures with caption besides the figure. To create such a figure, you have to do the following:

1. Create a wide figure float: Insert> Float> Figure, then right click in the figure and select Span columns.
2. Enter your caption text.
3. Press Return to move the cursor above the caption.
4. Insert your figure
5. Position the cursor behind the figure and insert a horizontal fill: Insert> Special Character> Horizontal Fill.
6. Switch to LATEX mode: M-c t.
7. Enter \parbox[b]{55mm}{. **Do not close the brace!**
8. Position the cursor behind the caption text, switch to LATEX mode and insert the closing brace: M-c t }.

Also, refer to the figures in the example paper.

5.1.8 Referee layout

For submission, the paper has to be formatted in a special double-spacing layout. For this purpose, you have to give the option referee to the documentclass. This must be done using the extra class options field in the Document> Settings dialog. Just enter the string referee there.

5.1.9 The example paper

The Examples directory contains an example paper written with LYX. It is the example paper from the original macro package, translated to LYX. Use it for inspiration, and compare the original LATEX code with LYX way of writing.

5.2 AASTEX

by Mike Ressler
5.2 AASTeX

5.2.1 Introduction

AASTeX is a set of macros produced by the American Astronomical Society to facilitate electronic manuscript submission to the three journals they publish: the Astrophysical Journal (including the Letters and Supplement), the Astronomical Journal, and the Publications of the Astronomical Society of the Pacific. LYX has proven to be an excellent tool for generating these documents, especially given its equation, citation, and figure handling capabilities. LYX requires version 5.0 (or higher) of these macros; preferably 5.2, which is the version described here, or higher. Versions prior to 5.0 are intended for use with \LaTeX2.09 and are fundamentally incompatible with LYX. The AASTeX package may be downloaded from the AASTeX Web site

http://www.journals.uchicago.edu/AAS/AASTeX

A complete user guide is contained in that package and you should familiarize yourself with it thoroughly before embarking on writing a paper in LYX. LYX will not reduce the need to figure out all the AASTeX commands, it will only reduce the drudgery of typing everything in. It is your responsibility to ensure that the final exported \LaTeX document conforms completely to the requirements of the journal to which you are submitting your paper.

5.2.2 Starting a New Paper

I strongly suggest that you start with the AASTeX template file. Click on File\textgreater New from Template, enter the new file name, then choose the aastex.lyx template. This will show the most common fields found in a manuscript. Simply overwrite the existing text (including the brackets, \langle\rangle) with the correct information. Many of the AASTeX commands and environments can be implemented directly in LYX, but some cannot: most noticeably \altaffilmark and \altaffiltext, which should stick out like a sore thumb if you actually just opened the template file. For commands such as these, the \LaTeX code must be entered directly and marked as such. Such commands are referred to as \LaTeX code, or Evil Red Text. I tried to minimize the amount of \LaTeX code needed in an AASTeX document, but there is still a bit more required than any of us would like.

5.2.3 Finishing Your Paper

When the paper is finished to your satisfaction and previews/prints correctly, there are a few “postprocessing” actions which need to be done before you submit it to the journals.

1. Export your paper as a \LaTeX file (File\textgreater Export\textgreater \LaTeX).
2. Edit the resulting .tex file with your favorite text editor.
5 Special Document Classes

a) remove the comment lines before the \documentclass command
b) remove the \usepackage...{fontenc} line if it appears (usually just after \documentclass); also remove the \secnumdepth line if it appears.
c) remove everything between (and including) the \makeatletter and \makeatother commands, except for any commands you specifically put into the \LaTeX preamble (which should appear immediately after the “User specified \LaTeX commands” comment in the .tex file).

3. Run the resulting file through \LaTeX to make sure it still processes correctly.

4. Reread the journal requirements to make sure your filenames and formats are correct.

5. Submit it.

5.2.4 Comments On Specific Commands

I will not describe the detailed usage of the individual AASTeX commands: the AASTeX User Guide (aasguide.tex) gives a good description of each. Thus it’s probably easiest for me to go down the list as found in the guide and offer comments where necessary. So let’s begin ...

5.2.4.1 Things that work as expected

Because they work as you might expect, I simply list them and the section they are found in: \documentclass (2.1.1), \begin{document} (2.2), \title (2.3), \author (2.3), \affil (2.3), \abstract (2.4), \keywords (2.5), \section (2.7), \subsection (2.7), \subsubsection (2.7), \paragraph (2.7), \facility (2.10), \begin{displaymath} (2.12), \begin{equation} (2.12), \begin{eqnarray} (2.12), \begin{mathletters} (2.12), \begin{thebibliography} (2.13.1), \bibitem (2.13.2), all the cite commands and their variations (2.13.2), the generic graphicx figure commands (2.14.1), \begin{table} (2.15.4), \begin{tabular} (2.15.4), \caption (2.15.4), \label (2.15.4, amongst other places), \tablerefs (2.15.5), \tablecomments (2.15.5), \url (2.17.4), \end{document} (2.18).

The following style options also work correctly: longabstract (2.4), preprint (3.2.1), preprint2 (3.2.2), eqsecnum (3.3), flushrt (3.4). Simply put them in the Options box in Layout>Document.

5.2.4.2 Things that work, but require more comment

The following items work, but require a little more discussion:

- These items are reserved for use by the journal editors, but you can put them into the \LaTeX preamble if you feel compelled to do so: \received, \revised, \accepted, \ccc, \cpright (all from 2.1.3)
5.2 \LaTeX

- These items may be placed in the \LaTeX\ preamble, and are included as blanks in the template file: `\slugcomment (2.1.4), \shorttitle (2.1.5), \shortauthors (2.1.5)

- `\email (2.3) – can only be used “standalone”, not in the middle of a paragraph. Use \TeX\ code if you need to embed it.

- `\and (2.3) – will have extra {} after it. This should not cause an error.

- `\notetoeditor (2.6) – can only be used “standalone”, not in the middle of a paragraph. Use \TeX\ code if you need to embed it.

- `\placetable (2.8) – can’t insert a cross-reference tag, you must type the tag name by hand.

- `\placefigure (2.8) – same as for `\placetable.

- `\acknowledgements (2.9) – will have extra {} after it. This should not cause an error.

- `\appendix (2.11) – will have extra {} after it. This should not cause an error.

- `\figcaption (2.14.2) – you can insert an optional filename argument by placing the cursor at the beginning of the text and selecting Insert>Short Title. “Short Title” inserts an optional argument of the type needed by `\figcaption. Hopefully it will be renamed someday.

- `\objectname (2.17.1) – same as `\figcaption for the catalog ID optional parameter.

- `\dataset (2.17.1) – same as `\figcaption for the catalog ID optional parameter.

5.2.4.3 Things not implemented, use \TeX\ code

`\altaffilmark (2.3), `\altaffiltext (2.3), `\eqnum (2.12), `\setcounter{equation} (2.12), Journal name abbreviations (2.13.4), `\figurenum (2.14.1), `\epsscale (2.14.1), `\plotone (2.14.1), `\plottwo (2.14.1), `\tablenum (2.15.4), `\tableline (2.15.4, insert it as the first element in the lefthand cell after where you want it. Don’t use any of \LaTeX\’s rules in the table), `\tablenotemark (2.15.5), `\tablenotetext (2.15.5), much of Misc (2.17, except `\objectname, `\dataset, `\url, and `\email; see above), `\singlespace (3.1), `\doublespace (3.1), `\onecolumn (3.2), `\twocolumn (3.2)
5 Special Document Classes

5.2.4.4 Things that cannot be implemented

... at least in any meaningful sort of way, so I suggest ignoring them. They are the references environment (2.13.3), and the deluxetable environment (2.15). If you really, really need to use deluxetable, I suggest editing it in a separate file with a text editor, then using Insert ▶ Child Document to include it in your LyX document. See the aas_sample.lyx file to see an example of this.

5.2.5 FAQs, Tips, Tricks, and Other Ruminations

5.2.5.1 Getting LyX and AASTeX to cooperate

It can be a bit tricky to get LyX to recognize a new layout and document class. When all else fails, do this:

1. Make certain that \LaTeX{} can find AASTeX. Copy sample.tex (and perhaps table.tex) from the AASTeX distribution into a directory completely unrelated to \LaTeX{} or AASTeX and run \LaTeX{} on sample.tex.

2. Make certain that aastex.layout appears in LyX’s layouts folder

3. Rerun Tools ▶ Reconfigure in LyX, then restart LyX.

4. Open a regular new file, not from a template. Does AASTeX appear in the class list in Document ▶ Settings?

If you get a warning from an existing AASTeX document about not being able to find the AASTeX layout or a message about “You should not mix title layouts with normal ones”, things haven’t been installed correctly.

5.2.5.2 \LaTeX{} error processing a table

LyX, by default, attempts to center the table caption/title. This seems to produce a bad interaction in AASTeX so you should click somewhere in the caption/title, then select Edit ▶ Paragraph Settings, then set the Alignment to Block. This took care of it for me.

5.2.5.3 References

A couple of things: 1) I have noticed some funny spacing in the reference entries in the text. When you enter the bibliography item data, make sure their is no space between the last author and the parenthesis setting off the year; e.g. type Ressler (1992), not Ressler (1992). 2) Entering the references at all is not obvious. The easiest thing is to start typing your first reference at the end of the document, then mark it as type References. That will put a small gray box in front of what you just typed. Click on the box to fill in the rest of the information. For new references, go to the end of an existing reference and press return. That will create a new line with its own box, etc.
5.2.5.4 Including EPS files

Even though AASTeX provides its own figure commands (\plotone, for example), I much prefer \LaTeX’s standard figure commands (with the default graphicx). You can insert the \plotone, etc. commands as \TeX code into a Figure Float box if you desire, but I never have much luck getting the layout right. With the standard graphics, \LaTeX will insert a \usepackage{graphicx} command into the \LaTeX preamble and handle the figures in the standard \LaTeX2\epsilon way, interspersing the figures in the text. I believe ApJ accepts figures exactly this way now; AJ might still use the “stack everything at the end” technique.

5.2.5.5 Things I could have done, but didn’t

There are a few “pretty” things I could have implemented, but chose not to. For instance, I saw no point in double-spacing the text in the \LaTeX window, even though it is double-spaced in the paper manuscript. Also, I chose not to make separate layouts for the preprint and preprint2 styles. Since I assume you will spend most of your time in the plain manuscript mode anyway, I decided not to chew up more disk space with this.

5.2.6 Final Caveat

Your mileage may vary. I’ve now had papers published by both ApJ and AJ that have had 98% of the effort done in \LaTeX; the last 2% was the \LaTeX post-processing and a few cleanups. I have had no trouble with the submission process, and I’m sure the journals were never aware that there might be a difference. So, go forth and publish!

5.3 AMS \LaTeX

by David Johnson; updated by Richard Heck

The AMS \LaTeX layouts are set up to conform to suggested styles for mathematical papers to be submitted to American Mathematical Society publications. The layouts are not tailored to a specific journal, but easily can be. You should refer to the AMS documentation for specific instructions for each journal (usually it will entail only changing a single line in the \TeX output). That documentation is available on the Web at http://www.ams.org or by ftp at ftp://ftp.ams.org/pub/tex/amslatex/. These layouts are appropriate, and useful, for any mathematical writing.

There are two basic AMS \LaTeX layouts:

- amsart: The standard AMS article format.
- amsbook: the standard AMS book (really, monograph) format.
The layouts themselves contain only the minimum necessary to use the AMS classes. They do not, in particular, contain any of the ‘theorem’ environments used for setting theorems, lemmas, and the like. These are contained, instead, in the Theorems (AMS) module, which is loaded by default when you select one of the AMS classes. (It can also be used with other classes and can be removed, if you would rather use something else.) Less commonly used environments are in the Theorems (AMS-Extended) module, which must be loaded manually.

By default, theorems and the like are numbered consecutively throughout the document, but this may be modified by loading the module Theorems (Order by Section) or, if you are using book (AMS), the module Theorems (Order by Chapter). These will number the results as \( n.m \), where the first number refers to the section (or chapter) and the second refers to the total number of results so far in that section (or chapter). Many environments are also available unnumbered. These are indicated by an asterisk at the end. If you happen to want only unnumbered results, the the module Theorems (Starred) provides that option.

Note that these modules do not have to be used with the AMS classes. It is perfectly possible to use the Theorems (AMS) module, and the others mentioned, with other classes, such as article, report, book (KOMA-script), and so forth.

### 5.3.1 What these layouts provide

There is a long list of included environments provided by these layouts. In AMSTeX, there is, in fact, an opportunity to define an unlimited variety of ‘theorem’ environments. However, the AMS recommends the environments that are available in LyX.

The following environments—as well as the standard environments, such as section, bibliography, title, author, and date—are provided by article (AMS) and book (AMS):

**Address** This should be the author’s permanent address.

**Current Address** This should be the author’s temporary address at the time of submission, if different from the Address.

**Email** Author’s e-mail address

**URL** Author’s Web address, if desired.

**Keywords** Key words or phrases used to identify specific topics discussed in the paper.

**Subjectclass** These refer to the AMS Subject Classifications, published and described in Mathematical Reviews. These are also available online at the AMS cites listed above.

**Thanks**
Dedicatory

Translator

The following environments are provided by both the Theorems and Theorems (AMS) modules, in the latter case in both starred (unnumbered) and unstared (numbered) versions. These same environments are provided only in the starred versions by the Theorems (Starred) module:

**Theorem 1.** This is typically used for the statements of major results.

**Corollary.** This is used for statements which follow fairly directly from previous statements. Again, these can be major results.

**Lemma 2.** These are smaller results needed to prove other statements.

**Proposition 3.** These are less major results which (hopefully) add to the general theory being discussed.

**Conjecture 4.** These are statements provided without justification, which the author does not know how to prove, but which seem to be true (to the author, at least).

**Definition.** Guess what this is for. The font is different for this environment than for the previous ones.

**Example.** Used for examples illustrating proven results.

**Problem 5.** It’s not really known what this is for. You should figure it out.

**Exercise.** Write a description for this one.

**Remark 6.** This environment is also a type of theorem, usually a lesser sort of observation.

**Claim.** Often used in the course of giving a proof of a larger result.

**Case 1.** Generally, these are used to break up long arguments, using specific instances of some condition.

**Case 2.** The numbering scheme for cases is on its own, not together with other numbered statements.

**Proof.** At the end of this environment, a QED symbol (usually a square, but it can vary with different styles) is placed. If you want to have other environments within this one—for example, Case environments—and have the QED symbol appear only after them, then the other environments need to be nested within the proof environment. See the section Nesting Environments of the User’s Guide for information on nesting.

And these environments are provided by Theorems (AMS-Extended):
5 Special Document Classes

**Criterion.** A required condition.

**Algorithm.** A general procedure to be used.

**Axiom.** This is a property or statement taken as true within the system being discussed.

**Condition.** Sometimes used to state a condition assumed within the present context of discussion.

**Note.** Similar to a Remark.

**Notation.** Used for the explanation of, yes, notation.

**Summary 7.** Do we really need to tell you?

**Acknowledgement.** Acknowledgement.

**Conclusion.** Sometimes used at the end of a long train of argument.

**Fact 8.** Used in a way similar to Proposition, though perhaps lower on the scale.

In addition, the AMS classes automatically provide the AMS \TeX and AMS fonts packages. They need to be available on your system in order to use these environments.

5.4 AGU journals (aguplus)

by Martin Vermeer

5.4.1 Description

These are the layout files for some of the journals of the American Geophysical Society. It is assumed that you have both the AGU’s own class files and AGUplus installed (everything to be found at ftp://ftp.agu.org/journals/latex/journals).

5.4.2 New styles

Redefined are Paragraph, Paragraph*. They are still called this in the \LaTeX GUI, though their \LaTeX equivalents in the AGU classes are Subsubsubsection and Subsubsection*.

Newly defined styles are Left_Header, Right_Header, Received, Revised, Accepted, CCC, PaperId, AuthorAddr, SlugComment. These are mostly manuscript attributes and defined in the AGU class documentation. I suspect this is still badly incomplete.

5.4.3 New floats

Planotable and Plate. We also have a new Table_Caption.
5.4.4 Supported journals

- *Journal of Geophysical Research*: \texttt{jgrga.layout} — Martin Vermeer

Add your own, it’s not so hard! Look at the \texttt{jgrga.layout} example and \texttt{aguplus.inc}.

5.4.5 Bugs and things to remember

In order to use the new layouts, you must remember to do the following for a new document:

1. \emph{Turn off babel}. This can be done in the Layout$\to$Document or Document$\to$Settings menu item. (AGU articles are always in English, right? So \emph{don’t} choose a language.)

2. Enter \texttt{jgrga} into the document’s Extra Options field. (Yes, this is a bug.)

3. Make sure you use the \texttt{agu.bst} bibliography style, by entering \texttt{agu} into the second field of the Bib\TeX{} inset. None of the standard styles will do.

5.5 Broadway

by Garst Reese

5.5.1 Introduction

Broadway is for writing plays. The format is more decorative than Hollywood, and much less standardized. This format should be suitable for workshops.

5.5.2 Special problems

The same as in Hollywood.

5.5.3 Special features

Insert the \texttt{Speaker} names as labels then cross-reference the label to insert the name. The cross-reference dialog will show the current cast of characters.

5.5.4 Paper size and Margins

USLetter, left 1.6in, right 0.75in, top 0.5in, bottom 0.75in
5.5.5 Environments

The following environments are available. You can use broadway.bind to get the bind keys shown at the right.

- **Standard**
  You should not have to use this, but it is here for anything that does not fit otherwise.

- **Narrative** *M-z n*
  Used to describe stage setting and the action. First use of speaker names in all CAPs.

- **ACT** *M-z a*
  Automatically numbered. On screen it will be arabic, but will print as Roman.

- **ACT** *M-z S at*
  Subtitle for ACT. It is just centered text.

- **SCENE** *M-z S-S*
  Not automatically numbered. You supply the number. This is because I couldn’t figure out how.

- **AT_RISE:** *M-z S-R*
  A special case of Narrative to describe the setting and action as the curtain rises.

- **Speaker** *M-z s*
  The speaker’s (actor’s) title, centered in all CAPS.

- **Parenthetical** *M-z p*
  Instructions to the speaker. The parentheses are automatically inserted. The ( will appear on screen, but both will be in the printed play. This environment is only used within Dialogue.

- **Dialogue** *M-z d*
  What the Speaker says.

- **CURTAIN** *M-z S-C*
  The curtain comes down.

- **Title** *M-z S-T*
- **Author** *M-z S-A*
- **Right_Address** *M-z r*

Hello there.
5.6 Dinbrief

The document class dinbrief can be used to type letters according to German conventions. A template file is included in .../lyx/share/templates for you to use as a starting point.

5.7 EGS journals (egs)

by Martin Vermeer

5.7.1 Description

This is the layout file for the European Geophysical Society journals. The needed egs.cls can be downloaded from the web site of the EGS under www.copernicus.org.

5.7.2 New styles

Right_address, Latex_Title, Affil, Journal, mnumber, FirstAuthor, Received, Accepted, Offsets. The current layout file is unfortunately very unmodular and would benefit from using the various std*.inc file inclusions.

5.8 Elsevier Journals

By Rod Pinna

Elsevier Science Publishers B.V. provides a standard LATEX document class (elsart.cls) for submitting articles to their various journals. The style file can be downloaded directly from their web site: http://authors.elsevier.com/. Instructions are supplied along with the class file, which details the requirements of the publishers. LyX includes package that allows for the use of this class, by a layout and a template file. Installation of the class file is the same as for any other LATEX package; instructions are provided in the Elsevier documentation.

To make use of elsart.cls, a file elsart.layout is supplied. As the Elsevier class file is based mainly on the standard article class, most of the normal functionality is provided. The Elsevier class defines a number of mathematical environments, which are similar to the AMS environments. These commands are all described in the Elsevier documentation, and are available in LyX.

The easiest way to use the Elsevier style is to base documents on the included template file. It is best not to use options such as fancy headings or the geometry package, as elements such as these are defined by Elsevier in their style file. Ideally, no extra packages except those mentioned in the Elsevier documentation should be used. Essentially, Elsevier require as “clean” a LATEX file as possible, as their intention is to take the supplied file and replace the class file with one for the particular journal.
to which the paper has been submitted. This also means that not too much time
should be spent on the formatting of the document. When it comes to be published,
this will change anyway. The rest of the usage for this layout is substantially the
same as for the normal article class. For details of what Elsevier do and don’t allow,
refer to their documentation.

5.9 Foils [aka Foil\TeX]

by Allan Rae

5.9.1 Introduction

This section describes how to use \LaTeX{} to make slides for overhead projectors. There
are two document classes that can do this: the default slides class and the Foil\TeX
slides class. This section documents the latter.

I’m going to say this again, nice and clear, so that there’s no misunderstanding:

This section documents the class “slides (Foil\TeX)” only.

If you’re looking for the documentation for “slides (default)”, check out section 5.21.
If your machine doesn’t have the foils class [“slides (Foil\TeX)" installed, you’ll prob-
ably have to use the default slides class, which isn’t quite as good as foils.

The foils class is designed for use with version 2.1 of the foils.cls \LaTeX{} class file
which is now an integral part of \LaTeX{} 2ε.

5.9.2 Getting Started

Obviously, to use this document class, you need to select “slides (Foil\TeX)” from the
Class entry in the Document Layout dialog. There are some settings in the Document Layout dialog that you should know about that are specific to this class:

- Don’t change the options Sides and Columns on the Document Layout dialog.
  They’re ignored by the foils class.
- The default font size is 20 pt with the other options being 17 pt, 25 pt and 30 pt.
- The default font is sans serif but all math equations are still typeset in the usual
  roman font.
- Foil\TeX{} supports A4 and Letter paper sizes as well as a special size for working
  with 35 mm slides. It doesn’t support A5, B5, legal or executive paper sizes.
- Don’t bother changing the Float Placement settings because they are ignored
  anyway. All floats appear where they are defined in the text.
The **Pagestyle** setting behaves a bit differently for this class. **Foil\TeX** provides extensive footer and header capabilities including a user-defined logo. See section 5.9.4.6 for more details. The title page is treated differently to all other pages in the document and is *always* unnumbered and *always* has the logo centered at the bottom of the page (if one is defined). The possible page style choices and what they do are as follows:

- **empty** The final output contains no page numbers, or other headers or footers (except footnotes of course).
- **plain** The final output contains page numbers centered at the bottom of the page. No other headings or footers (other than footnotes).
- **foilheadings** Page numbers in lower right corner. Additional headers and footers are also shown. This is also the default.
- **fancy** Gives you access to the `fancyheadings` package although its use with **Foil\TeX** is discouraged by the writer of the **Foil\TeX** package because of some potential page layout clashes.

### 5.9.2.1 Extra Options

The following options may be used in the extra class options in the Document> Settings dialog.

- **35mmSlide** This sets up the page layout for 7.33in by 11in paper, which is about the same aspect ratio as a 35mm slide, making it a bit easier to work with this medium.
- **headrule** Places a rule across the page below the header on every page except the title page.
- **footrule** Places a rule across the page above the footer on every page except the title page.
- **dvips** This is automatically set each time you create a new `foils` document. This option tells **Foil\TeX** to use the dvips driver to rotate those pages that are set as landscape foils.
- **landscape** Simply changes the page dimensions to those of a landscape page but doesn’t do any rotation. Thus if you use this option you need to use an external program to rotate each page or feed your paper through your printer as landscape. Note that this option effectively reverses the roles of the **Foilhead** and **Rotatefoilhead** environments (don’t worry these are described in the next section).
- **leqno** Equation numbers on the left.
- **fleqn** Flush-left equations.
5.9.3 Supported Environments

Most of the environments commonly supported in other classes are also supported by the foils class. There are several additional environments provided by Foil\TeX\ as well as a couple added by \LaTeX. The following environments are shared with other classes:

- Standard
- Itemize
- Enumerate
- Description
- List
- \LaTeX-Code
- Verse
- Quote
- Quotation
- Title
- Author
- Date
- Abstract
- Bibliography
- Address
- RightAddress
- Caption
- Comment

That is, all the major environments apart from the sectioning environments. Since foils are essentially self-contained sections, with a title and body, Foil\TeX\ provides specific commands for starting new foils and these are:

- Foilhead
- Rotatefoilhead

\LaTeX\ also provides slightly modified versions of these two environments called:

- ShortFoilhead
- ShortRotatefoilhead

and the differences will be explained in the next section.

Since foils are often used in presenting ideas or new theorems and such Foil\TeX\ also provides a comprehensive box of goodies for presenting them:

- Theorem
- Lemma
- Corollary
- Proposition
- Definition
- Proof
- Theorem*
- Lemma*
The starred versions are unnumbered while the unstarred versions are numbered. There are also two list environments added by LyX and these are:

- Corollary*
- Proposition*
- Definition*

5.9.4 Building a Set of Foils

This section will give a simple introduction to using the different environments to build a set of foils. If you want to see an example set of foils, take a look at the Foils.lyx file you find in LyX’s examples folder.

5.9.4.1 Give It a Title Page

Unlike other classes that provide Title, Author, Date and Abstract environments, foils creates the title on a page of its own. If you leave out the Date environment LaTeX will substitute the current date (every time you regenerate the output).

5.9.4.2 Start a New Foil

As I mentioned earlier, there are four ways of starting a new foil. For portrait foils you should use Foilhead or ShortFoilhead. The difference between these two environments is the amount of space between the title of the foil (the foilhead) and the body of the foil.

Landscape foils are generated using the Rotatefoilhead and ShortRotatefoilhead environments. Again the only difference is the spacing between foilhead and body. Both of the short versions have 0.5 inches less separation between the foilhead and the body.
One problem with the support for landscape foils is the requirement that you have to use the `dvips` driver to generate the PostScript output otherwise the foils won’t be rotated. It is possible to get landscape foils even if you haven’t got the `dvips` driver provided you can feed your foils sideways through your printer ;-) 

5.9.4.3 Theorems, Lemmas, Proofs and more

Due to a small bug in LYX you can’t have two of the same type of these environments directly following each other. They must be separated by something. If you try, you will just be extending the previous environment as if you had merged the two environments together. So, how do you get around this problem? The simplest option is to insert some text between the two environments or add a \LaTeX{} environment between the two with just a “\%” in it. This will force LYX to produce two separate environments and hence the correct \LaTeX{} output. An example is provided in the example file included with the LYX distribution. Remember, this problem only occurs if you are trying to place two of the same type of theorem-like environments one directly after the other.

5.9.4.4 Lists

You get all the commonly supported list styles found in other classes as well as two new ones. I’ll only describe the new ones here. If you want to find out more about the other list environments check out the User’s Guide. If you intend to use itemized lists you might also want to read about the Itemize Bullet Selection dialog described above in section 3.4.

The two new list styles, \texttt{TickList} and \texttt{CrossList}, are designed to make it easier for you to create lists of do’s and don’ts or right and wrong by providing dedicated environments that use a tick or a cross as the label of the list. These lists are in fact dedicated variants of the \texttt{Itemize} environment. They do however require that you have the psnfss packages installed.

5.9.4.5 Figures and Tables

\Foil\LaTeX{} redefines the floating tables and figures so that they appear exactly where they are in the text rather than pushing them to the top of the page or to some user specified location. In fact if you change the float placement settings they are simply ignored.

5.9.4.6 Page Headers and Footers

\texttt{My Logo} and \texttt{Restriction} are two commands used to control the left-footer text string. The first is meant to allow you to include a graphic logo on your foils and defaults to “-Typeset by \Foil\LaTeX{}-”. While the second is meant to provide a classification for the audience, \textit{e.g.} Confidential. It is empty by default.
The remaining page corners can be filled by Right Footer (which defaults to page numbers), Right Header (top right) and Left Header (top left).

5.9.5 Unsupported Foil\TeX Goodies

All the commands mentioned below need to be set in a \LaTeX environment or as \TeX within another environment.

5.9.5.1 Lengths

All lengths are adjusted using the \texttt{\setlength{\textit{lengthname}}{\textit{newlength}}} command. Where \textit{lengthname} should be replaced by the name given to the length you want to change and \textit{newlength} is the length value. All lengths should be specified in units of length such as inches (\texttt{in}), millimeters (\texttt{mm}) or points (\texttt{pt}) or relative to some document or font-based length such as \texttt{textwidth}.

It’s possible to change the spacing between a foilhead and the body of the foil by adjusting the length specified by \texttt{\foilheadskip}. For example, to make all foilheads 0.5 in closer to their bodies put the following in the preamble: \texttt{\setlength{\foilheadskip}{-0.5in}}

The spacings around floats can be adjusted by setting these lengths:

\texttt{\abovefloatskip} Separation between the text and the top of the float
\texttt{\abovecaptionskip} Separation between the float and the caption
\texttt{\belowcaptionskip} Separation between the caption and the following text
\texttt{\captionwidth} You can make the captions narrower than the surrounding text by adjusting this length. Best done relative to \texttt{textwidth}.

There are also several title page related lengths that you may find useful if you have a long title or several authors:

\texttt{\abovetitleskip} Separation from headers to Title
\texttt{\titleauthorskip} between Title and Author environments
\texttt{\authorauthorskip} between multiple Author lines
\texttt{\authordateskip} between the Author and the Date
\texttt{\dateabstractskip} between the Date and the Abstract

The last length related command affects all the list environments. If you place \texttt{\zerolistvertdimens} inside a list environment then all the vertical spacing between the list items is removed. Note that this is a command not a length so it doesn’t require \texttt{\setlength} like the stuff mentioned above.
5.9.5.2 Headers and Footers

The `\LogoOn` and `\LogoOff` commands control whether the logo in the MyLogo definition appear on a given page. If you put `\LogoOff` in the preamble then none of the foils will have the logo on them. If you don’t want the logo on a particular page place the `\LogoOff` directly after the foilhead of that page and the `\LogoOn` directly after the next foilhead.

If you decide to use the fancy page style setting in the Document Layout dialog you should probably add `\let\headwidth\textwidth` to your preamble so headers and footers on landscape pages are correctly placed when rotated. This is due to some clashes between the page layouts provided by the fancyheadings package and the foils class.

5.10 Hollywood (Hollywood spec scripts)

by Garst Reese

5.10.1 Introduction

Getting the format of a Hollywood script right is a “rite of passage.” It is designed to make the readers focus on content and to be easy and familiar for the actors to read. Each page of a script should be one minute of film. Nothing goes in a script that you cannot see or hear on screen. The courier 12 pt font should be used throughout. No italics.

5.10.2 Special problems

Speakers’ lines should NEVER break in mid-sentence. If a speaker’s lines continue over a page break, repeat the Speaker title followed by (Cont’d).

5.10.3 Special features

Insert the Speaker names as labels then cross-reference the label to insert the name. The cross-reference dialog will show the current cast of characters. You can use this to insert the speaker name in narratives also.

5.10.4 Paper size and Margins

USLetter, left 1.6in, right 0.75in, top 0.5in, bottom 0.75in

5.10.5 Environments

The following environments are available. You can use hollywood.bind to get the bind keys shown at the right.
• **Standard**  
Used where nothing else works. Try to avoid it.

• **FADE_IN:**  
M-z S-I  
Usually followed by something like “on Sally waking up.”

• **INT:**  
M-z i  
Introduces a new INTERIOR camera set-up. Always followed by DAY or NIGHT, or something similar to define the lighting required. Everthing on this line in CAPS.

• **EXT:**  
M-z e  
Introduces a new EXTERIOR camera set-up. Everthing on this line in CAPS.

• **Speaker**  
M-z s  
The character speaking.

• **Parenthetical**  
M-z p  
Instructions to the speaker. The () are automatically inserted, but only the ( will show in LYX. Both will be printed.

• **Dialogue**  
M-z d  
What the Speaker says.

• **Transition**  
M-z t  
Camera movement instruction. e.g. CUT TO:

• **FADE OUT:**  
M-z S-I

• **Author**  
M-z S-A

• **Title**  
M-z S-T

• **Right_Address**  
M-z r

### 5.10.6 Script jargon

- **(O.S)** — off screen
- **(V.O)** — voice over
- **b.g.** — background
- **C.U.** — close-up
- **PAN** — camera movement
- **INSERT** — cut to close-up of
5.11 ijmpc and ijmpd

by Panayotis Papasotiriou

5.11.1 Overview

The ijmpc package is a set of macros that facilitates electronic manuscript submission to the International Journal of Modern Physics C. Similarly, the ijmpd package is for creating manuscripts to be submitted to the International Journal of Modern Physics D. Both journals are published by World Scientific. The corresponding document classes are named ws-ijmpc.cls and ws-ijmpd.cls, respectively. These files, together with instructions for the authors, can be downloaded from the sites http://www.worldscinet.com/ijmpc/mkt/guidelines.shtml and http://www.worldscinet.com/ijmpd/mkt/guidelines.shtml. Both packages are modified versions of the standard “article” package, and they are almost (but not exactly) identical. Most of their features are supported by \LaTeX{}. I have used \LaTeX{} successfully to write articles submitted to both journals without any problem.

5.11.2 Writing a paper

As usual, the easiest way to write a paper is to start with a template. Click on File $\Rightarrow$ New from Template, then choose the ijmpc.lyx or ijmpd.lyx template. This will give an (almost) empty document that includes the most common fields found in a manuscript. Simply overwrite the existing text (including the brackets, $<>$) with your text. You should keep in mind the following remarks.

1. \LaTeX{} won’t let you change the font size and the page style of the document, because such modifications are not allowed by both packages.

2. The language of the document should not be changed. Before previewing your paper, be sure that the babel package is not used. To do this, click on Tools $\Rightarrow$ Preferences, select the Lang Opts tab, deselect the Use babel checkbox in the language settings, and click on Apply (or Save, if you wish to make this change permanent).

3. The “Keywords” style must be used to define keywords.

4. The ijmpc package provides a style named “Classification Codes”, which can be used to define classification codes, such as PACS numbers. Note that this facility is not supported by the ijmpd package.

5. Several new environments are available: “Definition”, “Step”, “Example”, “Remark”, “Notation”, “Theorem”, “Proof”, “Corollary”, “Lemma”, “Proposition”, “Prop”, “Question”, “Claim”, and “Conjecture”. Their use is more or less obvious. \LaTeX{} supports all these environments; it will use the proper label, text style, and numbering scheme for each of them.
5.11.3 Preparing a paper for submission

Before you submit your paper you must export the \texttt{\LaTeX} document as a \texttt{\LaTeX} file (File $\Rightarrow$ Export $\Rightarrow$ \LaTeX)$^1$, then make the following changes to the resulting .tex file.

1. Remove the comment lines before the \texttt{$\backslash$documentclass} command.

2. Remove everything between (and including) the \texttt{$\backslash$makeatletter} and \texttt{$\backslash$makeatother} commands, except for any commands you specifically put into the \texttt{\LaTeX} preamble.

$^1$Actually you have the choice between \texttt{\LaTeX} (plain) and pdflatex. If you intend to use pdflatex to prepare the paper, you should use the pdflatex option so that included graphics are converted to PDF format, ready for use by pdflatex.
5 Special Document Classes

The modified .tex file should be saved and processed through L\LaTeX as many times as necessary. You may also want to check the resulting .dvi document.

5.11.4 Use of TeX code

The use of TeX code is reduced to two commands, which must be placed at the top of the document. If you started writing your paper by using the ijmpc.lyx or the ijmpd.lyx template, the TeX code needed is already in its place; you usually don’t need to delete it. You may only modify the first TeX code to specify the information printed to the top of odd and even pages (authors’ names and short paper’s title, respectively). This TeX code must have the form \markboth{Authors’ Names}{Short Paper’s Title}.

5.12 iopart

by UWE STÖHR

5.12.1 Overview

The iopart package provides a document class to create electronic manuscript submission to the journals published by the Institute of Physics. Instructions for the authors how to create a paper using the iopart class can be downloaded together with the iopart package from the site ftp://ftp.iop.org/pub/journals/latex2e.

5.12.2 Writing a paper

The easiest way to write a paper is to start with the file IOP-article.lyx that is available in LyX’s examples files folder. Open this file, save it under a new name, and start writing. The example file explains how to use the special text environments. Here are the most important advices:

- To be able to compile your document to a PDF, PS, or DVI, assure that the two options Use AMS math package in the document settings under Math Options are not used!

- The title environment defines the kind of your paper. So use one of the following environments for the title:
  - Title for a Paper
  - Review for a Review
  - Topical for a Topical review
  - Comment for a Comment
  - Note for a Note
5.13 Kluwer

by PANAYOTIS PAPASOTIRIOU

5.13.1 Overview

The Kluwer package is a set of macros produced by Kluwer Academic Publishers that facilitates electronic manuscript submission to the journals they publish. Most known of them (at least in my domain of interest) are Astrophysics and Space Science and Solar Physics, but there are many others (see a complete list at http://www.wkap.nl/jrnllist.htm/JRNLHOME). The Kluwer package may be downloaded from the site http://www.wkap.nl/kaphtml.htm/STYLEFILES. A complete user guide is contained in that package (but it can also be downloaded separately).

\LaTeX{}X supports many features of the package but not everything. However, the TeX code needed is reduced to some “peculiar” commands of the package (see 5.13.4). I have recently used \LaTeX{}X to write an article submitted to the Astrophysics and Space Science without any problem.

5.13.2 Writing a paper

The easiest way to write a paper is to start with the Kluwer template file. Click on File\textgreater New from Template, then choose the \texttt{kluwer.lyx} template. This will give an (almost) empty document that includes the most common fields found in a manuscript and a short description of their use. As in most templates, simply overwrite the existing text (including the brackets, \texttt{<>}) with the correct information.
5 Special Document Classes

5.13.3 Preparing a paper for submission

As in the AASTeX package, before you submit your paper to a journal you must “postprocess” it as follows.

1. Export your paper as a \LaTeX file. To do this, click on File\rightarrow Export\rightarrow \LaTeX.

2. Edit the resulting .tex file with a text editor and make the following changes
   a) remove the comment lines before the \documentclass command,
   b) remove everything between (and including) the \makeatletter and \makeatother commands, except for any commands you specifically put into the \LaTeX preamble.

Save the resulting .tex file.

3. Run the .tex file through \LaTeX as many times as necessary (usually up to three).

4. View the resulting .dvi document using, e.g. xdvi, and check if everything is OK (it should, if you didn’t make any mistake).

5.13.4 “Peculiarities” of the Kluwer package

The Kluwer package has the following “peculiarities”.

1. It is possible to write multiple articles in the same \LaTeX file\textsuperscript{2}. Each article must be included in the environment “article”. Unfortunately, this environment cannot be omitted, even if you write just one article. Therefore, each article starts with the command \begin{article} and, obviously, ends with the command \end{article}. Although this can be implemented in LyX, I didn’t included it, since it looks ugly and can confuse the novice user. Therefore, you need to enter them directly and mark them as \LaTeX code (the well-known “\TeX code”).

2. Information given at the beginning of the article (i.e., title, subtitle, author, institution, running title, running author, abstract and keywords) must be included in an environment called “opening”. This is not implemented in LyX, so you must enter title, subtitle etc. between two \TeX code lines (\begin{opening} and \end{opening}).

3. According to the user manual, the label of each bibliography item must be written as \protect\citeauthor\citeyear{author1\citeyear{year}}.

The kluwer.lyx template takes care of all these “peculiarities”. If you start a new paper using this template you don’t need to do anything special. Just

\textsuperscript{2}I can’t imagine any good reason to do this.
1. don’t delete the \TeX code included in the template, and

2. copy the example bibliography item included in the template and modify it as necessary to enter new bibliography items.

5.14 Koma-Script

by Bernd Rellermeyer

5.14.1 Overview

The \LaTeX document classes \texttt{article (koma-script)}, \texttt{report (koma-script)}, \texttt{book (koma-script)}, and \texttt{letter (koma-script)} correspond to the \LaTeX document classes \texttt{scartcl.cls}, \texttt{scrreprt.cls}, \texttt{scrbook.cls}, and \texttt{scrletr.cls}, resp. of the Koma-Script family. They are replacements for the standard document classes \texttt{article.cls}, \texttt{report.cls}, \texttt{book.cls} and \texttt{letter.cls}, resp., and fit better to European typography conventions in a number of points.

- Standard character size is 11pt in \texttt{article (koma-script)}, \texttt{report (koma-script)}, and \texttt{book (koma-script)}, and 12pt in \texttt{letter (koma-script)}.

- Headings, labels of the description environment, and a number of elements of the \texttt{letter (koma-script)} document class are set in a bold sans serif font.\footnote{There is a big difference between the bold sans serif old cm fonts and new ec fonts, especially in the appearance of headings. In comparison, the ec bold sans serif fonts look a bit thin. Here the \LaTeX package \texttt{cmsd.sty} by Walter Schmidt helps to produce the “usual” appearance when using the ec fonts.} The numbering of chapter headings is made in the same way as the numbering of section headings, that is without the extra line “Chapter...”. In addition, the appearance of the headings can be modified by using a number of options (in \LaTeX to be entered in the field \texttt{Extra Options} of the dialog \texttt{Layout $\rightarrow$ Document}). A detailed German description of these options can be found in the Koma-Script documentation \texttt{scrguide}.

- The main means in the Koma-Script document classes to design the type area are the options \texttt{BCOR} and \texttt{DIV} (in \LaTeX to be entered in the extra class options field in the dialog \texttt{Document $\rightarrow$ Settings}). They make a clearer modification of page margins possible as do the options of the dialog \texttt{Document $\rightarrow$ Settings}. A detailed German description of these and other type area options can be found in the Koma-Script documentation \texttt{scrguide}.

- The \LaTeX document classes of the Koma-Script family define a number of additional commands. Those part of it which makes sense in \LaTeX is implemented in corresponding paragraph types.
A detailed German description of the \LaTeX\ document classes of the Koma-Script family can be found in the Koma-Script documentation scrguide.\textsuperscript{4} The following sections describe only those aspects, which are relevant in \textsc{lyx}.

### 5.14.2 article (koma-script), report (koma-script), and book (koma-script)

The document classes article (koma-script), report (koma-script), and book (koma-script) are implemented in the layout files \texttt{scrartcl.layout}, \texttt{scrreprt.layout}, and \texttt{scrbook.layout}, resp. They contain all the paragraph types of the corresponding standard document classes article, report, and book, resp., partly modified, with the exception of the \textsc{lyx} specific List-type, which is replaced by the new \texttt{Labeling}-type having the same functionality. Beside the \texttt{Labeling}-Type there is a number of new paragraph types added. They are not part of letter (koma-script).

- \texttt{Addpart}, \texttt{Addchap}, \texttt{Addsec}: are equivalents to \texttt{Part*}, \texttt{Chapter*} and \texttt{Section*}, resp., additionally inserting an entry in the table of contents. \texttt{Addpart} and \texttt{Addchap} are not contained in article (koma-script).

- \texttt{Addchap*}, \texttt{Addsec*}: behave exactly as \texttt{Addchap} and \texttt{Addsec}, resp., additionally clearing running heads. \texttt{Addchap*} is not contained in article (koma-script).\textsuperscript{5}

- \texttt{Minisec}: generates a heading directly above the following paragraph in the standard character size without affecting the structure of the document.

- \texttt{Captionabove} and \texttt{Captionbelow} are special captions which respect the different space settings needed for captions placed above or below an element (if you follow strict typographic rules, you might want to place table captions always above the table). You can also use the class option \texttt{tablecaptionsabove}, which will switch caption to \texttt{captionabove} for tables and \texttt{captionbelow} for figures. You need at least Koma-Script version 2.8q to use this.

- \texttt{Dictum}: can be used to set a bonmot, e.g. at the beginning of a chapter. If you use the optional argument (\texttt{Insert \& Short Title}), you can insert the dictum’s author there. Dictum and author are separated by a line. You need at least Koma-Script version 2.8q to use this. \texttt{Dictum} is not contained in article (koma-script).

The following types, together with the standard types Title, Author, and Date, form the title area of the document. They must be entered ahead of the first “ordinary” paragraph.\textsuperscript{6} When such a type is used more than once, the latter usage overwrites the former one, that means, for every type only the latest usage is valid. The order of the

\textsuperscript{4}There is an English translation \texttt{scrguide}, but it is not a complete one.

\textsuperscript{5}There is also an \texttt{\textbackslash addpart*} command in book (koma-script) and in report (koma-script), but since this is identical to \texttt{Part*}, is has not been implemented in \textsc{lyx}.

\textsuperscript{6}The corresponding \LaTeX\ commands must appear before the \texttt{\textbackslash maketitle} command.
different types however has, like Title, Author, and Date, no effect on the appearance of the produced document.

- **Subject**: produces a centered paragraph above the ordinary title (Title, Author, Date) for the subject of the document.

- **Publishers**: produces a centered paragraph below the ordinary title (Title, Author, Date) for the publishers’ name.

- **Dedication**: in report (koma-script) and book (koma-script) produces a centered paragraph on its own page behind the title page, or in article (koma-script) produces a centered paragraph below the ordinary title (Title, Author, Date, Publishers) for a dedication.

- **Titlehead**: produces a left aligned paragraph above the ordinary title (Title, Author, Date, Subject) for a document’s head.

- **Uppertitleback**: produces in a double-sided print in report (koma-script) and book (koma-script) a left-aligned paragraph at the top of the title page’s back or has no effect in a single-sided print or in article (koma-script).

- **Lowertitleback**: produces in a double-sided print in report (koma-script) and book (koma-script) a left-aligned paragraph at the bottom of the title page’s back or has no effect in a single-sided print or in article (koma-script).

- **Extratitle**: produces a special “dirty” page ahead of the actual document containing a paragraph without special formatting.

The layout files for the document classes article (koma-script), report (koma-script), and book (koma-script) do include the file scrmacros.inc. This is thought of as a place to define your own types. Copy scrmacros.inc in your personal layout directory and edit the file!

### 5.14.3 letter (koma-script)

The document class letter (koma-script) is implemented in the layout file scrlettr.layout. It contains all the paragraph types of the corresponding standard document class letter, partly modified, with the exception of the LyX specific types LyX-Code and Comment and the List type, which is replaced by the new Labeling type. In addition, it contains, in contrast to the standard document class, the standard types \LaTeX, Quotation, Quote, and Verse. Furthermore, there are a number of new letter specific types.

The appearance of the letter produced by this document class can be controlled by a number of \LaTeX commands, which you can put in the \LaTeX preamble.\footnote{For example, the standard appearance of the letter’s heading, consisting of name and address, is quite self-willed. An “ordinary” heading is produced by the following \LaTeX commands in the}
5 Special Document Classes

detailed German description of such \LaTeX commands can be found in the Koma-Script documentation \texttt{scrguide}. With it, the letter’s author can produce his personal letter layout.

The types \texttt{Letter} and \texttt{Opening} define the beginning of the letter and must be used in every letter. To emphasize them in the \LaTeX document class, they are marked with the letter \texttt{L} or \texttt{O}, resp. in the left margin. It is possible to write any number of letters in one file. An \texttt{Opening} type produces a new letter using the same addressee and a \texttt{Letter} type produces a new addressee. The types \texttt{Closing}, \texttt{PS}, \texttt{CC}, and \texttt{Encl} are ordinary paragraph types and can also be used several times in one and the same letter.

- \texttt{Letter}: produces a paragraph for the addressee and implicitly defines the beginning of the letter.
- \texttt{Opening}: produces a paragraph for the form of address and implicitly produces a new letter.
- \texttt{Closing}: produces a paragraph for a close.
- \texttt{PS}: produces a paragraph for a postscript.
- \texttt{CC}: produces a paragraph for a distribution list.
- \texttt{Encl}: produces a paragraph for enclosures.

The types \texttt{Name}, \texttt{Signature}, \texttt{Address}, \texttt{Telephone}, \texttt{Place}, \texttt{Backaddress}, \texttt{Specialmail}, \texttt{Location}, \texttt{Title}, and \texttt{Subject} are input types provided with a label to enter information, which will be processed by the document class.\footnote{It could be seen as a matter of inconsequence, that the types \texttt{Letter} and \texttt{Opening} described above are not such input types as well. Because of the special meaning of those types, however, I have implemented them as ordinary paragraph types with a one letter mark in the left margin. Moreover, it would affect my feeling of symmetry, if the \texttt{Opening} type and the \texttt{Closing} type had such a serious different appearance.} The types must be used ahead of the corresponding \texttt{Opening} type.

An implementation of these types in a WYSIWYG fashion does not seem to make sense, because the real appearance of the produced letter does not only depend on the usage of the particular type, but also on other factors. For example, a signature entered in the \texttt{Signature} type will in the standard behavior appear in the produced letter only, when in the same letter also a \texttt{Closing} type is used. The entered value of the \texttt{Telephone} type will in the standard behavior not appear in the produced letter.

\begin{verbatim}
\firsthead{\parbox[b]{\textwidth}{\ignorespaces \fromname\ \ignorespaces \fromaddress}}
\nexthead{\parbox[b]{\textwidth}{\ignorespaces \fromname \hfill \ignorespaces \pagename\ \thepage}}
\end{verbatim}
at all. The possibility to design the letter’s heading freely is already indicated in a footnote above. The input types can also be used as empty paragraphs. This makes sense e.g. for the Signature type. If the Signature type is not used at all, in the standard behavior the value of the Name type is used as signature, whereas if an empty Signature type is used, no signature value is defined.

By using the input types it is possible to write a letter template, containing filled input types with your personal dates (name, address, etc.) and empty input types for other dates you want to enter.

- **Name**: sender’s name, in the standard behavior appears as a centered paragraph in small caps in the letter’s heading.

- **Signature**: sender’s signature, in the standard behavior appears below the Closing type. If no Signature type is used, the value of the Name type appears instead.

- **Address**: sender’s address, in the standard behavior appears in a centered paragraph in the letter’s heading below the sender’s name.

- **Telephone**: sender’s telephone number, in the standard behavior only sets the \texttt{\LaTeX} variable `\texttt{\textbackslash telephonenum}`.

- **Place**: place of the letter’s making.

- **Date**: date of the letter’s making. Place and Date, in the standard behavior, produce the place and the date in a right-aligned line below the addressee’s field. If an empty Date type is used, neither place nor date appear, independent of the value of the Place type. If no Date type is used, the date of the letter’s production is used.

- **Backaddress**: sender’s back address, in the standard behavior appears above the addressee’s field in a small sans serif font.

- **Specialmail**: special mail information, in the standard behavior appears underlined above the addressee’s field below the back address.

- **Location**: additional information, in the standard behavior appears on right side below the addressee’s field.

- **Title**: the letter’s title, in the standard behavior appears in a big, bold, sans serif font above the subject.

- **Subject**: the letter’s subject, in the standard behavior appears in a bold font above the Opening paragraph.
The types Yourref, Yourmail, Myref, Customer, and Invoice produce a business letter like line above the Title line containing the fields “Your ref.”, “Your letter of”, “Our ref.”, “Customer no.”, “Invoice no.”, and “Date”. For the date field, the value of the Date type is used. If one of these “business letter types” is used, the value of the Place type however does not appear, but only the \fromplace variable is set. The ordinary output of place and date in a right-aligned line below the addressee’s field is suppressed. The types are implemented as input types provided with a label and must be used ahead of the corresponding Opening type.

- Yourref: Your ref.
- Yourmail: Your letter of.
- Myref: Our ref.
- Customer: Customer no.
- Invoice: Invoice no.

5.14.4 The new letter class: letter (koma-script v.2)

by Jürgen Spitzmüller

Koma-Script version 2.8 has introduced a new letter class scrlttr2 which supersedes the now unsupported scrlettr. It has — on the \LaTeX side — a completely new interface and is not compatible with the old class. Therefore, \lyx supports both, though it is recommended to use the new class.

This class covers the same functionality as letter (koma-script), and a few more. The basic items are Address (receiver’s address, same as Letter in the old layout), Opening, and Closing. NextAddress will start a new letter (i.e. you can write several letters per document). New elements are sender’s E-Mail, URL, Fax, Bank and the possibility to use a Logo (via Insert\ Graphics) in the header.

The biggest improvement is, though, that the letter’s layout is configurable at almost any needs. This can be done via the preamble or with a special style file (Letter Class Option, extension \*.lco), that will be read in as a class option.\ Have a look at the koma-letter2 template that is included in \lyx for examples. A detailed description is to be found in the Koma-Script documentation (scrguide).

5.14.5 Problems

Visualizing the Koma-Script document classes in \lyx, the \lyx internals cause some problems.

\footnote{The KOMA package comes with some default \*.lco files. There is, for instance, a DIN.lco file that follows german typesetting rules, or a KOMAold.lco that provides the default layout of the old scrlettr class. The latter can be loaded with the class option KOMAold, inserted via the Layout\ Document\ Extra Options field.}
• The chapter number of a Chapter type appears on a line of its own above the chapter heading instead of appearing in the same line ahead of it. The cause for that is the LyX internal behavior for the labeltype Counter_Chapter in the layout file.

• The headings of the types Addchap and Addsec are only put in the “true” \LaTeX\ table of contents, but not in the LyX table of contents (Document> Table of Contents).

• The paragraphs in a letter document class appear in a skip separation mode, not indented. This is the standard behavior, no special \LaTeX\ commands are needed for that. But in the Document> Settings dialog the corresponding radio button indicates Indent. A Skip value always has the effect that extra \LaTeX\ commands are inserted in the document to produce the gap, which is not what is wanted in this case.

5.15 Latex8 (IEEE Conference Papers)

by Allan Rae

5.15.1 Introduction

Since this class is specifically for writing submissions to IEEE sponsored conferences I strongly recommend that you get a copy of their Authors Kit. The latex.sty package and associated bibliography style file is included in the kit. The Authors Kit is usually sent out by email once your initial submission has been accepted. There is a lot of useful information in the Authors Kit explaining formatting restrictions and so on and I will assume you have read this since that means I don’t have to repeat it all here.

5.15.2 Getting Started

[AR. more to come]

5.15.3 Supported Environments

• Standard
• Title
• Author
• E-mail
• Affiliation
5 Special Document Classes

- Abstract
- Section
- SubSection
- Caption

5.15.4 Differences Between Screen and Paper

There are slight differences in appearance mainly with the presentation of section counters. On screen the trailing period of the section counter is missing but it will appear in the output so don’t let this worry you.

5.16 Memoir

By Jürgen Spitzmüller

5.16.1 Overview

Memoir is a very powerful and constantly evolving class. It has been designed with regard to fictional and non-fictional literature. Its aim is to let the user have maximum control over the typesetting of his document. Memoir is based on the standard book class, but it can also emulate the article class (see below).

Peter Wilson, the developer of Memoir, is known as the author of lots of useful packages in the \TeX world. Most of them have been merged with Memoir. Therefore, it is much easier to layout the table of contents, appendices, chapter designs and such. LyX, though, does not support all of these goodies natively. Some of them might be added to forthcoming releases\footnote{You are invited to send suggestions to lyx-devel@lists.lyx.org.}, lots will probably never, due to the limitations of LyX’s framework. Of course you can still use all features with the help of some native \TeX commands (\TeX code\footnote{Cf. section 2.3 for details.}). In this section, we can only list those features which are natively supported by LyX. For detailed descriptions (and for the rest of features) we are recommending to have a look at the detailed manual of the Memoir class\footnote{Cf. CTAN:/macros/latex/memoir/memman.pdf.}, which is not only a user guide for the class, but also both a comprehensive description on good typesetting and a superb example for good typesetting itself.

5.16.2 Basic features and restrictions

Memoir supports basically all features of the standard book classes. There are, however, some differences, as follows:

---

244
Font sizes: Memoir has a broader range of font sizes: 9, 10, 11, 12, 14, 17

Page style: The fancy page style is not supported, due to a command clash between Memoir and the fancyhdr package (they are both defining a command with the same name, which confuses \TeX). Instead, Memoir comes with a bunch of own page styles (see Layout > Document > Page Style). If you want to use these for the chapter pages, you have to use the command `\chapterstyle` in the main text or in preamble (e.g. `\chapterstyle{companion}).

Sectioning: Sectionings (chapter, section, subsection etc.) are coming with an optional argument in the standard classes. With this, you can specify an alternative version of the title for the table of contents and the headers (for instance, if the title is too long). In \LaTeX, you can do this via Insert > Short Title at the beginning of a chapter/section. Memoir features a second optional argument and thus separates the table of contents from the header. You can define three variants of a title with this: one for the main text, one for the table of contents, and one for the headers. Simply insert two optional arguments if you need this feature, the first one containing the short title for the Table of Contents, the second one containing an alternative short title for the headers.

TOC/LOT/LOF: In the standard classes (and in many other classes), the table of contents, the list of figures and the list of table start a new page automatically. Memoir does not follow this route. You have to insert a page break yourself, if you want to have one.

Titlepage: For some unknown reason, Memoir uses pagination on the title page (in the standard classes, title pages are “empty”, i.e. without pagina). If you want an empty title page, type `\aliaspagestyle{title}{empty}` in the preamble.

Article: With the class option `article` (to be inserted in Layout > Document > Extra Options), you can emulate article style. That is, counters (footnotes, figures, tables etc.) will not be reset on new chapters, chapters don’t start a new page (but are—in contrary to “real” article classes—still allowed), parts, though, use their own page, as in book.

Oldfontcommands: By default, Memoir does not allow the use of the deprecated font commands, which have been used in the old \TeX version 2.09 (e.g. `\rm`, `\it`). It produces an error and stops \TeX whenever such a command appears. The class option `oldfontcommands` reallows the commands and spits out warnings instead (which does at least not stop \TeX). Since a lot of packages and particularly Bib\TeX style files are still using those commands, we have decided to use this option by default.
5 Special Document Classes

5.16.3 Extra features

We will only describe the features supported by LyX (which is not much currently). Please consult the Memoir manual\textsuperscript{13} for details.

Abstract: You may wonder why an abstract is an extra feature. Well, it is in book class. Usually books don’t have abstracts. Memoir, however, has. You can use it wherever and how often you like.

Chapterprecis: You may know this from belletristic: The contents of a chapter is shortly described below the title and also in the table of contents (e.g. Our hero arrives in Troia; he loses some friends; he finds others). Chapterprecis does exactly this. It is therefore only sensible below a chapter.

Epigraph: An epigraph is a smart slogan or motto at the beginning of a chapter. The epigraph environment provides an elegant way of typesetting such a motto. The motto itself (text) and its author (source) are divided by a short line. Unfortunately, we have to fool LyX a bit here again, since the environment needs two arguments (text and source). In this case, we have to use curly brackets (in \TeX mode) between the two arguments: \texttt{<smart slogan> \H <author of the slogan>}.

Poemtitle: Memoir has lots of possibilities to typeset poetry (up to very complex figurative poems). LyX can only support a few of them. One is poemtitle, which is a centered title for poems, which will also be added to the table of contents (verse is the standard environment for poems. Memoir has some enhanced versions of verse, but you need to use \TeX code, because they have to be nested inside regular verse environments, which is not possible with LyX).

Poemtitle*: Same as poemtitle, but it adds no entry to the table of contents.

5.17 Article (mwart), book (mwbk) and report (mwrep)

by Tomasz Luczak

The LyX document classes \texttt{article (mwart)}, \texttt{report (mwrep)} and \texttt{book (mwbk)} correspond to the \LaTeX\ document classes \texttt{mwart.cls}, \texttt{mwrep.cls} and \texttt{mwbk.cls}, resp. They are replacements for the standard document classes \texttt{article.cls}, \texttt{report.cls} and \texttt{book.cls}, resp., and fit better to Polish typography conventions in a number of points.

Basic differences:

- Unnumbered titles (with star, e.g. Section*) are added into table of contents,

\textsuperscript{13}Cf. CTAN:/macros/latex/memoir/memman.pdf.

246
5.18 Paper

The document class paper provides an alternative to the standard article class. It provides similar functionality, but you might prefer this layout with sans serif sections, headings, and more.

5.19 RevTEX4

by Amir Karger

The Revtex 4 textclass works with the American Physical Sociey’s RevTEX 4.0 (the \( \beta \) release of May, 1999) class.

\LaTeX{} has a Revtex textclass, which works with RevTeX 3.1. However, v3.1 is basically obsolete, as it works with \LaTeX{} 2.09. That means that it doesn’t interact very well with \LaTeX{}, which requires \LaTeX{} 2\( \varepsilon \), although it has been kludged to work. Since Rev\LaTeX{} 4.0 has been designed to work much more cleanly with \LaTeX{} 2\( \varepsilon \), \LaTeX{} with the Rev\LaTeX{} 4 textclass should also be pretty easy to use.

These documents are supposed to be used in addition to the Rev\LaTeX{} 4.0 documents, so we don’t describe any of the special Rev\LaTeX{} macros, and assume you’ll know what to put in the preamble if necessary.

5.19.1 Installation

All you need to do is install Rev\LaTeX{} 4, as described in the package’s README file. The package can be found at The Rev\LaTeX{} 4 Web Site http://publish.aps.org/revtex4/. Install it somewhere that \LaTeX{} can see it. Test it by trying to \LaTeX{} a
5 Special Document Classes

short RevTex 4 document in some random directory (i.e. not the directory where you installed the class file.) Then, if you reconfigure LyX, it will find the class file and let you use the RevTeX4 textclass.

Probably the easiest way to get started is either to import a RevTeX 4 document using tex2lyx, or to use the Revtex 4 template, found in the templates directory.

5.19.2 Preamble Matter

Optional arguments to \documentclass, like “preprint” and “aps”, go in the Extra Options field in the Document Layout dialog, as usual. Remember that in RevTeX, at least one optional argument is required!

Other preamble matter, like \draft etc. goes in the LaTeX Preamble dialog, also as usual.

5.19.3 Layouts

The layouts basically correspond to the commands in RevTeX4.0. For example, the Email layout corresponds to \email{}. Note that (at least as of RevTeX 4.0 Beta), the Address and Affiliation layouts are exactly equivalent, so you shouldn’t need to use both.\footnote{In case you’re curious, both were included so that \texttt{tex2lyx} would be able to translate both \texttt{\address} and \texttt{\affiliation}.}

5.19.4 Important Notes

There are a couple of important unique aspects of RevTeX 4 which might cause bugs that will be even more confusing in LyX.

In RevTeX, the thanks command goes outside the author command. The LyX equivalent is that there is a separate Thanks layout. Do not write footnotes in the Author layout, or weird things may happen. See the RevTeX 4 documentation for more details.

Also, the Author Email, Author URL, and Thanks layouts must be placed in between the Author layout and the corresponding Address (or equivalent Affiliation) layout. If you put the Thanks after the Address, the \LaTeX won’t compile.

5.19.5 Drawbacks

The main problem with this layout is that you can’t use the optional arguments to layouts like Email and Title. (The problem is not unique to this layout; you can’t use optional arguments to the Section layouts either.) This means that after you export that file to \LaTeX (which you’ll need to do eventually to send it in to APS), you’ll need to edit the \LaTeX file with a text editor to add the optional arguments to set, e.g. the running title for the page headers. Lacking these layouts makes the
\altaffiliation (and the equivalent \altaddress) useless, so the corresponding layouts don't exist, and will have to be added by hand.\footnote{Note from JMarc: actually, \LaTeX 1.3.0 supports some forms of optional arguments, but this layout has not been updated yet to take advantage of it.}

5.20 Springer Journals (svjour)

by Martin Vermeer

5.20.1 Description

These are the layout files for some of the journal formats used by Springer Verlag and listed on \url{http://www.springer.de/author/tex/help-journals.html}, where you should also go to fetch the class files (yes, these are \LaTeX 2e now!). It is a modular system: the things common to all journals are implemented in \texttt{svjour.inc}, which journal-specific layout files (such as, e.g. \texttt{svjog.layout} for Journal of Geodesy) can include.

This means that implementing support for any other Springer journal on this list is as simple as writing your own \texttt{sv<myjournal>.layout} file following the outline given in \texttt{svjog.layout}.

It is reasonably well tested only for the Journal of Geodesy. \texttt{svjour} and \texttt{svjog} come with the standard \LaTeX\ X distribution. Install the relevant class file (downloaded from Springer) in a proper directory, reconfigure \TeX\ (in the \texttt{te\TeX} case by running \texttt{texhash}, as root if necessary — doesn’t \LaTeX\ X take care of this?), reconfigure \LaTeX\ X and it should work.

5.20.2 New styles

A large number of theorem-like styles — Claim, Conjecture, … Theorem.

Headnote, Dedication, Subtitle, Running_{\LaTeX} Title, Author Running, Institute, Mail, Offprints, Keywords, Acknowledgements, Acknowledgement. See the Springer class file documentation for details.

5.20.3 Supported journals

- \textit{Journal of Geodesy}: \texttt{svjog.layout} — Martin Vermeer

- \textit{Probability Theory and Related Fields}: \texttt{svprobth.layout} — Jean-Marc Lasgouttes

Add your own, it isn’t so hard!
5 Special Document Classes

5.20.4 Credits

These files are partly based on the older ejour2.layout, which was again based on a tinkered-with version of an old \LaTeX 2.09 style file from Springer. All this, and the ejour2 layout, are now defunct. Jean-Marc Lasgouttes helped out big in making me find my way around the \LaTeXX layout file mechanism.

5.20.5 Bugs

Probably. But probably less than in the old hacked-\LaTeX ejour2.

Limitations e.g.: does not display the number for theorem-like layouts, just #.

5.21 Slides [aka Sli\TeX]

by JOHN WEISS

5.21.1 Introduction

This section describes how to use \LaTeXX to make slides for overhead projectors. There are two document classes that can do this: the default slides class and the Foil\TeX slides class. This section documents the former.

I’m going to say this again, nice and clear, so that there’s no misunderstanding:

This section documents the class “slides (default)” only.

If you’re looking for the documentation for “slides (Foil\TeX)”, check out section 5.9. The foils class [“slides (Foil\TeX)”] is actually somewhat better than the default slides class,\footnote{...or so I’ve been told repeatedly by its advocates. Having never used it, I have no idea if this claim is true or not.} which this section documents.

This class is the \LaTeX2ε improvement of the old Sli\TeX package. Every \LaTeX2ε distribution includes this class [which I’ll just refer to as “slides” from now on], so you’re bound to have it. As I noted earlier, there are other classes, such as foils, which also produce slides for overhead projectors and do a better job at it. However, there are some things which slides can do which the others can’t, such as generate overlays. Read on to learn more!

5.21.2 Getting Started

Obviously, to use this document class, you need to select “slides (default)” from the class list in the Document>Settings dialog. There are some other special things you should know about this class:

If you’re looking for the documentation for “slides (Foil\TeX)”, check out section 5.9. The foils class [“slides (Foil\TeX)”] is actually somewhat better than the default slides class,\footnote{...or so I’ve been told repeatedly by its advocates. Having never used it, I have no idea if this claim is true or not.} which this section documents.

This class is the \LaTeX2ε improvement of the old Sli\TeX package. Every \LaTeX2ε distribution includes this class [which I’ll just refer to as “slides” from now on], so you’re bound to have it. As I noted earlier, there are other classes, such as foils, which also produce slides for overhead projectors and do a better job at it. However, there are some things which slides can do which the others can’t, such as generate overlays. Read on to learn more!

250
• Don’t bother changing the options Sides and Columns. They’re not supported by the slides class, anyways.

• The option Page style behaves a bit differently for this class. The possible choices and what they do are as follows:
  - **plain** The final output contains page numbers in the lower right corner.
  - **headings** Like plain, but also prints out any time markers you’ve put in. This is the default.
  - **empty** The final output contains no page numbers, time markers, or alignment markers.

• The slides class has an extra option: **clock**. To use it, put “clock” in the extra class options.
  Using this options allows you to add time markers to Notes. See section 5.21.4.3 for more details.

You can also use the template file “slides.lyx” to automatically set up a document to use the slides class [using File> New from Template to open your new document]. The template file also contains some examples of the special paragraph environments used by this class. I’ll describe those next.

### 5.21.3 Paragraph Environments

#### 5.21.3.1 Supported Environments

The first thing you’ll notice when you start up a new slides document is the font size and type: it’s the equivalent of the size “Largest” in the Sans Serif font. This is also what’s used in the output. Think of this as a “visual cue” to remind you that this is a slide. Your final slides will use a larger font; ergo, you’ll have less space. Of course, the larger default screen font isn’t WYSIWYG, only a reminder.

The next thing that becomes obvious is the changes to the paragraph environment pull-down box [at the far-left end of the toolbar]. Most of the paragraph environments you’re used to seeing are missing. There are also five new ones. That’s because the slides class itself only supports certain paragraph environments:

- Standard
- Itemize
- Enumerate
- Description
- List
- Quotation
All of the other standard environments, including the section-heading environments, aren’t used in the `slides` class.

On the other hand, you’ll notice the following new environments:

- Slide
- Overlay
- Note
- InvisibleText
- VisibleText

These five are kind of quirky, due to a “feature” in LyX. You see, LyX doesn’t permit you to nest any other paragraph environment into an empty environment. Now, that’s fine and dandy, but it means that you wouldn’t be able to start a slide with anything except plain text. To deal with this, I’ve performed a little “\LaTeX magic.”

### 5.21.3.2 Quirks of the New Environments

All five of the new paragraph environments are somewhat quirky due to inherent limitations in the current version of LyX. As I just mentioned, LyX forbids environments that begin with another environment. To get around this, the `Slide` environment isn’t a paragraph environment as described in the User’s Guide.

You should consider `Slide`, `Overlay`, and `Note` to be “pseudo-environments.” They look like a section heading or a “Caption,” but really begin a [and, if necessary, end the previous] paragraph environment. Likewise, treat `InvisibleText` and `VisibleText` as “pseudo-commands.” These two perform some action.

A common feature of all five environments, `Slide`, `Overlay`, `Note`, `InvisibleText` and `VisibleText`, is a rather long-ish label. The text following this label — ordinarily the contents of the paragraph environment — is utterly irrelevant for `Slide`, `Overlay`, `Note`, `InvisibleText` and `VisibleText`. LyX completely ignores it. In fact, you can leave these five environments completely empty.

While you don’t have to put any text after the rather long-ish label, you might want to. This could be a short description of the contents of the `Slide`, for example. In that case, enter in your descriptive comment and hit Return as you normally would.
If, on the other hand, you don’t want to enter in any descriptive text, you’ll hit another LyX quirk. LyX, like nature, abhors a vacuum, and will not let you start a new paragraph environment until you put something in the old one. So, do this:

- Start entering the text that will follow the new Slide, Overlay, Note, InvisibleText or VisibleText.
- Now move to the beginning of that paragraph.
- Next, hit Return.
- Finally, change this new, empty paragraph to a Slide, Overlay, Note, InvisibleText or VisibleText.

Some future version of LyX will, hopefully, resolve this quirkiness.

5.21.4 Making a Presentation with Slide, Overlay and Note

5.21.4.1 Using the Slide Environment

If you’re expecting this section to teach you how to actually make a presentation, you’ll be sorely disappointed. Naturally, I’ll describe all of the ways the slides class can assist you in preparing the materials for a presentation. Filling in the contents, however, is up to you. [Then again, that is the LyX philosophy.]

Choosing the Slide environment [in the manner described in section 5.21.3.2] tells LyX to begin a new slide [duh]. The label for this environment/”pseudo-command” is an “ASCII line,” in cool blue, followed by the label, “NewSlide:”. Any text or paragraph environments that follow this one go on the new slide. It’s that simple.

Slides are probably the only time you’ll need to forcibly end pages in LyX (this can be specified in the Paragraph Layout dialog). In fact, you’ll want to, once you finish entering the contents of one slide. If you’ve entered more text than can physically fit on a slide, the extra overflows onto a new slide. I don’t recommend doing this, however, since the overflow slide won’t have any page number on it. Furthermore, it may interfere with any Overlay you’ve made to accompany the oversized Slide.

The Overlay and Note environments work the same way as the Slide environment. They both create an “ASCII line” followed by a label [“NewOverlay:” and “NewNote:”, respectively]. The color is a stunning magenta instead of blue, and the “ASCII line” will look different, in style and in length. The label fonts of all three also differ from one another.

As with a Slide, if the contents of a Note or Overlay exceed the physical size of a slide or sheet of paper, the extra will overflow onto a new sheet. Again, you should avoid this. It defeats the whole purpose of Notes and Overlays.

5.21.4.2 Using Overlay with Slide

The idea behind an Overlay is a slide that sits atop another slide. Perhaps you wish to discuss a figure on the main Slide before displaying the text associated with it.
One way to accomplish this is tape a flap of dark paper over the part of the Slide you want to display later. This method fails, however, if you wish to overlap one graph with another, for example. You would then have to fumble while speaking to align the two separate, overlapping Slides to align the two graphs. The use of an Overlay environment in both cases makes life much easier.

Each Overlay receives the page number of its “parent” Slide, appended by “–a”.17 Clearly, you want the contents of both the Slide and the Overlay to each fit on a single physical slide! You should probably consider an Overlay as “part of” a Slide. Indeed, the LyX slides class provides a visual cue for this: the label at the start of an Overlay is shorter than that at the start of a Slide. Lastly, when you generate printable output, you’ll find alignment markers in all four corners of both the Overlay page and its parent Slide. These will assist you in lining up the two physical slides.

The major problem in overlaying two slides is aligning the contents of the two transparencies. How much space should you leave for that graph on the second slide? Worse still, what if you want a graph and a sentence on second slide, but there is text on the main transparency that goes in between them? You could try and insert vertical space of the right size. The better way is to use InvisibleText and VisibleText.

As their names imply, InvisibleText and VisibleText are two command-like paragraph environments that make all subsequent text invisible and visible, respectively. Note from section 5.21.3.2 that you don’t place anything into these two environments, however. When you create an InvisibleText, it inserts a centered, sky-blue label into the page reading “<Invisible Text Follows>”. For paragraphs following this label, the parts of the Slide [or Overlay; it doesn’t matter which] where they would be contain instead blank space.

For VisibleText, the corresponding centered label is “<Visible Text Follows>” in blazing green. Paragraphs following this label behave normally. Note that the beginning of a new Slide, Overlay, or Note automatically shuts off an InvisibleText. It’s therefore not necessary to use VisibleText at the end of a Slide.

By now, it should be obvious how to create overlay transparencies using the proper combination of InvisibleText and VisibleText on a Slide and Overlay:

1. Create a Slide, including everything that will appear on it, whether on the main slide or on the Overlay.

2. Before each figure or paragraph that will appear only on the Overlay, insert an InvisibleText environment. If necessary, insert a VisibleText environment after the Overlay-only text.

3. Start an Overlay immediately following the Slide.

4. Copy the contents of this Slide into the Overlay.

5. Within the Overlay, change all of the InvisibleText lines to VisibleText and vice-versa.

17Presumably, multiple Overlays would have “–a”, “–b”, “–c”, etc. appended to the page number of the parent Slide.
That’s it. You’ve just made an Overlay.

There’s one problem with the way I’ve designed the LyX slides class: you can’t make text in the middle of a paragraph invisible, nor make text in the middle of an invisible paragraph visible again. To accomplish this feat, you’ll need to use some inlined \LaTeX{} codes.\footnote{The commands of interest are:}

\begin{itemize}
  \item {\texttt{\invisible ... }}
  \item {\texttt{\visible ... }}
\end{itemize}

\ldots and need to be marked as \TeX{}. The text whose “visibility” you wish to change goes in between the brackets [and after the \texttt{\invisible} or \texttt{\visible} command]. If you don’t know how to mark text as \TeX{}, see the appropriate section of the User’s Guide.

\section*{5.21.4.3 Using Note with Slide}

Like an Overlay, a Note is associated with a “parent” Slide. Here, too, the LyX slides class provides visual cues. The label for a Note is shorter than that of a Slide [yet longer than that of an Overlay] and, like the label of an Overlay is shockingly magenta. Additionally, the printed Note has the page number of its “parent” Slide, appended by “-1”, “-2”, “-3”, etc. You can have multiple Notes associated with a single Slide, and, as with Slide and Overlay, you’ll probably want to break up long Notes so that they fit on a single sheet of paper.

The purpose of a Note is obvious: it contains anything additional you might want to say about a Slide. It could also be used as a sheet of reminders for a particular Slide. In the case of the latter, you might want to make use of time markers. Currently, the LyX slides class has no “native” support for time markers, a \LaTeX{} feature. So, you’ll have to resort to using the \LaTeX{} codes.

To use time markers, you’ll need to specify the extra class option “clock” [see section 5.21.2]. This option turns on timing marks, which will appear in the lower-left-hand corner of every Note you generate. To set what appears in the time marker, you use the $\LaTeX$ commands \texttt{\settime{}} and \texttt{\addtime{}}. The arguments of both commands are time measured in seconds. \texttt{\settime{}} sets the time marker to a given time. \texttt{\addtime{}} increments the time marker by the specified amount.

Using time markers and Notes in this fashion, you can remind yourself how much time to spend on a particular Slide.

There’s one last feature to describe. Clearly, you’d like to print out all of your Slides and Overlays on transparencies while printing all of your Notes on plain paper. However, a Note must follow the Slide with which it is associated. What’s a person to do?

Luckily, there are two \LaTeX{} commands that allow you to select what to print out. Both must be placed into the preamble of your document. The command \texttt{\onlyslides{\slides}} will cause the output to contain only the Slides and Overlays. Correspondingly, the command \texttt{\onlynotes{\notes}} prevents the output of anything but Notes. I’d advise placing both commands in the preamble and ini-
tially comment both out. You can then preview your entire presentation as you write. When you’re done writing, you can then uncomment one of the two to select what you want to print. I like to uncomment “\onlyslides{\slides}”, print to a file with “-slides” in its name, comment it back out, then uncomment “\onlynotes{\notes}” and print to a “*-notes.ps” file. I can then send either file to a printer, loading transparencies or plain paper as appropriate.

You can also provide other arguments to the “\onlyslides{}” and “\onlynotes{}” commands. See a good \LaTeX book for details.

5.21.5 The slides Class Template File

I have also provided a template file, “slides.lyx”, with the \slides class. To use it, begin your new presentation with File \texttt{New from Template}. Your new \lyx presentation file will contain an example \texttt{Slide – Overlay – Note} triplet. The \texttt{Slide} and \texttt{Overlay} additionally contain an example of the use of \texttt{InvisibleText} and \texttt{VisibleText}. Lastly, the preamble will contain:

\begin{verbatim}
% Uncomment to print out only slides and overlays
% \onlyslides{\slides}

% Uncomment to print out only notes
% \onlynotes{\notes}
\end{verbatim}

One final thing: I created this class to support the \LaTeX2ε “\\LaTeX emuation” class, one of the built-in \LaTeX2ε classes. Neither I nor the rest of the \lyx Team endorse or oppose the use of this built-in slide class. It’s here if you want it or need it. There exist other \LaTeX2ε classes for creating presentations, such as the \texttt{Foils} class [see section 5.9] or the “\texttt{seminar}” package [present on some \TeX distributions]. The latter is not yet supported under \lyx.\footnote{Perhaps you can take on the task…} I know nothing about these other classes. Try them out to see what sort of alternative they provide.
6 LyX Features needing Extra Software

6.1 Checking TeX

by Asger Alstrup

6.1.1 Introduction

If you have the chktex program installed, you’ll find in the Tools menu the entry: Check TeX. You can get chktex it from CTAN, http://www.ctan.org/tex-archive/help/Catalogue/entries/chktex.html.

The ChkTeX package is a program that was written by Jens T. Berger Thielemann in frustration because some constructs in LaTeX are sometimes non-intuitive, and easy to forget. The program runs over your LaTeX file, checks the integrity of the file, and flags some common errors. In other technical words, it is lint for LaTeX.

Well, what is a syntax checker doing in LyX which is supposed to produce correct LaTeX anyways? The answer is simple: Just as lint not only checks the syntax of C programs, but also does semantic checks for type-errors, ChkTeX catches some common typographic errors, in addition to the syntactical ones. Specifically, ChkTeX is capable of detecting several common errors, such as

- Ellipsis detection:
  Use ... instead of ...

- No space in front of/after parenthesis:
  ( wrong spacing )

- Enforcement of normal space after common abbreviations:
  e.g. is too wide spacing.

- Enforcement of end-of-sentence space when the last sentence ends with a capital letter:
  This is a TEST. And this is wrong spacing.

- Space in front of labels and similar commands:
  The label should stick right up to the text to avoid falling to a wrong page. The label is separated too much.

---

1 chktex is not yet available when you are using the LaTeX distribution MiKTeX.
2 This footnote is in danger of falling off to a wrong page
6 \LaTeX Features needing Extra Software

• Space in front of references, instead of hard spaces:
  In you are in bad luck, the text will break right between the referenced text
  and reference number, and that’s a pity. See section 6.1.1.

• Use of “x” instead of \times between numbers:
  2x2 looks cheap compared to $2 \times 2$.

and more … It is an invaluable tool when you are “finishing up” your document
before printing, and you should run it right after the obligatory spelling check, and
before you go fine tuning the typesetting.

6.1.2 How to use it

If you have the program installed, usage is as simple as choosing Tools\rightarrow Check \TeX.  
This will make \LaTeX generate a \LaTeX file of your document, start Chk\TeX to check
it, and then make \LaTeX insert “error boxes” with the warnings from Chk\TeX, if there
were any. The warnings will be placed close to the point of the mistake, and you can
quickly find them by using the Navigate\rightarrow Error menu item, or the shortcut key C-g
from the default cua bind file. Open the error boxes by clicking on them with the
mouse, or use the shortcut key C-i from cua bindings, or the corresponding C-o for
the alternate emacs bind file. Read the warning and correct the mistake, if it is a
mistake. If you have trouble understanding what the warning is about, you can safely
ignore it. Remember that there is a hidden layer between the document on screen
and the technical details in invoking Chk\TeX, and this gap can make some warnings
seem arcane or just right down plain silly.

This document is an excellent testing bed for the feature, and it should provide
quite a few warnings for you to fiddle with. Since computers are only so smart, expect
most of the warnings to be false alarms, though.

6.1.3 How to fine tune it

Sometimes, you’ll find that Chk\TeX makes more noise than suits your mood. Then
you can choose not to use it, wait until your mood changes, or try to customize Chk\TeX
to get better along with you. Another choice in the most desperate situations is to use View\rightarrow Remove All Error Boxes, which will get rid of all warnings instantly.

Although Chk\TeX is very configurable and extensible, you shouldn’t expect to solve
all problems with Chk\TeX in \LaTeX this way. Since \LaTeX has to generate a somewhat
special \LaTeX file to be able to match the line numbers from the Chk\TeX output\footnote{You can inspect the specific output from chktex by using Edit\rightarrow View \LaTeX Log right after a chktex run.} to the internal document structure, some of the warnings will not seen to appear
correctly. There are two things you can do about this:

• Fine tune the Chk\TeX invocation command line in Preferences (tabs Outputs,
  Misc), or the global Chk\TeX installation configuration file (usually with the file

258
6.1 Checking \TeX

\texttt{chktexrc}). See below to learn what warnings can be enabled and disabled on the command line.

- Export your document as a raw \LaTeX file using \texttt{File \textgreater Export \textgreater \LaTeX} and run \texttt{chktex} manually on that. Invoked in this way, it can be a hassle to find the corresponding place in the document inside LyX, but with a little patience, you should be able to do it.

Here follows the warning messages that can be enabled and disabled in Preferences. Use \texttt{-n#} to disable a warning, and \texttt{-w#} to enable a warning. The emphasized entries are disabled by default, because the default is "\texttt{chktex -n1 -n3 -n6 -n9 -n22 -n25 -n30 -n38}".

Notice that you should only use the options that enable and disable warnings, because LyX relies on some of the other command line parameters to be set in a specific way to have a chance to communicate with \texttt{chktex}.

1. \textit{Command terminated with space.}
2. Non-breaking space ("~") should have been used.
3. \textit{You should enclose the previous parenthesis with "\{\}.}
4. Italic correction ("\slash") found in non-italic buffer.
5. Italic correction ("\slash") found more than once.
6. \textit{No italic correction ("\slash") found.}
7. Accent command "\texttt{cmd}" needs use of "\texttt{cmd}".
8. Wrong length of dash may have been used.
9. "\texttt{\%s}" expected, found "\texttt{\%s}".
10. Solo "\texttt{\%s}" found.
11. You should use "\texttt{\%s}" to achieve an ellipsis.
12. Inter-word spacing ("\texttt{\textbackslash "}) should perhaps be used.
13. Inter-sentence spacing ("\texttt{\textbackslash @}") should perhaps be used.
15. No match found for "\texttt{\%s}".
16. Math mode still on at end of \LaTeX file.
17. Number of "\texttt{char}" doesn’t match the number of "\texttt{char}".
18. You should use either "\texttt{\textquoteleft\textquoteleft}" or "\texttt{\textquoteleft\textquoteleft}" as an alternative to "\textquoteleft\textquoteleft".
19. You should use "'" (ASCII 39) instead of "'" (ASCII 180).

20. User-specified pattern found.

21. This command might not be intended.

22. Comment displayed.

23. Either "," or "," will look better.

24. Delete this space to maintain correct page references.

25. You might wish to put this between a pair of “Ω”.

26. You ought to remove spaces in front of punctuation.

27. Could not execute \LaTeX command.

28. Don’t use \ in front of small punctuation.

29. $\times$ may look prettier here.

30. Multiple spaces detected in output.

31. This text may be ignored.

32. Use " to begin quotation, not ‘.

33. Use ‘ to end quotation, not ".

34. Don’t mix quotes.

35. You should perhaps use “cmd” instead.

36. You should put a space in front of/after parenthesis.

37. You should avoid spaces in front of/after parenthesis.

38. You should not use punctuation in front of/after quotes.

39. Double space found.

40. You should put punctuation outside inner/inside display math mode.

41. You ought to not use primitive \TeX in \LaTeX code.

42. You should remove spaces in front of “%s”

43. “%s” is normally not followed by “%c”.

In later versions of LYX, we hope to provide a more complete interface to this tool (and it’s smaller cousin lacheck) to exploit the full power of it. But it’s not exactly useless as it is now: go try it on one of your existing documents of a certain length and be surprised.
6.2 Version Control in LyX

by Lars Gullik Bjønnes, updated by Pavel Sanda

6.2.1 Introduction

A friend of mine wanted to try LyX for a group project. When he didn’t find support for version control or file locking, he dropped it. This angered me a bit, so I thought that I should at least make support for RCS (with the possibility of CVS and/or SCCS as a future improvement.) This has now been done. LyX now supports some of the most basic RCS commands. If you need to something a bit more sophisticated you will have to do that manually in an xterm.

Before you begin to use the version control features in LyX, you should read “rcsintro” (a man file, read it with man rcsintro). This file describes all the basic features of RCS. You should especially notice the comment about a RCS directory, and the notion of a master RCS file (the file ending in ,v).

Later basic CVS/SVN support was added. You should be familiar with CVS/SVN usage before start using it under LyX. Most of the log messages are not currently displayed after operations - you can check them in terminal window if unsure.

The implementation in LyX assumes a recent version of the GNU RCS or CVS/SVN package—no guarantees are made for older versions.

For introducing your own external commands consult vc-command in the manual of LyX functions.

6.2.2 RCS commands in LyX

The following sections describe the RCS commands supported by LyX. You can find them in the File > Version Control submenu. LyX was tested against RCS 5.7.

6.2.2.1 Register

If your document is not under revision control, this is the only item shown in the menu. And if it is under revision control, the Register item is not visible.

This command registers your document with RCS (unless you are under the directory managed by CVS). You are asked interactively to supply an initial description of the document. The document is now set in Read-Only mode and you have to Check Out For Edit, before making any changes to it. A document under revision control has a “[RCS:<version> <locker> ]” item tagged to the filename in the minibuffer.

RCS command that is run: ci -q -u -i -t-"<initial description>" <file-name>

Read man ci to understand the switches.
6.2.2.2 Check In Changes
When you are finished editing a file, you check in your changes. When you do this, you are asked for a description of the changes. This is stored in the history log. The version number is bumped, your changes are applied to the master RCS file, the document is unlocked and set to Read-Only mode.
RCS command: `ci -q -u -m"<description>" <file-name>`

6.2.2.3 Check Out For Edit
By doing this you lock the document so that only you can edit it. This will also make the document Read-Write only for you. You will usually continue editing for a while and when you are finished you check in your changes. The status line is changed to reflect that you have locked the file.
RCS command: `co -q -l <file-name>`

6.2.2.4 Revert To Repository Version
This will discard all changes made to the document since the last check in. You get a warning before changes are discarded.
RCS command: `co -f -u<version> <file-name>`

6.2.2.5 Undo Last Checkin
This makes as if the last check in never happened. No changes are made to the document loaded into LyX, but the last version is removed from the master RCS file.
RCS command: `rcs -o<version> <file-name>`

6.2.2.6 Show History
This show the complete history of the RCS document. The output of `rlog <file-name>` is shown in a browser. See `man rlog` for more info.

6.2.3 CVS commands in LyX
CVS is now partially supported by LyX. You can find the commands in the File> Version Control submenu.

6.2.3.1 Register
If your document is not under revision control, this is the only item shown in the menu. And if it is under revision control, the Register item is not visible.
This command registers in CVS your document ONLY in case you have already the documents directory under CVS control (in particular `CVS/Entries` file exists). This means you have to checkout the archive by yourself.
Then you are asked interactively to supply an initial description of the document. Don’t forget that registered file is not yet commited.

CVS command that is run: `cvs -q add -m"<entered message>" "<file-name>"`

Read `man svn` to understand the switches.

### 6.2.3.2 Check In Changes

When you are finished editing a file, you commit your changes. When you do this, you are asked for a description of the changes. After that changes are commited.

CVS command: `cvs -q commit -m"<description>" "<file-name>"`

### 6.2.3.3 Revert To Repository Version

This will discard all changes made to the document since the last check in. You get a warning before changes are discarded. Firstly the file is deleted, secondly CVS update command is run.

CVS command: `cvs update "<file-name>"`

### 6.2.3.4 Show History

This show the complete history of the CVS document. The output of `cvs log "<file-name>"` is shown in a browser.

### 6.2.4 SVN commands in LYX

SVN is now partially supported by LYX. You can find the commands in the File> Version Control submenu. Please note that if you use password protected access to repository via ssh, you will be asked in terminal window. LYX was tested against SVN 1.4 and 1.5.

#### 6.2.4.1 Register

If your document is not under revision control, this is the only item shown in the menu. And if it is under revision control, the Register item is not visible.

This command registers in SVN your document ONLY in case you have already the documents directory under SVN control (in particular `.svn/entries` file exists). This means you have to checkout the archive by yourself.

Then you are asked interactively to supply an initial description of the document. Don’t forget that registered file is not yet commited.

SVN command that is run: `svn add -q "<file-name>"`

Read `man svn` to understand the switches.
6.2.4.2 Check In Changes

When you are finished editing a file, you commit your changes. When you do this, you are asked for a description of the changes. After that changes are committed.

   SVN command: \(^{4}\) svn commit -q -m"<description>" <file-name>

6.2.4.3 Check Out For Edit

Updates the changes of this file from the repository. Be sure you understand SVN merging and conflicts resolving before using this function, because all conflicts has to be done manually by you!

   SVN command: \(^{5}\) svn update "<file-name>"

6.2.4.4 Revert To Repository Version

This will discard all changes made to the document since the last check in. You get a warning before changes are discarded.

   SVN command: svn revert -q "<file-name>"

6.2.4.5 Show History

This show the complete history of the SVN document. The output of \texttt{svn log} "<file-name>" is shown in a browser.

6.2.4.6 File Locking

The file exchange through various revision control systems brings the problem of merge conflicts in case two different users try to edit the same (parts of) document. When such conflict happens it needs manual resolving and one reasonable alternative is to provide some kind of locking mechanism, which guarantees that only one user is allowed to edit file at the given time.

   SVN has two mechanisms to provide such kind of mutual exclusivity for file access - locks and automatical setting of write permissions (see sec. 6.2.4.7) based on \texttt{svn:needs-lock} file svn property\(^{6}\). In a case this property is detected for a given document \LaTeX{} starts to use SVN locks for document editing automatically and the whole check-in/out mechanism switches to the same regimen as for RCS. This in particular means there are two different modes how file is used in \LaTeX{}:

   - Unlocked state. The loaded file is in the read-only mode. For editation on needs to check-out. \textit{Check-out} consists of update from repository and gaining write lock. If the lock is not possible to obtain, we remain in unlocked state.

\(^{4}\)In case locking is not enabled. See Section 6.2.4.6.
\(^{5}\)Ditto.
\(^{6}\)http://svnbook.red-bean.com/en/1.2/svn.advanced.locking.html
• Locked state. The loaded file is in the 'normal' edit mode. No other user is allowed to edit the file. Check-in consists of committing changes and releasing write-lock. If no changes have been made to the document, no commit will be produced\(^7\) and only the write-lock will be released.

SVN commands:

Check-in: `svn commit -q -m"<description>" "<file-name>"
svn unlock "<file-name>"

Check-out: `svn update "<file-name>"
svn lock "<file-name>"

6.2.4.7 Automatical Locking Property

The above mentioned automatical setting of write permissions of the .lyx file can be set through `File > Version Control > Toggle locking property`. This command is active only when the file is not locked on the svn server (i.e. you need to check-out before proceeding).

SVN commands:

Set: `svn propset svn:needs-lock ON "<file-name>"

Unset: `svn propdel svn:needs-lock "<file-name>"

6.2.4.8 Revision Information in Documents

Currently there is no way how to provide such kind of information directly from LyX. There are possibilities how to activate it with the help of svn features, but each has its own drawbacks.

One possibility is to use svn keywords\(^8\). In short – you set file keywords property (e.g. `svn propset svn:keywords 'Rev' file.lyx) and then paste keyword ERT\(^9\) tag in your document (e.g. `Rev`). This way svn client will automatically substitute revision number (e.g. `Rev: 59`) after each update and commit. There are more problems with this approach. Firstly, the '$' character is used in \TeX world for math equations, so any occurrence of math formula `Rev become `Rev : 59` in your LyX document. Similarly for other keywords like Id, Date, Author, etc. Secondly svn output is dependent on your locales, so its very easy that svn would produce some problematic strings once `Date` is used. Thirdly you get the whole 'Rev: 59' string in your document instead of the plain number. Until subversion implements user's custom keywords it will be hard to use this approach reliably or let LyX to support it directly.

The second possibility would be to write your own external-material template which calls either `svnversion` utility or parses the output of `svn info file.lyx` command and returns the result back, when typeseting the document.

\(^7\)Don’t be puzzled by the fact that you will be asked for commit message anyway.
\(^8\)http://svnbook.red-bean.com/en/1.4/svn.advanced.props.special.keywords.html
\(^9\)This is an easy way how to ensure that LyX won’t break the line in the middle of keyword tag.
6.2.5 SVN and Windows Environment

My inclination is to say that if the user cannot figure out the command line operations on their own fairly quickly, they would be well advised to use TortoiseSVN. —P. A. Rubin

6.2.5.1 Preparation

In addition to installing \LaTeX, and having access to a Subversion repository, the user will need to install the Subversion client program. A Windows installer for the client program is available from CollabNet. The user may also want to install TortoiseSVN, which integrates Subversion operations into the context (rightclick) menu of Windows Explorer. Operations done outside \LaTeX will typically be more convenient using the Explorer context menu. Note that TortoiseSVN is not a replacement for the client program, which is what \LaTeX itself will use.

6.2.5.2 Bringing a document under Subversion control

Before a \LaTeX document can be brought under version control in Subversion, its parent directory needs to be under version control. If the document is being added to a project already in the repository, this is accomplished by checking the project out to the directory where the new document will be placed. If the project itself is not yet under version control (for instance, if this document starts a new project), the directory must be imported into the repository. This is done outside \LaTeX. Both import and checkout are easily accomplished from the Explorer context menu using TortoiseSVN, or alternatively can be done using the command line client at a DOS prompt. The procedure for importing the project using TortoiseSVN is described below, assuming an existing repository and a new project being started in C:\new project. For information on using the Subversion client program, run \texttt{svn --help} in a DOS shell.

1. Locate C:\new project in Windows Explorer, right click it, and select TortoiseSVN > Repo-browser. If necessary, adjust the URL for the repository, then click OK.

2. Right click the level of the repository under which you want to place the new project folder (typically the top level) and click Create folder... Supply a name for the project folder and click OK. Add a message for the log file if desired, then click OK again. The new project folder should appear in the repository. Finally, click OK again to exit the repository browser.

3. Once again right click C:\new project, this time selecting SVN Checkout... Select the URL of the project folder you just created in the repository, and set the checkout directory to C:\new project. Click OK. You will be warned about a non-empty folder; click OK to proceed. You should now have a .svn directory under C:\new project.
4. Create or open your document in \LaTeX{} and click File ➤ Version Control ➤ Register. Add a log message and click OK to commit the document to version control.

From this point onward, you should have full functionality in the File ➤ Version Control menu. You also have the option of checking the document in and out, viewing its history, etc. using the TortoiseSVN context menu in Windows Explorer or the Subversion client program from a command prompt.

### 6.2.6 Further tuning

With the recent addition of the vc-command function \LaTeX{} power users are allowed to create their own commands for revision control.

As an example you can see how two TortoiseSVN commands could be integrated directly:

**Commit:** vc-command DR "." "TortoiseProc /command:commit /path:$$p"

**Revert:** vc-command DR "." "TortoiseProc /command:revert /path:$$p"

### 6.3 Literate Programming

Updated by Kayvan Sylvan (kayvan@sylvan.com), original documentation written by Edmar Wienskoski Jr. (edmar-w-jr@technologist.com)

#### 6.3.1 Introduction

The main purpose of this documentation is to show you how to use \LaTeX{} for literate programming. Where it is assumed that you are familiar with this programming technique, and know what “tangling” and “weaving” means. If that is not the case, please follow the web links provided in the following sections. There is a lot of good documentation out there covering old development history to the latest tools tips.

It is also assumed that you are familiar with \LaTeX{} itself to a point that you are comfortable changing your \LaTeX{} preferences, and X resources file. If that is not the case please refer to other \LaTeX{} documentation to cover your specific needs.

#### 6.3.2 Literate Programming

From the Literate Programming FAQ:

Literate programming is the combination of documentation and source together in a fashion suited for reading by human beings. In fact, literate programs should be enjoyable reading, even inviting! (Sorry Bob, I couldn’t resist!) In general, literate programs combine source and documentation in a single file. Literate programming tools then parse the file to produce either readable documentation or compilable source. The
WEB style of literate programming was created by D. g. Knuth during the development of his \TeX\ typesetting software.

Another excerpt says:

How is literate programming different from verbose commenting?
There are three distinguishing characteristics. In order of importance, they are:

- flexible order of elaboration
- automatic support for browsing
- typeset documentation, especially diagrams and mathematics

Now that I sparked your curiosity, take a look in the references.

### 6.3.2.1 References

The complete Literate Programming FAQ can be found at:

Literate Programming FAQ [http://shelob.ce.ttu.edu/daves/lpfaq/faq.html](http://shelob.ce.ttu.edu/daves/lpfaq/faq.html)

The FAQ lists 23 (twenty three!) different literate programming tools. Where some are specialized or “tailored” for particular programming languages, while other have general scope. I selected \texttt{Noweb} for my own use for several reasons:

- It can generate the documentation either in \LaTeX\ or HTML.
- It has a open architecture, i.e. it is easy to plug in new filters and to perform special processing that you may need.
- There is a good selection of filters available already (the HTML is one of them).
- It is free.

The Noweb web page can be found at:


Starting from there you can reach many other interesting links and even some literate program examples.

### 6.3.3 \texttt{LYX} and Literate Programming

The \texttt{LYX} support for Literate Programming is provided by using the generic \texttt{LYX} converters mechanism. This support is provided in a “Noweb independent” way, i.e. you will be able to use this new \texttt{LYX} feature with some other literate programming tool of your choice by just changing your \texttt{LYX} preferences.
6.3.3.1 Generating documents and code (weaving and tangling)

Selecting the document class  If you have installed Noweb and LyX successfully, whenever you open a new document or try to change the document class of an existing one, you will find that there are three new document classes available:

- Article (Noweb)
- Book (Noweb)
- Report (Noweb)

You must select one of them to create your literate documents from.

Note that literate documents are not limited to these three classes. New classes can be generated from other styles like letter or in combination with other class variations like Article (AMS). If you have special needs that cannot be covered by one of the existing classes, let the LyX developers list (lyx-devel@lists.lyx.org) know and we will arrange to insert a new entry, or teach you how to do it.\(^\text{10}\) Moreover, if you use a literate tool other than Noweb you may need to create a new set of document classes for it.

Typing code in  LyX enables you to write code with a layout named Scrap.\(^\text{11}\) Noweb delimits scraps like this:

```plaintext
<<My scrap>>=
code
more code
even more code
@
```

The problem is that whatever is written in between the `<<` and the `@` must be taken literally, i.e. LyX should be prevented from making any special interpretation of what has been written. This is handled by a special layout named Scrap, that works like a normal paragraph but has a free spacing capability.

The down side of the Scrap paragraph layout is that consecutive paragraphs of code will be spaced with one empty line in the source code and also in the printed documentation. The workaround is to enter each line of code within a single Scrap, with a newline (ctrl-return). The example above will look like this:\(^\text{12}\)

\(^{10}\) It is very simple, it involves the creation of a file with four lines, and re-running of the auto configuration.

\(^{11}\) The equivalent Noweb term is “Chunk”. For historical reasons, I got used to the term “scrap” introduced by other literate tool named Nuweb, which I used for many years before rendering myself to Noweb.

\(^{12}\) If you have a printed version of this document you will not see any difference between the previous example and this one.
This layout works fine. The only real inconvenience is that you have to type ctrl-return instead of a plain return.\footnote{It is in my list of “improvements” to fix that.}

As a special note, you can also use the “%def” construct of Noweb in your scraps to add items to Noweb’s identifier cross-reference:

<<My scrap>>=
def some_function(args):
    "This is the doc string for this function."
    print "My args: ", args
@ %def some_function

For an example of this usage and the resulting cross-reference output, look at the Literate python program in LIBDIR/examples/listerrors.lyx which should make this all clear.

Generating the documentation  At this point you already have a new document file with a proper document class, and with some code and text on it. How do I print it? The answer is simple, you select View\&DVI, etc. Just like you would do for a plain document. No special procedure is required.

To help orientate you, I will now explain what happens inside \LaTeX:\

1. When the Update\&DVI menu option is chosen, a \LaTeX{} file is generated.

   If the document is of any literate class the generated file will be named with an extension name defined by the “literate” format (defined in the Preferences panel), otherwise the file will have the usual .tex extension.

2. Note that the only difference so far is in the name of the file, no special processing is required by \LaTeX{}. Given that you formatted the code using the Scrap layout that, by itself, takes care of the business.

3. If the document is of any literate class \LaTeX{} will then use the internal \LaTeX{} to Noweb converter, followed by the Noweb to \LaTeX{} converter\footnote{The converters are defined in the Tools\&Preferences panel, under the “Conversion” tab. See section Converters of the Customization manual for general information about converters.} to generate the \LaTeX{} file.

   Otherwise it will just skip this step.
4. Finally, \LaTeX{} is invoked and the regular post processing continues as in a plain document.

Independence from a particular “literate tool” is easily achieved by changing the commands that are run by the various converters.

**Generating the code** When the build menu option is chosen or the corresponding button in the toolbar is pressed, a \LaTeX{} file is generated just like step 1 above. Next, LyX invokes the Noweb\textrightarrow{}Program converter. This converter needs to be defined by the user and is not installed by default, though the Program format is. This converter (like any other converter) will have two parts:

1. The converter program itself. This program performs the conversion from the one format to the other (in this case, from the Noweb format to the Program pseudo-format).

2. The error log parser. This is a program whose sole purpose is to rewrite error messages in a format that LyX understands. This makes it possible for LyX to place error boxes in the right places in the file buffer.

The first part, the “Converter” setting, should be set to “build-script $$i$”. This basically means that LyX will call “build-script” (a program or script) with the name of the Noweb file (normally a file in the LyX temp directory).

The converter code looks in `MYLYXDIR/scripts` first, then in `LIBDIR/scripts` then on the path for the “listerrors” program.

The next part of the converter setting is the “Flags” which is to be set to “parselog=listerrors”. This will run any errors that are generated by the “build-script” process through the “listerrors” program.

The build will normally take place in LyX’s temporary directory, so the files produced by the conversion will be in that directory. LyX will copy out what it regards as the ‘main’ file, but the Noweb\textrightarrow{}Program conversion may produce several files, and so most of these would then be deleted when LyX was closed. The present solution is to use a ‘copier’, in this case, the `ext_copy.py` script in its default mode, so that the entire contents of the temporary directory is copied. More will get copied than is needed, to be sure, but nothing will be lost. If, however, you know what extensions the generated files will have, this can be improved by using the `-e` option to `ext_copy`. This option takes a comma-separated list of extensions to copy. So,

15See section *Copiers* of the *Customization* manual for information on these.
for example, if the conversion will generate only files with the extensions .c and .h, then the correct definition would be:

```bash
python -tt $$s/scripts/ext_copy.py -e c,h $$i $$o
```

The result will be that only files with these two extensions will be copied out.

**Build instructions in the document**  The last piece of the integration between LYX and noweb is the “build-script” scrap. Generally, the instructions for building your program should be embedded in a scrap of its own. The noweb-specific “build-script” above uses the notangle command to look for this scrap (called “build-script”) and runs its contents through “sh”.

Typically, such a scrap would look something like this:

```bash
<<build-script>>=
#!/bin/sh
if [ -z "${NOWEB_SOURCE}" ]
then
    NOWEB_SOURCE=myfile.nw
fi
[... code to extract files ...]
[... code to compile files ...]
@
```

Look in `LIBDIR/examples/listerrors.lyx` or in `LIBDIR/examples/Literate.lyx` which implement two versions of the “listerrors” program for some illustrations of how all of these pieces go together or in `LIBDIR/examples/noweb2lyx.lyx`. Interestingly, these three files show off the language-indepence of the LYX literate programming support since they are written in Python, C and Perl respectively.

### 6.3.3.2 Configuring LYX

All the Literate Programming support is configured by the **Tools**: **Preferences** panel in the “Conversion” tab. The important parts are:

**the “literate” format**  Set up via the Formats tab, this is where the Noweb-specific pieces are set up. The **GUI Name** is set to **Noweb**, the file extension is set to .nw. This tells LYX to create a file with a .nw extension in the first step of the conversion process.

**the Program format**  This is an empty format whose sole purpose is to be the endpoint of a conversion (which then allows us to set up a converter for it).

**Noweb->\LaTeX**  This converter performs the “weaving” of the literate document. For Noweb, it is set to “noweave -delay -index $$i > $$o”
6.3 Literate Programming

**NoWeb->Program** This performs the “tangling step”. As stated above, the Converter is set to “build-script $$i$$”, with Flags set to “originaldir,parselog=listerrors”.

6.3.3.3 Debug extensions

There is also a new function implemented in the LyX server, the “server-goto-file-row” function, to be used with ddd/gdb or other debugger.

When debugging code with ddd/gdb, it is possible to invoke a text editor at the current execution position with a single key stroke. The default ddd configuration for that is shift-ctrl-V. It happens that you can define the editor command line invocation in ddd by accessing the **Edit>Preferences>Helpers** dialog and changing the ’Edit Sources’ entry.

I take advantage of the new created LyX server function and this ddd feature, and set “Edit Sources” to:

```
echo "LYXCMD:monitor:server-goto-file-row:@FILE@ @LINE@" > ~/.lyxpipe.in
```

With this, whenever you are using ddd and find a point in the program that you want to edit, you just press shift-ctrl-V (in the ddd window), and ddd you forward this information to LyX through the LyX server and then the LyX window will show the same file with the cursor at the same position ddd was pointing to. No more guessing or long scrolling to locate a point in the program back from debugging!

Note however that you must enable the LyX server to get this feature working (it is disabled by default). You can enable it in **Preferences** (tabs Inputs, Paths) by entering in the LyXserver pipe a path like “/home/<your-home-directory>/lyx/lyxpipe”.

Read the LyX server documentation in the **Customization Manual** for further information.

6.3.3.4 Toolbar extensions

There are six new buttons that can be added to your LyX toolbar. Five of these buttons are short cuts to layout styles: **Standard**, **Section**, **LaTeX**, **LyX-Code**, and **Scrap**. The last one is a short cut to the “Build Program” File menu entry.

LyX has a range of buttons that are available for tool bar customization. In my toolbar I like to combine the six short cuts above with two more: One for **View>Update>DVI** and the other for **View>DVI** File menu entries. Here is how it looks like:

```
ToolBar
Layouts
Icon "layout Standard"
Icon "layout Section"
Icon "layout LaTeX"
Icon "layout LyX-Code"
Icon "layout Scrap"
```
6 \LaTeX\ Features needing Extra Software

6.3.3.5 Colors customization

There are a number of colors in LyX that can be customized in Preferences. One of the things that bothers people is the \LaTeX\ font color. The default color is red, since the scraps uses \LaTeX\ font, and there is a lot of scraps in literate documents, you may get tired of seeing everything in red. You can change it by going to the tabs Look&Feel, Colors.

The next thing is the visible presence of the newline character in the screen. You can choose the color of this particular character and make it blend in the background. I recommend you choosing a color that is close to the background but not equal, that way you still can see it is there, but it is not bothering you anymore.
7 Secrets of the \LaTeX\ Masters

Though LyX is a powerful tool, it cannot hope to support everything that can be done with pure TeX/\LaTeX. However, many familiar dirty TeX and \LaTeX tricks can be done within LyX, as long as you are not afraid to use that “TeX” button on the toolbar or add things to the \LaTeX preamble. This section lists some tips, tricks, and otherwise cool ideas to give your document that extra little flair. \textit{Do try this at home}, just start with something a little smaller and less important than your dissertation!

Most ideas in this section require less common files in your \LaTeX installation. If you have a system like \TeX, most will already be available. A few, however, will need to be downloaded from one of the CTAN archives. Often, there are several ways to do something, or several \LaTeX style files which do the same thing. We do not endorse one choice over another, we simply claim that we have done a particular task with a particular file. Put on your wizard hat, keep an eye out for dragons, and let us begin.

7.1 Multiple Columns

by \textsc{Lars Gullik Bjønnes}

7.1.1 Purpose

The aim for this chapter\footnote{Editor’s note: Lars’ original chapter was a masterful description of how to use the \texttt{multicol} package. However, it was too long to flow smoothly in this document. I have therefore chosen to excerpt the most important sections here (sorry, Lars); you can read the original chapter (and more of the story!) in the example file \texttt{examples/multicol.lyx}. — mer} is to show how the \LaTeX\ package \texttt{multicol} can be used in a LyX document. As LyX doesn’t support the \texttt{multicol} package natively yet, we have to use some small hacks. By reading this section it should be obvious how to do this.

7.1.2 Limitations

The \texttt{multicol} package allows switching between one and multicolumn format on the same page. Footnotes are handled correctly (for the most part), but will be placed at the bottom of the page and not under each column. \LaTeX’s float mechanism, however, is partly disabled in the current implementation. At the moment only page-wide floats can be used within the scope of the environment.
7.1.3 Examples

7.1.3.1 Two columns

If you want to have two columns in your text, you have use \LaTeX{} mode to insert \begin{multicols}{2} at the point where you want the two column layout to start, and then \end{multicols} where you want it to end. Like this:

The Adventure of the Empty House
by Sir Arthur Conan Doyle

It was in the spring of the year 1894 that all London was interested, and the fashionable world dismayed, by the murder of the Honourable Ronald Adair under most unusual and inexplicable circumstances. The public has already learned those particulars of the crime which came out in the police investigation, but a good deal was suppressed upon that occasion, since the case for the prosecution was so overwhelmingly strong that it was not necessary to bring forward all the facts. Only now, at the end of nearly ten years, am I allowed to supply those missing links which make up the whole of that remarkable chain. The crime was of interest in itself, but that interest was as nothing to me compared to the inconceivable sequel, which afforded me the greatest shock and surprise of any event in my adventurous life. Even now, after this long interval, I find myself thrilling as I think of it, and feeling once more that sudden flood of joy, amazement, and incredulity which utterly submerged my mind. Let me say to that public, which has shown some interest in those glimpses which I have occasionally given them of the thoughts and actions of a very remarkable man, that they are not to blame me if I have not shared my knowledge with them, for I should have considered it my first duty to do so, had I not been barred by a positive prohibition from his own lips, which was only withdrawn upon the third of last month.

7.1.3.2 Multiple columns

The same pattern is used when you want more than two columns:

You can have more than 3 columns if you want to, but that might not be very pleasant for the eye.
7.1.3.3 Columns inside columns

You can even have columns inside columns:

The Honourable Ronald Adair was the second son of the Earl of Maynooth, at that time governor of one of the Australian colonies. Adair’s mother had returned from Australia to undergo the operation for cataract, and she, her son Ronald, and her daughter Hilda were living together at 427 Park Lane.

The youth moved in the best society—had, so far as was known, no enemies and no particular vices. He had been engaged to Miss Edith Woodley, of Carstairs, but the engagement had been broken off by mutual consent some months before, and there was no sign that it had left any very profound feeling behind it. For the rest [sic] the man’s life moved in a narrow and conventional circle, for his habits were quiet and his nature unemotional. Yet it was upon this easy-going young aristocrat that death came, in most strange and unexpected form, between the hours of ten and eleven-twenty on the night of March 30, 1894.

Ronald Adair was fond of cards—playing continually, but never for such stakes as would hurt him. He was a member of the Baldwin, the Cavendish, and the Bagatelle card clubs. It was shown that, after dinner on the day of his death, he had played a rubber of whist at the latter club. He had also played there in the afternoon. The evidence of those who had played with him—Mr. Murray, Sir John Hardy, and Colonel Moran—showed that the game was whist, and that there was a fairly equal fall of the cards. Adair might have lost five pounds, but not more. His fortune was a considerable one, and such a loss could not in any way affect him. He had played nearly every day at one club or other, but he was a cautious player, and usually rose a winner. It came out in evidence that, in partnership with Colonel Moran, he had actually won as much as four hundred and twenty pounds in a sitting, some weeks before, from Godfrey Milner and Lord Balmoral. So much for his recent history as it came out at the inquest.

Please do read the file examples/multicol.lyx for more advanced examples including column and header spacing, vertical separator lines, and more.

7.2 Numbering in the Enumerate Paragraph Environment

by John Weiss

The default numbering for the Enumerate paragraph environment begins with Arabic numbers and ends with uppercase letters. Suppose, however, you wanted a different type of numbering scheme. Here’s a quickie example of how to change the numbering scheme:

\renewcommand{\labelenumi}{\Roman{enumi}.}
\renewcommand{\labelenumii}{\Alph{enumii}.}
\renewcommand{\labelenumiii}{\arabic{enumiii}.}
\renewcommand{\labelenumiv}{\alph{enumiv}).}

... which changes the numbering scheme to uppercase Roman numerals, uppercase letters, Arabic numbers, and lowercase letter.
Additionally, the previous example also adds a little bit extra to the numbering scheme. For example, the first level label actually looks like: “I.”. For ease of reading, we’ll describe what the numbering schemes look like using a notation something like this: <“I”, “A.”, “1.”, “a.”>.

As you can see in the example, there is a label command for each nesting level, \labelenumi \ldots \labelenumiv, as well as a counter, enumi \ldots enumiv. There are also five “number printing” commands, \arabic{}, \roman{}, \Roman{}, \alph{}, \Alph{}, each of which take one counter as an argument. You can add characters before or after these, but there’s no need to add spaces.

You can get really fancy with these. For example:

```
\renewcommand{\labelenumi}{\#\Alph{enumi}\#}
\renewcommand{\labelenumii}{\Alph{enumi}.\arabic{enumii}}
\renewcommand{\labelenumiii}{\alph{enumiii}+}
\renewcommand{\labelenumiv}{(\roman{enumiv})}
```

produces the somewhat out of hand numbering scheme: <“#A#”, ”A.1”, ”a+”, “(i)”>.

### 7.3 Dropped Capitals

by Mike Ressler

Those of you who like the style of old books probably also like “dropped capitals”—those large capital letters which begin each new chapter or section. Implementing them with plain \LaTeX is straightforward (assuming you know some plain \TeX!) but does require a lot of work and many iterations, as you can see by all the ugly \TeX-mode stuff at the beginning of this paragraph.

```
\bigdrop{-1em}{3}{ptmri}{T}here is a much easier way of doing this, of course.
```

The dropcaps (or the newer dropping) package from CTAN allows a simple way to add such letters to your documents. Since this package is not a standard part of \TeX, I can’t demonstrate it within this document, but if you copy this paragraph to a new document, delete the “\verb” and the pluses from the \TeX code at the beginning of the paragraph, and add \usepackage{dropcaps} to your \LaTeX preamble, you will get a nice Times Roman Italic “T”, whose height is three lines of text and which protrudes 1 em into the margin. (Make certain you have copied “dropcaps.sty” into a directory where \TeX can see it.) The first argument is the amount of indentation; in this case the negative sign moves it into the margin. The second argument is the height of the letter in number of lines of text. The third argument is the font name: virtually anything which has a tfm file should work (wade through the .../texmf/fonts/tfm directory for possibilities). My personal favorite is “yinit”, a fancy German font specifically designed for dropped capitals. The fourth argument is the letter (or letters) to be dropped. The \dropping package also offers the \bigdrop command, as well as a slightly simplified \dropping command.
7.4 Non-standard Paragraph Shapes

by Mike Ressler

There are times when the tyranny of rectangular paragraphs must be overthrown. In such situations, a call to the delightful plain \TeX command \texttt{\parshape} is called for. As you can see, completely arbitrary shapes can be laid out with a suitable set of linelength definitions. While this parshape may look a bit silly and useless, one could conceive of situations such as finely tuned dropped capitals, word wrapping around non-rectangular graphics, etc. which will benefit from such handcrafting.

The syntax is \texttt{\parshape numlines \#1indent \#1length \#2indent \#2length \ldots \#nindent \#nlength}, where \texttt{numlines} is the number of lines of text which define the paragraph. If there turn out to be fewer lines, the shape is truncated; if there are more, the excess lines have the same dimensions as the last line of the definition. The \texttt{\#indent} and \texttt{\#length} entries specify the indentation of the line from the left margin, and the length of the line as measured from that point. The shape applies only to the current paragraph; everything is reset to normal for the next paragraph.

7.5 Summary

As you can see, the examples in this section range from the useful to the whimsical. While I don’t expect that anyone will ever need the paragraph shape demonstrated in the last section, the important point is that you can do almost anything you want
7 Secrets of the \LaTeX Masters

in \LaTeX if you are willing to figure out how to do it in \TeX and \LaTeX. \TeX is a fantastically powerful typesetting system and all that power is available to you since \LaTeX uses it as its backend. Happy \LaTeXing!
LyX’s detailed Figure, Table, Floats, Notes, Boxes and External Material manual

by the LyX Team*

Version 1.6.x

August 17, 2009

*If you have comments or error corrections, please send them to the LyX Documentation mailing list: lyx-docs@lists.lyx.org
Contents

2.12.3. Customized Cell/Column Format ................................................. 310
2.12.4. Line Thickness ................................................................. 312
2.12.5. Dashed Lines ................................................................. 313

3. Floats ......................................................................................... 315
3.1. Introduction ........................................................................... 315
3.2. Float Types ........................................................................... 315
  3.2.1. Algorithm Floats ............................................................... 315
  3.2.2. Wrap Floats ................................................................. 316
3.3. Float Numbering ..................................................................... 317
3.4. Referencing Floats ................................................................. 318
  3.4.1. Cross-Reference Formats ............................................... 318
  3.4.2. Automatic Reference Naming ........................................... 319
  3.4.3. Reference Position ......................................................... 320
3.5. Float Placement ................................................................. 320
3.6. Rotated Floats ........................................................................ 322
3.7. Subfloats ............................................................................... 322
3.8. Floats Side by Side ............................................................... 322
3.9. Caption Formatting ............................................................... 325
3.10. Caption Placement ............................................................... 326
3.11. Listings of Floats ................................................................. 327

4. Notes ......................................................................................... 329
4.1. LyX Notes ............................................................................... 329
4.2. Footnotes ............................................................................... 330
  4.2.1. Footnote Numbering ....................................................... 331
  4.2.2. Footnote Placement ....................................................... 332
4.3. Margin Notes ......................................................................... 334

5. Boxes ......................................................................................... 337
5.1. Introduction ........................................................................... 337
5.2. Box Dialog ........................................................................... 337
  5.2.1. Size ................................................................................. 337
  5.2.2. Alignment ....................................................................... 338
  5.2.3. Decoration ....................................................................... 339
5.3. Box customization ................................................................. 340
5.4. Minipages ............................................................................... 342
5.5. Parboxes ............................................................................... 344
5.6. Boxes for Words and Characters ............................................ 344
  5.6.1. Prevent Hyphenation ....................................................... 344
  5.6.2. Vertical Alignment .......................................................... 344
5.7. Colored Boxes ........................................................................ 345
  5.7.1. Color for Text .................................................................. 345
  5.7.2. Color for Paragraphs ....................................................... 346
## Contents

5.8. Rotated and Scaled Boxes ........................................ 347
  5.8.1. Rotated Boxes ........................................ 347
  5.8.2. Scaled Boxes ........................................ 348

6. External Document Parts ........................................ 351
  6.1. External Material ........................................ 351
  6.2. Child Documents ........................................ 352
    6.2.1. External Subsection 1 ............................... 354
    6.2.2. External Subsection 2 ............................... 355

7. Program Code Listings ........................................ 357

A. Units available in LyX ........................................ 359

B. Output File Formats with Graphics ............................ 361
    B.1. DVI ................................................ 361
    B.2. PostScript .......................................... 361
    B.3. PDF ............................................... 362

C. Explanation of Equation (2.1) ................................ 363

Bibliography ...................................................... 365

Index ............................................................. 367

List of Figures .................................................. 369

List of Tables .................................................. 371

List of Algorithms ............................................ 373
1. Figures

1.1. Graphics Dialog

To insert an image into your document, place the cursor at the text position you want and click on the toolbar icon or use the menu Insert ➤ Graphics. Then a dialog will appear to choose the file to load. The image will appear in the output exactly at the position where it is in the text.

The graphics dialog can be called at any time by clicking on an image. This dialog has three tabs:

- **Graphics** Here you can choose an image file and adjust its appearance in the output. The available units for the image size are explained in appendix A. You can rotate images counter-clockwise by setting a rotation angle and a rotation origin. The image will also be rotated inside LyX. Images can be scaled by using a percentage value or by setting the width and height explicitly. If you set only the width or only the height, the other size will be determined automatically. If you set both, then the image will be transformed to the given size, possibly distorting it. To prevent the image from distortion, use the option Maintain aspect ratio. The image will then be scaled so that its width and height don’t exceed the specified dimensions. Images can be opened in a program of your choice by right-clicking on it choosing the entry Edit externally in the appearing context menu. The program can be set for every image format in the file format settings in LyX’s preferences.

- **Clipping** Alternatively to the usage of scaling units it is possible to set image coordinates to adjust the height and width of the image in the output. The coordinates can also be calculated automatically by pressing the button Get from File. The option Clip to bounding box will only print the image region within the given coordinates. Normally you don’t need to take care about image coordinates and can ignore this tab.

- **\LaTeX and \textsf{LyX} options** In this tab you can modify the appearance of the image within \textsf{LyX} and \LaTeX experts can specify additional \LaTeX options. The option Draft mode makes the image appear in the output only as a frame with the size of the image. The Don’t unzip on export option only affects zipped EPS-graphics, e.g. \texttt{x.eps.gz}. When the option is used the images will not be unzipped on export, since \LaTeX
can handle them as they are.
Zipped EPS-graphics are useful to save disk space when you choose PostScript as output format, see appendix B.2. To zip EPS-graphics, use the following commands in a UNIX-shell or a Windows console:

\texttt{gzip x.eps}
\texttt{zgrep \textasciicircum\textasciicircum Bounding x.eps.gz > x.eps.bb}

The second command creates the bounding box file “x.eps.bb” that is needed by \LaTeX{} for zipped graphics.
The field \textbf{Initialize Group Name} allows you define or join an image settings group. Images within such a group share their settings, so adjusting one image of the group automatically also adjusts all other images of the group in the same way. So you can for example change the size for a bunch of images without the need to manually change each of them. Joining an existing group can also be done using the context menu of the image by checking the name of the desired group.

This is an example image in EPS format\textsuperscript{1} within a separate, horizontally centered paragraph:

\begin{center}
\includegraphics{example_image.eps}
\end{center}

This is the same image like the one above but in draft mode:

\begin{center}
\includegraphics{example_image_draft.eps}
\end{center}

\textsuperscript{1}Image formats are explained in section 1.3.
1. Figures

1.2. Figure Floats

For general explanations about floats, have a look at section 3.1.

The toolbar button and the menu \texttt{Insert} \texttt{Float} \texttt{Figure} inserts a float with a caption that has the label “Figure #:” (\# is the actual number). You can insert the image above the caption, like in Figure 1.1 or below the caption, like in Figure 1.2. More about the caption placement is described in section 3.10.

Figure 1.1 and 1.2 are examples of referenced figures. Figures can be referenced in the text by referencing their label. To do this insert a label in the caption using the menu \texttt{Insert} \texttt{Label} or the toolbar button . You can now refer to the label using the menu \texttt{Insert} \texttt{Cross reference} or the toolbar button . It is important to use references to floats, rather than using vague references like “the figure above”, because as \LaTeX will reposition the floats in the final document, it might not be “above” at all. Referencing is explained in detail in section 3.4.

Normally only one image is inserted to a figure float, but sometimes you might want to use two images with separate subcaptions. This can be done by inserting image floats into existing image floats. Note that only the main caption of the float is added to the List of Figures. Figure 1.3 is an example of a figure float with two images set side by side. You can also set the images one below the other. Figure 1.3a and 1.3b are the subfigures.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{platypus_float}
\caption{A severely distorted platypus in a float.}
\end{figure}

Figure 1.2.: M.C. Escher on acid.

Figure 1.3.: Two distorted images. Both images are in the image settings group named “distorted”.

(a) Undefinable structure.
(b) A Platypus.
1. Figures

1.3. Image Formats

You can insert images in any known file format. But as explained in appendix B, every output document format allows only a few image formats. **\LaTeX** uses therefore the program ImageMagick in the background to convert the images to the right format. To increase your work flow by avoiding these conversions in the background, you can use only the image formats that can directly be embedded in the output file format. The output file formats are explained in appendix B.

Similar to fonts there are two types of image formats:

**Bitmap images** consist of pixel values, often in a compressed form. They are therefore not fully scalable and look pixeled in large zooms. Well-known bitmap image formats are “Graphics Interchange Format” (GIF, file extension “.gif”), “Portable Network Graphics” (PNG, file extension “.png”), and “Joint Photographic Experts Group” (JPG, file extension “.jpg” or “.jpeg”).

**Vector images** consist of vectors and can therefore be scaled to any size without data loss. The scaling ability is necessary if you want to create presentations, because presentations are always scaled by the video projector. Scaling is also useful for online documents to let the user zoom into diagrams. Well-known scalable image formats are “Scalable Vector Graphics” (SVG, file extension “.svg”), “Encapsulated PostScript” (EPS, file extension “.eps”), “Portable Document Format” (PDF, file extension “.pdf”), and “Windows Metafile” (WMF, file extension “.wmf”). We wrote “can be”, because you can convert any bitmap image to a PDF or EPS-image and the result will still be a bitmap image. In this cases only a header with the image properties is added to the original image\(^2\). The PDF-files generated by **Adobe Photoshop** are for example bitmap images.

Normally it is not possible to convert a bitmap image into a scalable one, only vice versa. Only the image formats PDF and EPS can directly be embedded to PDF and PostScript output files, respectively. SVG and WMF-images are recalculated to bitmaps when the output file is generated because there is currently no adequate WMF/SVG→PDF/EPS converter available.

\(^2\)In the case of PDF, the original image is additionally compressed.
2. Tables

2.1. Introduction

You can insert a table using either the toolbar button or the menu Insert > Table. The toolbar button offers you a graphical selection: Move the mouse to set the column/row number of the table that should be created and then press a mouse button. When you use the menu to create a table, a dialog will appear, asking you for the number of rows and columns.

The default table has lines around any cell and the first row appears separated from the rest of the table. This separation occurs due to a double line: The cells of the first row have a line below them and the cells of the second row have a line above them. Here is an example table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.2. Table Dialog

You can alter a table by clicking on it with the right mouse button, which brings up the table dialog. Here you can adjust the settings of that cell and row/column respectively where the cursor is currently placed. Most of the dialog options also work on selections. This means if you select more cells, columns or rows, the action is done for the whole selection. Note that there is a difference between selecting the contents of the cell, and the cell itself. You can alter tables with the following tabs of the table dialog:

Table Settings Here you can set the horizontal alignment and the width of the current column. When you have set a width you can also adjust the vertical alignment of the current row. A given width will allow the cell to have line breaks and multiple paragraphs of text, see section 2.8.1. If you set no width, the column is as wide as their widest cell content is.

Furthermore, you can mark one or multiple cells of one row as a multicolumn
2. Tables

cell, see section 2.8.2.
The rotate check boxes rotates the current cell, a selection, or the whole table counter-clockwise by 90°. The rotation is not shown within LYX, only in the output.

Note: Not all DVI-viewers are able to display rotations.
It is also possible to enter a \LaTeX-argument which is needed for special table formattings, see section 2.8.2.2 and 2.11.

Borders In this tab you can add and delete border lines for the current row/column.
Using the style option Formal will convert the table to a formal table as described in section 2.9.
You can also add here space to table rows as described in section 2.12.1.

Longtable This tab is to make a table a so called “longtable” that can run over several pages. Section 2.6 and 2.7 describe the longtable features in detail.

When the table dialog is open, you can move the cursor with the arrow keys from cell to cell and the property of the current cell will immediately be displayed in the dialog.

2.3. Table Toolbar

The table toolbar is an alternative to the table dialog to be able to alter tables faster. It should normally appear at the bottom of LYX’s main window when the cursor is inside a table. You can alternatively switch it on to appear always, by right-clicking in LYX’s main menu bar.

The toolbar has the following icons:

- \includegraphics[width=0.1\textwidth]{icon-add-row.png}
  adds a row below the current cell or selection

- \includegraphics[width=0.1\textwidth]{icon-add-column.png}
  adds a column right beside the current cell or selection

- \includegraphics[width=0.1\textwidth]{icon-delete-row.png}
  deletes the current row or selection

- \includegraphics[width=0.1\textwidth]{icon-delete-column.png}
  deletes the current column or selection

- \includegraphics[width=0.1\textwidth]{icon-add-top-line.png}
  adds a line at the top of the current cell / row or of a selection

- \includegraphics[width=0.1\textwidth]{icon-add-bottom-line.png}
  adds a line at the bottom of the current cell / row or of a selection

- \includegraphics[width=0.1\textwidth]{icon-add-left-line.png}
  adds a line at the left side of the current cell / row or of a selection

- \includegraphics[width=0.1\textwidth]{icon-add-right-line.png}
  adds a line at the right side of the current cell / row or of a selection
2.4. Edit Table Menu

- adds lines around the current or selected cells - if the current cell no multicolumn this also affects the current row and column
- deletes all lines of the current or selected cells - if the current cell no multicolumn this also affects the current row and column
- left-aligns the content of the current cell / column
- centers the content of the current cell / column horizontally
- right-aligns the content of the current cell / column
- aligns the content of the current cell vertically to the top
- centers the content of the current cell vertically
- aligns the content of the current cell vertically to the bottom
- rotates the current cell or selection counter-clockwise by 90°
- rotates the whole table counter-clockwise by 90°
- sets the current cell or selection as a multicolumn

**Note:** For the output the vertical alignment of the first cell in a row is used for all following cells in the row.

2.4. Edit Table Menu

Additionally to the table dialog and toolbar, the menu Edit > Table allows you to add and delete border lines for the current row/column and to set the current selection as multicolumn. The menu is only available when the cursor is inside a table.

2.5. Table Floats

For general explanations about floats, have a look at section 3.1.

Table floats can be inserted using the menu Insert > Float > Table or the toolbar button

The float appears as a collapsible box with a caption that has the label “Table #:” (# is the actual table number). You can insert tables to the float above or below the caption.
2. Tables

Table 2.1.: A table float.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mary</td>
<td></td>
<td>2</td>
<td>Mary</td>
</tr>
<tr>
<td>Ted</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>∫x²dx</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[a b]</td>
<td>a</td>
<td>b</td>
<td>a</td>
</tr>
<tr>
<td>[c d]</td>
<td>c</td>
<td>d</td>
<td>c</td>
</tr>
<tr>
<td>1 + 1</td>
<td>1</td>
<td>1</td>
<td>1 + 1</td>
</tr>
</tbody>
</table>

Table 2.1 is an example table within a table float.

Having the caption above the table is the common rule that is unfortunately not supported in \LaTeX’s standard classes. That means if you are using the document classes article, book, letter, or report there will be no space between the caption and the table. To insert the needed space, add the following option to the load command of the \LaTeX-package caption in your document preamble:\footnote{For more information have a look at section 3.10.}

tableposition=top

The package caption, which is described in section 3.9, is used to adjust the caption format.

Tables can be cross-referenced in the text by referencing their label. To do this insert a label in the caption using the menu Insert\ Label or the toolbar button \Label. You can now refer to the label using the menu Insert\ Cross reference or the toolbar button \Cross reference.

Referencing is explained in detail in section 3.4.

2.6. Longtables

If the table is too long to fit on one page, you can use the option Use long table in the tab Longtable of the table dialog to split the table automatically over more pages. Doing this enables the following options:

Header: The current row is defined to be a header row that appears on all pages of the longtable; except for the first page, if First header is defined. This therefore called the main header.

First header: The current row is defined to be a header row that appears on the first page of the longtable.

Footer: The current row is defined to be a footer row that appears on all pages of the longtable; except for the last page, if Last footer is defined.

Last footer: The current row is defined to be a footer row that appear on the last page of the longtable.
## 2.6. Longtables

**Caption:** The current row contains the table caption. The row is reset as single column and a caption is inserted. More about longtable captions is explained in sec. 2.6.3.

You can also specify a row where the table is split. See the following longtable to see how it works:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annovi</td>
<td>Silvia 111</td>
</tr>
<tr>
<td>Bertoli</td>
<td>Stefano 111</td>
</tr>
<tr>
<td>Bozzi</td>
<td>Walter 111</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maria 111</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maurizio 111</td>
</tr>
<tr>
<td>Cinquemani</td>
<td>Giusi 111</td>
</tr>
<tr>
<td>Colin</td>
<td>Bernard 111</td>
</tr>
<tr>
<td>Concli</td>
<td>Gianfranco 111</td>
</tr>
<tr>
<td>Dal Bosco</td>
<td>Carolina 111</td>
</tr>
<tr>
<td>Dalpiaz</td>
<td>Annamaria 111</td>
</tr>
<tr>
<td>Feliciello</td>
<td>Domenico 111</td>
</tr>
<tr>
<td>Focarelli</td>
<td>Paola 111</td>
</tr>
<tr>
<td>Galletti</td>
<td>Oreste 111</td>
</tr>
<tr>
<td>Gasparini</td>
<td>Franca 111</td>
</tr>
<tr>
<td>Rizzardi</td>
<td>Paola 111</td>
</tr>
<tr>
<td>Lassini</td>
<td>Giancarlo 111</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Luciano 111</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Valeriano 111</td>
</tr>
<tr>
<td>Meneguzzo</td>
<td>Roberto 111</td>
</tr>
<tr>
<td>Mezzadra</td>
<td>Roberto 111</td>
</tr>
<tr>
<td>Pirpamer</td>
<td>Erich 111</td>
</tr>
<tr>
<td>Pochiesa</td>
<td>Paolo 111, 222</td>
</tr>
<tr>
<td>Radina</td>
<td>Claudio 111</td>
</tr>
<tr>
<td>Stuffer</td>
<td>Oskar 111</td>
</tr>
<tr>
<td>Taccchelli</td>
<td>Ugo 111</td>
</tr>
<tr>
<td>Tezzele</td>
<td>Margit 111</td>
</tr>
<tr>
<td>Unterkalmsteiner</td>
<td>Frieda 111</td>
</tr>
<tr>
<td>Vieider</td>
<td>Hilde 111</td>
</tr>
<tr>
<td>Vigna</td>
<td>Jürgen 111</td>
</tr>
<tr>
<td>Weber</td>
<td>Maurizio 111</td>
</tr>
</tbody>
</table>

continued on next page
<table>
<thead>
<tr>
<th>NAME</th>
<th>TEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winkler</td>
<td>Franz</td>
</tr>
<tr>
<td>Annovi</td>
<td>Silvia</td>
</tr>
<tr>
<td>Bertoli</td>
<td>Stefano</td>
</tr>
<tr>
<td>Bozzi</td>
<td>Walter</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maria</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maurizio</td>
</tr>
<tr>
<td>Cinquemani</td>
<td>Giusi</td>
</tr>
<tr>
<td>Colin</td>
<td>Bernard</td>
</tr>
<tr>
<td>Concli</td>
<td>Gianfranco</td>
</tr>
<tr>
<td>Dal Bosco</td>
<td>Carolina</td>
</tr>
<tr>
<td>Dalpiaz</td>
<td>Annamaria</td>
</tr>
<tr>
<td>Feliciello</td>
<td>Domenico</td>
</tr>
<tr>
<td>Focarelli</td>
<td>Paola</td>
</tr>
<tr>
<td>Galletti</td>
<td>Oreste</td>
</tr>
<tr>
<td>Gasparini</td>
<td>Franca</td>
</tr>
<tr>
<td>Rizzardi</td>
<td>Paola</td>
</tr>
<tr>
<td>Lassini</td>
<td>Giancarlo</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Luciano</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Valeriano</td>
</tr>
<tr>
<td>Meneguzzo</td>
<td>Roberto</td>
</tr>
<tr>
<td>Mezzadra</td>
<td>Roberto</td>
</tr>
<tr>
<td>Pirpamer</td>
<td>Erich</td>
</tr>
<tr>
<td>Pochiesa</td>
<td>Paolo</td>
</tr>
<tr>
<td>Radina</td>
<td>Claudio</td>
</tr>
<tr>
<td>Stuffer</td>
<td>Oskar</td>
</tr>
<tr>
<td>Tacchelli</td>
<td>Ugo</td>
</tr>
<tr>
<td>Tezzele</td>
<td>Margit</td>
</tr>
<tr>
<td>Unterkalmsteiner</td>
<td>Frieda</td>
</tr>
<tr>
<td>Vieider</td>
<td>Hilde</td>
</tr>
<tr>
<td>Vigna</td>
<td>Jürgen</td>
</tr>
<tr>
<td>Weber</td>
<td>Maurizio</td>
</tr>
<tr>
<td>Winkler</td>
<td>Franz</td>
</tr>
</tbody>
</table>
2.6. Longtables

2.6.1. Footnotes in Longtables

Footnotes can be inserted to every longtable cell. They appear at the bottom of the page where the table cell with the footnote appears. Table 2.6 has for example a footnote.

2.6.2. Longtable Alignment

Longtables are by default centered. In contrary to the alignment of the table columns and row, the table alignment can currently not be changed in the table dialog. To change the alignment of longtables you have to change the value of the lengths \LTleft and \LTright by inserting this line as \TeX-Code before the corresponding longtable:

\setlength{\LTleft}{value}

Where the value can have any of the units listed in Table A.1. \LTleft controls the horizontal distance from the left page border to the longtable, \LTright the distance from the right side. The default value for both lengths is \fill, which is in this case the same as an horizontal fill in LyX.

The following longtable was left-aligned by setting \LTleft to 0 pt.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
</tr>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
</tr>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
</tr>
<tr>
<td>asd</td>
<td>asd</td>
<td>asd</td>
<td>asd</td>
<td>asd</td>
</tr>
</tbody>
</table>

2.6.3. Longtable Captions

A longtable cannot be put into a table float because floats can only be on one page. But the caption environment of floats can also be used for longtables when you use for a table row the longtable option Caption as described in sec. 2.6. Only one table row can contain the caption.

Here is a short longtable to see how it works:

Table 2.2.: Longtable with caption

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
</tr>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
</tr>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
</tr>
</tbody>
</table>

291
2. Tables

| asd | asd | asd | asd | asd |

**Note 1:** The table number is increased for every longtable, also if you didn’t set a caption for it. For this reason you could have the case that e.g. Table 2.4 follows on Table 2.1 in the list of tables if there are two longtables without captions. To avoid this you can add the following command in \TeX-Code behind every longtable without a caption:

\addtocounter{table}{-1}

This is not needed when none of your longtables have a caption and you add the following code to the document preamble:

```latex
\let\myEnd\endlongtable
\renewcommand{\endlongtable}{\myEnd\addtocounter{table}{-1}}
```

**Note 2:** If you are using \hyperref in the PDF Properties of the Document Settings dialog to link cross-references, the link to a longtable caption will always point to the beginning of the document.

2.6.3.1. References to Longtables

**Table 2.3.:** Referenced longtable

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
</tr>
<tr>
<td></td>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
</tr>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
<td></td>
</tr>
<tr>
<td>asd</td>
<td>sad</td>
<td>asd</td>
<td>asd</td>
<td>asd</td>
<td></td>
</tr>
</tbody>
</table>

To reference a longtable, insert a label into the caption.

This is a reference to Table 2.3.

The caption layout can be set together with all other captions of your document using the \LaTeX-package \caption, see section 3.9.

2.6.3.2. Caption Width

The maximal width of of caption lines is defined by the length \LTcapwidth. Its default value is 4in. To change it add the following command to your document preamble or as \TeX-Code into your document before the longtable that should be affected:

\setlength{\LTcapwidth}{width}
where the width could have one of the units listed in appendix A.

The following tables show the difference:

**Table 2.4.:** long full title with default width

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
<td></td>
</tr>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
<td></td>
</tr>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
<td></td>
</tr>
<tr>
<td>asd</td>
<td>sad</td>
<td>asd</td>
<td>asd</td>
<td>asd</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2.5.:** long full title with width set to 5 cm

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
<td></td>
</tr>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
<td></td>
</tr>
<tr>
<td>asd</td>
<td>s</td>
<td>s</td>
<td>s</td>
<td>asd</td>
<td></td>
</tr>
<tr>
<td>asd</td>
<td>sad</td>
<td>asd</td>
<td>asd</td>
<td>asd</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** When the \LaTeX-package caption is used, as in this document, the full page width is used for the caption when you use the default value of 4 in for \LTcapwidth. To get in this case exactly a 4 in wide caption, you can either use a value slightly different from 4.0 in, e.g. 3.99 in, or the \LaTeX-command \captionsetup{width=value} that is provided by the caption-package.

**2.6.3.3. Different Captions for Table Pages**

When the table captions for the following pages should differ from the one of the first table page, insert a caption with the \LaTeX code command

```
\caption*{caption text}\
```

in a dummy caption row that is marked as header. Table 2.6 is an example for a longtable with different heading where the second caption doesn’t include the table number.
Table 2.6.: Example Phone List

<table>
<thead>
<tr>
<th>NAME</th>
<th>TEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annovi</td>
<td>Silvia</td>
</tr>
<tr>
<td>Bertoli</td>
<td>Stefano</td>
</tr>
<tr>
<td>Bozzi</td>
<td>Walter</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maria</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maurizio</td>
</tr>
<tr>
<td>Cinquemani</td>
<td>Giusi</td>
</tr>
<tr>
<td>Colin</td>
<td>Bernard</td>
</tr>
<tr>
<td>Concli</td>
<td>Gianfranco</td>
</tr>
<tr>
<td>Dal Bosco</td>
<td>Carolina</td>
</tr>
<tr>
<td>Dalpiaz</td>
<td>Annamaria</td>
</tr>
<tr>
<td>Feliciello</td>
<td>Domenico</td>
</tr>
<tr>
<td>Focarelli</td>
<td>Paola</td>
</tr>
<tr>
<td>Galletti</td>
<td>Oreste</td>
</tr>
<tr>
<td>Gasparini</td>
<td>Franca</td>
</tr>
<tr>
<td>Rizzardi²</td>
<td>Paola</td>
</tr>
<tr>
<td>Lassini</td>
<td>Giancarlo</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Luciano</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Valeriano</td>
</tr>
<tr>
<td>Meneguzzo</td>
<td>Roberto</td>
</tr>
<tr>
<td>Mezzadra</td>
<td>Roberto</td>
</tr>
<tr>
<td>Pirpamer</td>
<td>Erich</td>
</tr>
<tr>
<td>Pochiesa</td>
<td>Paolo</td>
</tr>
<tr>
<td>Radina</td>
<td>Claudio</td>
</tr>
<tr>
<td>Stuffer</td>
<td>Oskar</td>
</tr>
<tr>
<td>Tacchelli</td>
<td>Ugo</td>
</tr>
<tr>
<td>Tezzele</td>
<td>Margit</td>
</tr>
<tr>
<td>Unterkalmsteiner</td>
<td>Frieda</td>
</tr>
<tr>
<td>Vieider</td>
<td>Hilde</td>
</tr>
<tr>
<td>Vigna</td>
<td>Jürgen</td>
</tr>
<tr>
<td>Weber</td>
<td>Maurizio</td>
</tr>
<tr>
<td>Winkler</td>
<td>Franz</td>
</tr>
</tbody>
</table>

²Example footnote

294
2.7. Special Longtable Issues

2.7.1. Longtable Calculation

\LaTeX\ calculates the height of table pages and their page breaks using so called chunks. Chunks are pieces of the tables that are at once in \LaTeX\’s memory. The default value is historically set to only 20 table rows. If you are using longtables with many pages this may slow down the creation of your document. You can safely increase the chunk size to values of 100-1000 by adding this command line to your document preamble:

\setcounter{LTchunksize}{100}

<table>
<thead>
<tr>
<th>NAME</th>
<th>TEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annovi</td>
<td>Silvia</td>
</tr>
<tr>
<td>Bertoli</td>
<td>Stefano</td>
</tr>
<tr>
<td>Bozzi</td>
<td>Walter</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maria</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maurizio</td>
</tr>
<tr>
<td>Cinquemani</td>
<td>Giusi</td>
</tr>
<tr>
<td>Colin</td>
<td>Bernard</td>
</tr>
<tr>
<td>Concli</td>
<td>Gianfranco</td>
</tr>
<tr>
<td>Dal Bosco</td>
<td>Carolina</td>
</tr>
<tr>
<td>Dalpiaz</td>
<td>Annamaria</td>
</tr>
<tr>
<td>Feliciello</td>
<td>Domenico</td>
</tr>
<tr>
<td>Focarelli</td>
<td>Paola</td>
</tr>
<tr>
<td>Galletti</td>
<td>Oreste</td>
</tr>
<tr>
<td>Gasparini</td>
<td>Franca</td>
</tr>
<tr>
<td>Rizzardi</td>
<td>Paola</td>
</tr>
<tr>
<td>Lassini</td>
<td>Giancarlo</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Luciano</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Valeriano</td>
</tr>
<tr>
<td>Meneguzzo</td>
<td>Roberto</td>
</tr>
<tr>
<td>Mezzadra</td>
<td>Roberto</td>
</tr>
</tbody>
</table>
2. Tables

2.7.2. Floats and Longtables

There might be problems when a float appears on the same page where a longtable starts. To avoid such situation, add the command `\clearpage` as \TeX-Code before your longtable.

2.7.3. Forced Page Breaks

By default tables are only broken between rows. If you have a cell with multiples lines and want to have a page break within the cell, insert the new line command “\” as \TeX-Code at this point of the cell where it can be broken. Before the “\” command you have to insert in \TeX-Code so many “&” characters like the number of the following table columns. The & is the character to separate table cells. Write in \TeX-Code after each & the content of the corresponding following cell and delete the content of these cells.

Behind the the “\” command, insert so many & characters like the number of table columns before the current column. In Table 2.7 the cell that should be broken is in the second column followed by another column. Therefore the following command was inserted in the cell as \TeX-Code behind “Castelchiodato,”:

```
& 111\ \newpage
```

The “111” in the third columns of the row was deleted. \newpage is only needed when a page break should definitively occur at this position, otherwise it is only a possibility to break. If your footer row of the longtable has for a certain reason no upper line but you would have a horizontal line where the cell is broken, use this command instead:

```
& 111\
\hline &
```

When the cell to be broken is in the last column, the command \setlength{\parfillskip}{0pt} must be inserted as \TeX-Code at the beginning of the cell. This assures that the part of the cell that will be displayed on the new page appears with the full width.

**Table 2.7.:** Table with forced page break in table cell

<table>
<thead>
<tr>
<th>Example Phone List (ignore the names)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
</tr>
<tr>
<td>Annovi</td>
</tr>
<tr>
<td>Bertoli</td>
</tr>
</tbody>
</table>

continued on next page
## 2.7. Special Longtable Issues

Continued Example Phone List

<table>
<thead>
<tr>
<th>NAME</th>
<th>TEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bozzi</td>
<td>Walter</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maria</td>
</tr>
<tr>
<td>Cachia</td>
<td>Maurizio</td>
</tr>
<tr>
<td>Cinquemani</td>
<td>Giusi</td>
</tr>
<tr>
<td>Colin</td>
<td>Bernard</td>
</tr>
<tr>
<td>Concli</td>
<td>Gianfranco</td>
</tr>
<tr>
<td>Dal Bosco</td>
<td>Carolina</td>
</tr>
<tr>
<td>Dalpiaz</td>
<td>Annamaria</td>
</tr>
<tr>
<td>Feliciello</td>
<td>Domenico</td>
</tr>
<tr>
<td>Focarelli</td>
<td>Paola</td>
</tr>
<tr>
<td>Galletti</td>
<td>Oreste</td>
</tr>
<tr>
<td>Gasparini</td>
<td>Franca</td>
</tr>
<tr>
<td>Lassini</td>
<td>Giancarlo</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Luciano</td>
</tr>
<tr>
<td>Malfatti</td>
<td>Valeriano</td>
</tr>
<tr>
<td>Meneguzzo</td>
<td>Roberto</td>
</tr>
<tr>
<td>Mezzadra</td>
<td>Roberto</td>
</tr>
<tr>
<td>Pirpamer</td>
<td>Erich</td>
</tr>
<tr>
<td>Pochiesa</td>
<td>Paolo</td>
</tr>
<tr>
<td>Radina</td>
<td>Claudio</td>
</tr>
</tbody>
</table>

continued on next page
<table>
<thead>
<tr>
<th>NAME</th>
<th>TEL.</th>
</tr>
</thead>
<tbody>
<tr>
<td>11. Herzog von Poggionativo,</td>
<td></td>
</tr>
<tr>
<td>11. Markis von Mentana,</td>
<td></td>
</tr>
<tr>
<td>Norma,</td>
<td></td>
</tr>
<tr>
<td>Civitella,</td>
<td></td>
</tr>
<tr>
<td>Pratica, Moricone und Percille,</td>
<td></td>
</tr>
<tr>
<td>Baron von Cropalati, 11.</td>
<td></td>
</tr>
<tr>
<td>Herr von Scarpa, Edelmann von Rom,</td>
<td></td>
</tr>
<tr>
<td>Patrizier von Venedig, Neapel und</td>
<td></td>
</tr>
<tr>
<td>Genua</td>
<td></td>
</tr>
<tr>
<td>Stuffer</td>
<td>Oskar 111</td>
</tr>
<tr>
<td>Tacchelli</td>
<td>Ugo 111</td>
</tr>
<tr>
<td>Tezzele</td>
<td>Margit 111</td>
</tr>
<tr>
<td>Unterkalmsteiner</td>
<td>Frieda 111</td>
</tr>
<tr>
<td>Vieider</td>
<td>Hilde 111</td>
</tr>
<tr>
<td>Vigna</td>
<td>Jürgen 111</td>
</tr>
<tr>
<td>Weber</td>
<td>Maurizio 111</td>
</tr>
<tr>
<td>Winkler</td>
<td>Franz 111</td>
</tr>
</tbody>
</table>
2.8. Multiple Lines Columns and Rows

2.8.1. Multiple Lines in Table Cells

Table 2.8.: Table with multiple lines in cells

<table>
<thead>
<tr>
<th>multiple lines</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td>g</td>
<td>h</td>
<td>i</td>
</tr>
</tbody>
</table>

Adjusting a fixed width for a column, enables to enter text as a paragraph with multiple lines and hyphenations.

To produce Table 2.8, create a $3 \times 3$ table, mark the first cell and right-click on it. In the appearing table dialog we set a cell width of 2.5 cm and choose centered for the vertical and horizontal alignment. The vertical alignment is used for all cells of the row. As our text is smaller than than 2.5 cm, only one line will appear. To get two lines, a justified line break (shortcut Ctrl+Shift+Return) was added. If the text is wider than the set cell width, it will automatically be broken to several lines.

If you have a long word in a cell with a fixed width, it cannot be hyphenated by \LaTeX if it is the first entry. Therefore you need to insert something, to make the word not being the first entry. So add a horizontal space of 0 pt before the word. As the space is zero, it doesn’t change the output. Table 2.9 shows the effect.

Table 2.9.: Table with and without hyphenation

<table>
<thead>
<tr>
<th>verylongtablecellword</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td>g</td>
<td>h</td>
<td>i</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>verylongtablecellword</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td>g</td>
<td>h</td>
<td>i</td>
</tr>
</tbody>
</table>

2.8.2. Multicolumns

2.8.2.1. Multicolumn Basics

To span a cell over multiple columns, mark as much cells within a line that should be one spanned cell and use either the table-toolbar button \, or the menu Edit
2. Tables

Table Multicolumn, or right click on the marked cells and choose multicolumn in the appearing table dialog under the tab Table Settings.

Multicolumns have their own cell settings. That means changing cell borders, cell alignment, and the width only affects the multicolumn. Here is an example table with a multicolumn cell in the first row and one in the last row without the upper border:

<table>
<thead>
<tr>
<th>abc</th>
<th>def ghi</th>
<th>jkl</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

2.8.2.2. Multicolumn Calculations

L\LaTeX{}X supports multicolumns directly, but we have to take notice of the cell width of the columns spanned by the multicolumn cell.

Table 2.10.: Table with centered multicolumn text above two columns that have exactly half the width of the multicolumn cell

<table>
<thead>
<tr>
<th>multiple lines</th>
<th>multicolumn</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td>g</td>
<td>h</td>
<td>i</td>
</tr>
</tbody>
</table>

To create for example Table 2.10, mark the first two cells in the first row of a 3×3 table and right-click on them. Now choose for this cell multicolumn, centered alignment and a width of 2.5 cm in the table dialog. The spanned columns should have exactly half the width of the multicolumn cell, so that you would adjust a width of 1.25 cm for the first column. The second column has then automatically a width of 1.25 cm (multicolumn width - width of first column). This was done for Table 2.11.

You can see that the first column has not the half width of the multicolumn cell, it is a bit bigger. The reason is that the given width of a cell $W_g$ is not its total width

Table 2.11.: Table where the spanned table columns have not exactly half the width of the multicolumn cell

<table>
<thead>
<tr>
<th>multiple lines</th>
<th>multicolumn</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td>g</td>
<td>h</td>
<td>i</td>
</tr>
</tbody>
</table>
$W_{\text{tot}}$ because a cell is always a bit larger than its given width. Appendix (C) explains it in detail.

The needed given width $W_{gn}$ when $n$ columns are spanned can be calculated, so that each column has a total width of $W_{\text{tot multicolumn}}/n$:

$$W_{gn} = (W_{\text{multicolumn}} + (1 - n) \cdot (12.4 \text{ pt}))/n \quad (2.1)$$

In our case we have $n = 2$, $W_{\text{multicolumn}} = 2.5 \text{ cm}$ and the default values for the lengths, so that equation 2.1 becomes

$$W_{g2} = 1.25 \text{ cm} - 6.2 \text{ pt} \quad (2.2)$$

To enable calculations in \LaTeX, the \LaTeX-package `calc` must be loaded with the document preamble line

\begin{verbatim}
\usepackage{calc}
\end{verbatim}

\LaTeX does not allow to calculate lengths in the width-field of the table dialog. Therefore you have to format the column by inserting a \LaTeX-argument in the dialog. Here is an overview about the arguments:

- \texttt{p\{width\}} creates cell with a fixed width, its text is vertically top-aligned
- \texttt{m\{width\}} creates cell with a fixed width, its text is vertically centered
- \texttt{b\{width\}} creates cell with a fixed width, its text is vertically bottom-aligned

By entering a \LaTeX-argument, all cell settings set in the table dialog are overwritten. **Note:** Due to a bug, \LaTeX shows the overwritten settings anyway.

As the text should be horizontally centered, the command \texttt{\centering} is added. You can now enter the following \LaTeX-argument for the first spanned column:

\begin{verbatim}
>\{\centering\}m\{1.25cm-6.2pt\}
\end{verbatim}

The command >\{ \} means, that the commands inside the braces are applied before the cell is created.

Although we have chosen centered alignment for the text of the multicolumn cell, it is still left aligned. This is because \LaTeX only applies the alignment to single columns. So we have to use for the multicolumn the \LaTeX-argument

\begin{verbatim}
>\{\centering\}m\{2.5cm\}
\end{verbatim}

\subsection{Multirows}

In contrary to multicolumns multirows are not yet supported by \LaTeX so a bit of \TeX-Code needs to be used. To use multirows load the \TeX-package `multirow` in your document preamble with the command
2. Tables

\texttt{\usepackage{multirow}}

Multirows are created with the command

\texttt{\multirow\{number of rows\}\{cell width\}\{cell entry\}}

To create the following table:

\begin{tabular}{ccc}
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>multirow entry</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td></td>
<td>h</td>
<td>i</td>
</tr>
</tbody>
</table>
\end{tabular}

create a $3\times3$ table. To get rid of the line above the last cell in the first column, the cell is marked as multicolumn and the upper border is unset. The multirow is now created in the second row of the first column by inserting there the command

\texttt{\multirow\{2\}\{2.5cm\}\{}

as $\text{T}\LaTeX$-Code. According to the command parameters the multirow spans now two rows and has a width of 2.5 cm. The content of the multirow cell follows outside the $\text{T}\LaTeX$-Code box and the command is finished with a right brace } in another $\text{T}\LaTeX$-Code-box behind the text.

\texttt{\multirow} left-aligns its content by default. To override the default, renew the command $\texttt{\multirowsetup}$ with the command

\texttt{\renewcommand\{\multirowsetup\}\{\centering\}}

in $\text{T}\LaTeX$-Code in the document preamble. Then all entries of multirow cells in the document are centered. If centering is only needed for several tables, you can renew the command in an $\text{T}\LaTeX$-Code box just before the table instead of the preamble. If the text should be right-aligned, replace $\texttt{\centering}$ by $\texttt{\raggedright}$. To return to left-alignment $\texttt{\raggedleft}$ is used.

2.9. Formal Tables

Tables are often typeset in books similar to Table 2.12. This kind of tables is called “formal”. To make a table a formal table use the option $\texttt{Formal}$ in the $\texttt{Borders}$ tab of the table dialog.

Spaces to table rows can be added using the $\texttt{Borders}$ tab of the table dialog as described in section 2.12.1.

In contrary to normal tables, formal tables have no vertical table lines. The horizontal table lines can be set like for normal tables but they appear with different width in the output:

The first and the last table line have a default width of 0.08 em while the other lines have a default width of 0.05 em.
The default widths can be changed with the following preamble lines:
\let\mytoprule\toprule
\renewcommand{\toprule}{\mytoprule[width]}

This example is for the first line, the so called \texttt{toprule}. If you want to change the width for the last line, replace \texttt{toprule} by \texttt{bottomrule}. To change the width for the other lines replace \texttt{toprule} by \texttt{midrule}. You can use all units listed in appendix A to set the width.

Lines that don’t span over all table columns can be created by setting a table line for multicolumn cells. \LaTeX{} will then internally use the command \texttt{\cmidrule} to create this line. Its full scheme is
\texttt{\cmidrule[width](trim){startcol-endcol}}

The options of \texttt{\cmidrule} are are currently not supported by \LaTeX{} so you have to use \TeX{}-Code to be able to use them. \texttt{\cmidrules} can manually be created by inserting the command as \TeX{}-Code as first cell entry of the first cell of a row. The line is then drawn in the output above the current row.

The default for the width is 0.03\,em. Startcol is the number of the column where the line starts and endcol the column number where the line ends. The endcol always needs to be specified, also when the line should span only one column. The optional parameter trim could be either \texttt{l(trimwidth)}, or \texttt{r(trimwidth)} where the trimwidth is also optional. Using for example the parameter \texttt{l{2pt}} means that the line is trimmed from its left end by 2\,pt. If you don’t specify the trimwidth the lines are trimmed by the default of 0.5\,em.

Table 2.12 was created using the commands
\texttt{\cmidrule(r){2-2}\cmidrule(l){3-4}}

at the beginning of the in the second row and

\begin{table}
\centering
\caption{Example \texttt{booktabs}-table}
\begin{tabular}{lrr}
\hline
System & Medipix 1 & Medipix 2 \\
\hline
Detector thickness [\textmu m] & 300 & 300 & 700 \\
Edge angle [$^\circ$] & 3.55 & 2.71 & 7.99 \\
Spatial resolution [\textmu m] & 4.26 & 10.17 & 10.56 \\
MTF at $f_{\text{max}}$ & 0.53 & 0.37 & 0.39 \\
\hline
LSF-spatial resolution & \\
in \textmu m & 129.7 & 52.75 & 50.78 \\
in \% of pixel size & 76.3 & 95.9 & 92.3 \\
\hline
\end{tabular}
\end{table}
2. Tables

\texttt{\cmidrule(l{10pt}){1-1}}
in the sixth row.

You might want to have overlapping \texttt{\cmidrules} like in Table 2.13. This can be achieved with the \TeX-Code command

\texttt{\morecmidrules}

The command that was used for the second row of Table 2.13 is

\texttt{\cmidrule(r){2-2}\cmidrule(l){3-4}\morecmidrules\cmidrule{2-4}}

The command for the sixth row is

\texttt{\midrule\morecmidrules\cmidrule{3-4}}

If you are anyway not satisfied with the border line spacing, you can use the following command to produce lines that span over all table columns:

\texttt{\specialrule{width}{space above}{space below}}

For more information about these specialties, we refer to the manual of the L\TeX-package \texttt{booktabs}[4].

\begin{table}[h]
\centering
\begin{tabular}{llll}
\hline
System & Medipix 1 & Medipix 2 \\
\hline
Detector thickness [\SI{}{\mu m}] & 300 & 300 & 700 \\
\hline
Edge angle [$^\circ$] & 3.55 & 2.71 & 7.99 \\
Spatial resolution [\SI{}{\mu m}] & 4.26 & 10.17 & 10.56 \\
MTF at $f_{\text{max}}$ & 0.53 & 0.37 & 0.39 \\
\hline
LSF-spatial resolution & & & \\
in $\SI{}{\mu m}$ & 129.7 & 52.75 & 50.78 \\
in \% of pixel size & 76.3 & 95.9 & 92.3 \\
\hline
\end{tabular}
\caption{Special booktabs-table}
\end{table}

2.10. Vertical Table Alignment

To align tables vertically in a text line the table must be inside a box. The box can then be vertically aligned as described in section 5.2.

In the following example the tables are inside a minipage\textsuperscript{3} box that has a width of 15 \texttt{col\%}:

\textsuperscript{3}Minipages are described in section 5.4.
2.11. Colored Tables

2.11.1. Colored Cells

Table 2.14.: Table colored without using the package \texttt{colortbl}

If you only need colored text, mark the cells and choose a color in the menu \texttt{Edit} > \texttt{Text Style}. This was used to create Table 2.14. In any other case you have to use the \LaTeX-package \texttt{colortbl}.

4Raiseboxes are described in section 5.6.2.
To create colored tables, \texttt{colortbl} must be loaded in the preamble with the line
\usepackage{colortbl}

The color of a column is adjusted with the command
\columncolor\{name of color\}
inside the command >\{ \}. More about the command >\{} is described in section 2.8.2.2.

The following color names are predefined:
\texttt{red}, \texttt{green}, \texttt{yellow}, \texttt{blue}, \texttt{cyan}, \texttt{magenta}, \texttt{black} and \texttt{white}

You can also define your own color with the command
\definecolor\{color name\}\{color model\}\{color values\}

The color model can be
\texttt{cmyk: cyan, magenta, yellow, black}
\texttt{rgb: red, green blue}
\texttt{gray}
and the color values are comma separated numbers between 0 and 1 describing the factor for the corresponding color of the color model.

You can e.g. define the color "\texttt{darkgreen}" in the preamble with
\definecolor\{darkgreen\}\{cmyk\}\{0.5, 0, 1, 0.5\}
and the color "\texttt{lightgray}" with
\definecolor\{lightgray\}\{gray\}\{0.8\}

Lines are colored with the command
\rowcolor\{name of color\}
and cells are colored with the command
\cellcolor\{name of color\}

Both commands are inserted at the beginning of a cell as \TeX-Code.

To color characters in the table, mark the cells and use the LyX menu Edit > Text Style. If a cell contains \TeX-Code mark only the characters, otherwise the colored \TeX-Code will cause \TeX-errors.

\textbf{Note:} Not all DVI-viewers are able to display self-defined colors.

To create Table 2.15 do the following: The color of the first column should be \texttt{darkgreen}. So insert

>\{\columncolor\{darkgreen\}\texttt{\centering}c

306
2.11. Colored Tables

as \LaTeX-argument for this column. The first row should be blue, therefore the \LaTeX-Code command
\rowcolor{cyan}

is inserted to the first cell of this row. Note that this overwrites the column color for the first cell. The last cell of the last row is colored magenta by inserting the \LaTeX-Code command
\cellcolor{magenta}

The characters could now be colored using the menu Edit > Text Style.

Table 2.15.: Table colored using the package \texttt{colortbl}

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td>g</td>
<td>h</td>
<td>I</td>
</tr>
</tbody>
</table>

2.11.2. Colored Lines

As described in section 2.12.4, the line thickness for all lines in a table can be adjusted with the length \texttt{arrayrulewidth}. It is set to 1.5 pt for all tables of this section.

To color vertical lines for example with green, create the following column format in the document preamble, according to the description in section 2.12.3:
\newcolumntype{W}{!{\color{green}\vline}}

For Table 2.16 the \LaTeX-argument \texttt{WcW} was used for the last column and \texttt{Wc} for the other columns.

If you want to have several colors, define more column formats.

Table 2.16.: Table with colored vertical lines

<table>
<thead>
<tr>
<th>sd</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>sd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sd</td>
</tr>
</tbody>
</table>

To color horizontal lines for example with red, like in Table 2.17, insert these commands in \LaTeX-Code before the table or table float:
\let\myHlineC\hline
\renewcommand{\hline}
{\arrayrulecolor{red}\myHlineC\arrayrulecolor{black}}
2. Tables

Table 2.17.: Table with colored horizontal lines

<table>
<thead>
<tr>
<th>sd</th>
<th></th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sd</td>
<td></td>
</tr>
</tbody>
</table>

To return to the default line color black, insert this command in TeX-Code behind the table or table float:

\renewcommand{\hline}{\myHlineC}

Table 2.18 is an example with colored vertical and horizontal lines.

Table 2.18.: Table with colored lines

<table>
<thead>
<tr>
<th>sd</th>
<th></th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sd</td>
<td></td>
</tr>
</tbody>
</table>

2.12. Table Customization

2.12.1. Row Spacing

You can add vertical space to table rows in the Borders tab of the table dialog. You find there three possibilities:

Top of row will add space above the characters of the table row. If the table is a formal table\textsuperscript{5} \LaTeX{} will insert as default 0.5\text{em} space. For normal tables the inserted space will unfortunately destroy the vertical table lines as in the following table:

<table>
<thead>
<tr>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 mm space top of row</td>
</tr>
<tr>
<td>C</td>
</tr>
</tbody>
</table>

So inserting space to the top of row for normal tables is only useful when you don’t have vertical lines.

Bottom of row will add space below the characters of the table row. If the table is a formal table \LaTeX{} will insert as default 0.5\text{em} space, for normal tables the default size is 2 pt.

\textsuperscript{5}Formal tables are explained in section 2.9.
2.12. Table Customization

**Between rows** adds space between the current and the following row. If the table is a formal table \LaTeX{} will insert as default 0.5em space. For normal tables the inserted space will unfortunately destroy the vertical table lines as in the following table:

\[
\begin{array}{c}
A \\
\downarrow 3 \text{ mm space between row} \\
\uparrow 3 \text{ mm space between row}
\end{array}
\]

So inserting space between rows for normal tables is only useful when you don’t have vertical lines.

When you want to add extra height to all cells of all tables, you can do this with the following preamble lines:

\[
\begin{array}{l}
\text{\@ifundefined{extrarowheight}} \\
\{\text{\usepackage{array}}\}{} \\
\text{\setlength{\extrarowheight}{height}}
\end{array}
\]

But this has the disadvantage that the cell texts are no longer exactly vertically centered. (The package array will be loaded automatically by \LaTeX{} when you use self defined table formats. To avoid that it is loaded twice the command \texttt{\@ifundefined} is used in the above command.)

In case you are using font sizes larger than the normal size, the table borders are often too close to the letters. This can be corrected by inserting the command \texttt{\strut} in \TeX{} code at the beginning of a table row. Table 2.19 visualizes the effect.

**Table 2.19.** Vertical alignment of text with large font sizes.

(a) Normal table.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
</tr>
<tr>
<td>g</td>
<td>g</td>
<td>g</td>
<td>g</td>
<td>g</td>
</tr>
<tr>
<td>Large</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
</tr>
<tr>
<td>Larger</td>
<td>Larger</td>
<td>Larger</td>
<td>Larger</td>
<td>Larger</td>
</tr>
<tr>
<td>Largest</td>
<td>Largest</td>
<td>Largest</td>
<td>Largest</td>
<td>Largest</td>
</tr>
<tr>
<td>Huge</td>
<td>Huge</td>
<td>Huge</td>
<td>Huge</td>
<td>Huge</td>
</tr>
<tr>
<td>Huger</td>
<td>Huger</td>
<td>Huger</td>
<td>Huger</td>
<td>Huger</td>
</tr>
</tbody>
</table>

(b) Table using the command \texttt{\strut}.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
</tr>
<tr>
<td>g</td>
<td>g</td>
<td>g</td>
<td>g</td>
<td>g</td>
</tr>
<tr>
<td>Large</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
</tr>
<tr>
<td>Larger</td>
<td>Larger</td>
<td>Larger</td>
<td>Larger</td>
<td>Larger</td>
</tr>
<tr>
<td>Largest</td>
<td>Largest</td>
<td>Largest</td>
<td>Largest</td>
<td>Largest</td>
</tr>
<tr>
<td>Huge</td>
<td>Huge</td>
<td>Huge</td>
<td>Huge</td>
<td>Huge</td>
</tr>
<tr>
<td>Huger</td>
<td>Huger</td>
<td>Huger</td>
<td>Huger</td>
<td>Huger</td>
</tr>
</tbody>
</table>
2.12.2. Special Cell Alignment

Sometimes it looks better when the cell entries of a column are aligned with a special character, e.g. with the decimal separator as in Table 2.20.

Table 2.20.: Table cells of a column aligned with the decimal separator.

<table>
<thead>
<tr>
<th>heading</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.6</td>
</tr>
<tr>
<td>0.68</td>
</tr>
<tr>
<td>-123.0</td>
</tr>
</tbody>
</table>

This table was created with a $4 \times 2$ table. The heading is a centered multicolumn. The first column is right-aligned and contains the digits before the decimal point and the decimal point. The second column is left aligned and contains the digits after the decimal point. To omit the space that is normally between two table columns, use the following \LaTeX-argument for the second column:

@{}l

Table 2.21 shows some example alignments. For the alignment with the relation sign, you must add the second smallest math-space at the beginning of the last column to get the correct space surrounding the relation sign.

There is also the \LaTeX-package \texttt{dcolumn} that provides table cell alignments. But this unfortunately treats the cell entries as math and doesn’t allow formulas in table cells: The first column of Table 2.21 will look with \texttt{dcolumn} like the first column in Table 2.22 and only with some tricks like the expected. The alignment of the second and third column of Table 2.21 is not possible with \texttt{dcolumn}.

2.12.3. Customized Cell/Column Format

Calculating the needed width for spanned columns like in section 2.8.2.2 is very annoying if you have several tables with multicolumn cells. To make life easier, you can define a cell/column format in the preamble, so that it can be used in all tables of the document. The format is defined with the command

\begin{verbatim}
@{}l
\end{verbatim}

Table 2.21.: Several table cell alignments.

<table>
<thead>
<tr>
<th>units</th>
<th>exponents</th>
<th>relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$12 \times 24$ bottles</td>
<td>$10 \cdot 10^{-17}$</td>
<td>$\Gamma(t) \propto \Upsilon(t)$</td>
</tr>
<tr>
<td>$1024 \times 768$ Pixels</td>
<td>$5.78 \cdot 10^7$</td>
<td>$A \neq B_{\text{red}}$</td>
</tr>
<tr>
<td>$32 \times 6$ cm</td>
<td>$-33.5 \cdot 10^4$</td>
<td>$\sin(\alpha) \geq \sin(\beta)$</td>
</tr>
</tbody>
</table>
Table 2.22.: Alignments when \LaTeX-package dcolumn is used. For all column alignments tricks have to be used to get the output.

<table>
<thead>
<tr>
<th>units</th>
<th>units</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>12\times 24 bottles</td>
<td>12\times 24 bottles</td>
<td>12 \times 24 bottles</td>
</tr>
<tr>
<td>1024\times 768 Pixels</td>
<td>1024\times 768 Pixels</td>
<td>1024 \times 768 Pixels</td>
</tr>
<tr>
<td>32\times 6 cm</td>
<td>32\times 6 cm</td>
<td>32 \times 6 cm</td>
</tr>
</tbody>
</table>

\texttt{\newcolumntype\{name of format\}[number of arguments]{commands}}

The format name may only consist of one letter. The letters b, c, l, m, p and r are predefined and cannot be used. But all letters are allowed as capitals.

For vertically and horizontally centered multicolumn cells with a fixed width you can define the cell format

\texttt{\newcolumntype\{M\}[1]{>{\centering\hspace{0pt}}m{#1}}

where \texttt{\hspace{0pt}} avoids the problem of hyphenating the first word, as described in section 2.8.1. Now you can simply enter

\texttt{M\{width\}}

as \LaTeX-argument in the table dialog to create a multicolumn.

For cells spanned by a multicolumn cell, you can define the format

\texttt{\newcolumntype\{S\}[2]{>{\centering\hspace{0pt}}m{(\#1+(2\\tabcolsep+\arrayrulewidth)*(1-\#2))/\#2}}

This format uses equation (2.1) to calculate the needed width so that each spanned cell has the same width.

You can now enter

\texttt{S\{width of multicolumn cell\}\{number of spanned columns\}}

as \LaTeX-argument of the column.

For colored columns, you can define

\texttt{\newcolumntype\{K\}[1]{>{\columncolor\{\#1\}\hspace{0pt}}c}

The “c” at the end creates a column with a flexible width whose text is horizontally centered. You can now enter

\texttt{K\{color name\}}

as \LaTeX-argument.

To create Table 2.23 use the \LaTeX-arguments
2. Tables

M{2.5cm}

for the first column and the multicolumn,

K{red}

for the the last column, and

S{2.5cm}{2}

for the cells in the second column.

Table 2.23.: Table using user-defined table formats

<table>
<thead>
<tr>
<th>very-longtablecell-word</th>
<th>multiple lines multicolumn</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td></td>
<td>g</td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>i</td>
<td>j</td>
</tr>
<tr>
<td></td>
<td>k</td>
<td></td>
</tr>
</tbody>
</table>

2.12.4. Line Thickness

The line thickness for all lines in a table can be adjusted with the length \arrayrulewidth. To set for example a line thickness of 1.5pt, like in Table 2.24, insert the command

\setlength{\arrayrulewidth}{1.5pt}

in \TeX-Code before the table or table float. The changed thickness is valid for all following tables. To use the default value again, set \arrayrulewidth to 0.4pt in \TeX-Code behind the table or table float.

To set the line thickness to 1.5pt only for horizontal lines, like in Table 2.25, insert these commands in \TeX-Code before the table or table float:

\let\myHline\hline
\renewcommand{\hline}{\noalign{\global\arrayrulewidth 1.5pt} \myHline \noalign{\global\arrayrulewidth 0.4pt}}

Table 2.24.: Table with 1.5 pt thick lines

<table>
<thead>
<tr>
<th>sd</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sd</td>
</tr>
<tr>
<td></td>
<td>sd</td>
</tr>
</tbody>
</table>
Table 2.25.: Table with 1.5 pt thick horizontal lines

<table>
<thead>
<tr>
<th>sd</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>sd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sd</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.26.: Table with 1.5 pt thick vertical lines

<table>
<thead>
<tr>
<th>sd</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>sd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sd</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To return to the default line thickness, insert this command in \TeX{}-Code behind the table or table float:

\renewcommand{\hline}{\myHline}

To set the line thickness to 1.5 pt only for vertical lines, create the following column format in the document preamble, according to the description in section 2.12.3:

\newcolumntype{V}{!{\vrule width 1.5pt}}

For Table 2.26 the \LaTeX{}-argument

VcV

was used for the last column and

Vc

for the other columns.

2.12.5. Dashed Lines

Table 2.27.: Table with dashed lines

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
</tr>
<tr>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
</tr>
<tr>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td></td>
</tr>
<tr>
<td>t</td>
<td>u</td>
<td>v</td>
<td>w</td>
<td>x</td>
</tr>
</tbody>
</table>

\L{}y\TeX{} doesn’t natively support dashed lines, so you have to use \TeX{}-Code. As prerequisite the \LaTeX{}-package \texttt{arydshln} must be loaded in the document preamble with the command
To make a vertical line dashed, enter the colon “:” together with the character for the horizontal alignment as LaTeX-argument in the table cell dialog.

For a horizontal dashed line add the command

\hdashline

in \LaTeX-Code as first element of the first cell in the table row.

For dashed multicolumn lines use the command

\cdashline{line number}

in \LaTeX-Code as first element of the first cell in the table row. If you have for example a multicolumn spanning over columns 2 to 4 and you want to have a dashed line above, add the command

\cdashline{2-4}

as first element of the first cell in the row of the multicolumn.

Table 2.27 was created using “:c” as \LaTeX-argument of the third column. The \LaTeX-Code command \hdashline was inserted to the first cell of the third row and the \cdashline{4-5} was inserted to the first cell of the fourth row.

**Note:** The used \LaTeX-package **arydshln** is apparently not compatible with the \LaTeX-package **colortbl** that is used for colored tables in section 2.11. That means colored tables cannot have dashed lines.
3. Floats

3.1. Introduction

A float is a block of text associated with some sort of label, which doesn’t have a fixed location. It can “float” forward or backward a page or two, to wherever it fits best. Footnotes and Margin Notes are also floats, because they can float to the next page when there are too many notes at the page.

Floats allow a high quality layout. Images and tables can evenly be spread to the pages to avoid white space and pages without text. As the floating often destroys the context between the text and the image/table, every float can be referenced in the text. Floats are therefore numbered. Referencing is described in section 3.4.

To insert a float, use the menu Insert ➤ Float. This inserts the Caption inset, a box with a label. The label will automatically be translated to the document language in the output. Behind the label you can insert the caption text. The image or table is inserted above or below the caption in a separate paragraph within the float. More about the caption placement is described in section 3.10. To keep your LyX-document readable, you can open and close the float box by left-clicking on the box label. A closed float box looks like this: float Figure – a gray button with a red label.

It is recommended to insert floats as a separate paragraph to avoid possible \LaTeX- errors that can occur when the surrounding text is specially formatted.

Existing figures or tables can be put into a float by marking them and then pressing the corresponding toolbar button for a new float.

3.2. Float Types

Besides figure and table floats that are described in section 1.2 and 2.5, respectively, LyX offers the float types Algorithm and Wrap.

3.2.1. Algorithm Floats

This float type is inserted with the menu Insert ➤ Floats ➤ Algorithm. It is used for program codes and descriptions of algorithms and can be seen as an alternative
3. Floats

Algorithm 3.1 Example Algorithm float

\begin{algorithm}
\begin{algorithmic}
\For {$I$ in 1..N loop}
\State $\text{Sum} := \text{Sum} + A(I)$; /*comment*/
\EndFor
\end{algorithmic}
\end{algorithm}

to program code listings that are explained in chapter 7. A possible environment for algorithms is the {\LaTeX}-Code, described in {\LaTeX}'s Userguide. Algorithm 3.1 is an example of an algorithm float where -4mm vertical space was added at the end of the float to have the bottom rule exactly below the last text line.

The float label is not automatically translated into the document language. If your document is not in English, you have to do this manually by adding the following line to the document preamble:

\begin{verbatim}
\texttt{\textbackslash{floatname}{algorithm}\{your name\}}
\end{verbatim}

where \texttt{your name} is the word “\texttt{algorithm}” in your language.

To insert the list of algorithms you can use the menu Insert $\Rightarrow$ List / TOC $\Rightarrow$ List of Algorithms when your document has the same language as {\LaTeX}'s menu names. In other cases use this command in {\TeX}-Code instead:

\begin{verbatim}
\texttt{\textbackslash{listof}{algorithm}\{your name\}}
\end{verbatim}

where \texttt{your name} is the word “\texttt{List of Algorithms}” in your language.

Algorithm floats are not by default numbered in the scheme “chapter.algorithm” like it is the case for table and figure floats in many document-classes. To number algorithm floats in the same scheme, add this command to your document preamble:

\begin{verbatim}
\texttt{\textbackslash{numberwithin}{algorithm}\{chapter\}}
\end{verbatim}

To be able to use the command \texttt{\textbackslash{numberwithin}}, set in the tab Math Options in the document settings the option Use AMS math package.

3.2.2. Wrap Floats

This float type is used if you want to wrap text around a figure or table so that it only occupies some fraction of the column width. It can be inserted using the menu Insert $\Rightarrow$ Floats $\Rightarrow$ Figure Wrap Float or Table Wrap Float if the {\TeX}-package \texttt{wrapfig} is installed.\footnote{Installing a {\TeX}-package is explained in the {\TeX} Configuration manual.} The settings of the float can be modified by right-clicking on the float box. The mandatory settings are the float Placement and its Width. Optional are the Overhang...
that specifies how much the float is set into the paragraph/page margin, and the Line span that specifies how many text lines the float will approximately need. The line span is often hard to be approximated, so better only use it when you encounter float placement problems. You can furthermore decide if \LaTeX is allowed to let the float float within the paragraph or to surrounding paragraphs. Figure 3.1 is an example text wrap float with a width of 40\% col, 1 cm overhang, set to the left.\footnote{Available units are explained in appendix A.}

\textbf{Note:} Text wrap float floats are fragile! E.g. having a figure too close to the bottom of the page can mess things up in the way that the float doesn’t appear in the output or that it is placed over some other text.

In general:

- Wrap floats should not be placed in paragraphs that run over a page break. That means that wrap floats should better be inserted to the exact place when the document is nearly finished and you are able to estimate where page breaks will appear.
- Wrap floats should either be placed in an own paragraph before the paragraph where they should wrap into or within a paragraph.
- Wrap floats in consecutive paragraphs may cause troubles, so assure that there is a text paragraph between them as separator.
- Wrap floats are not allowed in section headings or tables.

\section*{3.3. Float Numbering}

Floats are usually numbered either independent from the sections the floats are in, or in the scheme “chapter.number” or “section.number”. This depends on the used document class.

To change the section independent numbering, you can use this command in the document preamble:

\begin{verbatim}
\renewcommand{\thetable}{\roman{table}}
\thetable
\end{verbatim}

The command \roman prints in the command above the table number as small roman number.

To change the numbering scheme for example to “subsection.number”, use this command in the preamble:

\begin{verbatim}
\numberwithin{table}{subsection}
\end{verbatim}
3. Floats

To be able to use the command `\numberwithin`, set in the tab Math Options in the document settings the option Use AMS math package.

Please also have a look at section 4.2.1 for the details and important notes about the numbering commands.

3.4. Referencing Floats

To reference a float, insert a label into its caption using the menu Insert > Label or the toolbar button ![Label](image). A grey label box like this one: ![Label](image) will be inserted and the label window pops up asking for the label text. \LaTeX\ offers as text the first words of the caption with a prefix. The prefix depends on the float type, e.g. for figure floats the prefix will be "fig:"

The label is used as anchor and name for the reference. You can refer to the label using the menu Insert > Cross-reference or the toolbar button ![Label](image). A grey cross-reference box like this one: ![Label](image) will be inserted and the cross-reference window appear showing all labels of the document. If you have multiple \LaTeX-documents opened, choose the one you are working on from the drop-list at the top of the dialog. You can now sort the labels alphabetically and then choose one. At the position of the cross-reference box the float number will appear in the output.

It is recommended to use a protected space between the cross-reference name and its number to avoid line breaks between them. If a cross-reference refers to a non-existing label, you will see two question marks in the output instead of the reference.

You can change labels at any time by clicking on the label box. References to the changed label will automatically change its link to the new label text, so that you don’t need to take care about this.

The button Go to Label in the cross-reference window sets the cursor before the referred label. The button text changes then to Go Back and you can use it to set the cursor back to the cross-reference. Right-clicking on a cross-reference box also sets the cursor before the referenced label but without a possibility to go back.

3.4.1. Cross-Reference Formats

There are six varieties of cross-references:

- `<reference>`: prints the float number, this is the default: 1.3
- `( <reference> )`: prints the float number within two parentheses, this is the style normally used to reference formulas, especially when the reference name “Equation” is omitted: (2.1)
3.4. Referencing Floats

<page>: prints the page number: Page 283

on page <page>: prints the text 'on page' and the page number: on page 283

<reference> on page <page>: prints the float number, the text 'on page', and the page number: 1.3 on page 283

Formatted reference: prints a self defined cross-reference format. Note: This feature is only available when you have the \LaTeX-package prettyref installed.

Note that the style <page> won’t print the page number if the label is on the previous, the same, or the next page. You will e.g. see the text “on this page” instead.

The number and current page of the referred document part in the output, is automatically calculated by \LaTeX. The varieties are adjusted in the field Format of the cross-reference window, that appear when you click on the cross-reference box.

3.4.2. Automatic Reference Naming

The \LaTeX-package hyperref, that is enabled in the PDF Properties of the Document Settings dialog, provides a very useful feature that cross-references automatically include the name of the referenced floats (or text parts like section). So you will save to write e.g. the name “Figure” before every reference to a figure. To use this feature, enable hyperref and insert this line to the \LaTeX preamble:

\AtBeginDocument{\renewcommand{\ref}[1]{\mbox{\autoref{#1}}}}

When you prefer other reference names than the default ones, for example you want instead of “section” the name “sec.”, you can redefine the name by inserting this to the preamble:

\addto\extrasenglish{\renewcommand{\sectionautorefname}{sec.\negthinspace}}

When you are using another document language than English, replace \extrasenglish by \extras***, where *** is the name of the used language.

To get automatic names, but not for certain reference types, for example not for equations, use this preamble code:

\newlength{\abc}
\settowidth{\abc}{\space}
\addto\extrasenglish{\renewcommand{\equationautorefname}{\hspace{-\abc}}}

More about this topic can be found in hyperref’s documentation [10].

Note: Automatic reference naming cannot be used when you use cross-references in the Formatted reference style, as described in section 3.4.1.

The Math mammal is an example where automatic reference naming is used.
3. Floats

3.4.3. Reference Position

If you use `hyperref` in the PDF Properties of the Document Settings dialog to link cross-references in the output, you will see that clicking on an image float reference jumps to the image label. The caption will be the first text part on the screen, so that you cannot see the image without scrolling. This is because the reference link anchor is placed at the position of the label. With the use of the package `hypcap`, which is part of the L\TeX-package `oberdiek`, the link anchor is placed at the beginning of a float. To use this feature for figure floats, load `hypcap` in the document preamble with the line

\usepackage[figure]{hypcap}

You can also use `hypcap` for all float types but this is not recommended for stability reasons. For more information, have a look at `hypcap`'s manual [9].

Note: `hypcap` has no effect for references to subfigures.

3.5. Float Placement

Right-clicking on a float-box opens a dialog where you can alter the placement options that \LaTeX uses for positioning the float.

The option `Span columns` is only useful for two-column documents: If you select it, the float will span across both columns on the page instead of being confined to just one.

The option `Rotate sideways` is used to rotate floats, see section 3.6.

You can use one or more of the following options in the float dialog to set the placement for a particular float when you uncheck the option `Use default placement`:

- **Here if possible** try to place the float on the position where it is inserted
- **Top of page** try to place the float on the top of the current page
- **Bottom of page** try to place the float on the bottom of the current page
- **Page of floats** try to place the float on an own page

The order of the above option is always used by \LaTeX. That means, if you use the default placement, \LaTeX will first try out **Here if possible**, then **Top of page**, and then the others. If you don’t use the default, \LaTeX will try only the checked options but in the same order. If none of the 4 placements are possible the procedure is internally repeated but it is tried to put the float on the following page.

By default, each option has its own rules:

- **Top of page** only floats occupying less than 70% of the page can be placed at the top of a page (\texttt{\topfraction})

320
3.5. Float Placement

**Bottom of page:** only floats occupying less than 30% of the page can be placed at the bottom of a page. (\texttt{bottomfraction})

**Page of floats:** only if more than 50% of the page are occupied by floats, several floats can be set together on a page. (\texttt{floatpagefraction})

If you don’t like these rules, you can ignore them by using the additional option \texttt{Ignore \LaTeX rules}.

You can also redefine the rules with \LaTeX-commands that are given in parentheses behind the rules description above. To increase for example the often too small default of the bottom-rule to 50% of the page, add this line to your document preamble:

\texttt{\renewcommand{\bottomfraction}{0.5}}

Sometimes you might need, under all circumstances, a float to be placed exactly at the position where it is inserted. For this case you can use the option \texttt{Here definitely}. Use this option very rarely and only if the document is nearly ready to be printed. Because the float is then no longer able to “float” when you change your document and this will often destroy the page layout.

There are no placement options for text wrap floats, because they are always surrounded by the text of a certain paragraph.

Sometimes you have the problem that a float is placed at the top of a page while its corresponding section starts at the middle of the page, so that the reader could think the float is part of the previous section. To avoid this the \LaTeX-command \texttt{\supressfloats} can be used. It suppresses a given float placement for the page where it is inserted and can therefore be used to avoid that floats could be set before a section starts. To get this, add these commands to your document preamble:

\texttt{\let\mySection\section \renewcommand{\section}{\supressfloats[t]\mySection}}

You can define the same for all section headings, like chapters and subsections. This definition is not recommended to be used for small text parts like subsubsections because \LaTeX may then have problems to find a suitable placement.

In some cases it is required to have all figures/tables at the end of the document. For this purpose the \LaTeX-package \texttt{endfloat} was developed. It puts all figure and table floats at the end of the document into own sections. At the original float position a text hint like “[Figure 3.2 about here.]” is inserted. The endfloat-package is loaded in the preamble with the line

\texttt{\usepackage[options]{endfloat}}

There are various package options to format the created figure/table sections. For more information we refer to the \texttt{endfloat} documentation [6].

**Note:** \texttt{endfloat} doesn’t provide an automatic translation for the text hint, you have to do this manually, see section 4 in [6].
3. Floats

**Note:** There is currently a bug in `endfloat` when the caption contains a German “ß”. Use in this case the command “\ss” in \TeX-code instead of “ß”.

For more details about float placements, have a look at \LaTeX books, [1, 2, 3].

3.6. Rotated Floats

Especially for wide tables you might have floats rotated. To rotate a whole float including the caption, right-click on the float-box and use the option `Rotate sideways`.

Rotated floats are always placed on its own page (or column, in case you have a multi-column document). You can let them span several columns using the float settings option `Span columns`. Floats are rotated so that you can read them from the outside margin. To force a certain rotation direction for all pages, you can add either the option `figuresleft` or `figuresright` to the document class options.

Referencing rotated floats is the same like for normal floats, the caption format is also the same: Table 3.1 is an example of a rotated table float.  

**Note:** Not all DVI-viewers are able to display rotated floats.

3.7. Subfloats

Subfloats are for example used when a figure consists of several images. They are created by inserting a float to an existing float. The placement of the subfloats can be controlled like for paragraphs as shown in table 3.2 and 3.3.

Referencing subfloats works as for normal floats: Table 3.2a and 3.2b are subtables of table 3.2.

3.8. Floats Side by Side

To place floats side by side, like for Figure 3.2 and 3.3, only one float is used. In it two minipage boxes are inserted.³ The width is set to 45–50 column% and the box alignment to Bottom for each minipage. The minipage boxes contain the image and the caption in the same way as they are in a float. The only difference is that the image unit `Column Width %` is now calculated according to the width of the minipage boxes.

³Minipages are explained in section 5.4.
Table 3.1.: Rotated table

<table>
<thead>
<tr>
<th>test</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
</table>
3. Floats

**Table 3.2.** Two subtables placed side by side.

(a) This is subtable a.

| test | b | c | d | e |

(b) This is subtable b.

| e | d | c | b | test |

**Table 3.3.** Two subtables placed one upon the other. (a) table with 4 cells, (b) table with 5 cells.

(a)

| test | test | test | test |

(b)

| a | b | c | d | e |

**Figure 3.2.** Float on the left side.

**Figure 3.3.** Float on the right side.
3.9. Caption Formatting

The Caption environment is the default paragraph environment for Floats. On the \LaTeX{} screen captions appear as label, e.g. “Figure 
\#:” followed by the caption text. “\#” is the actual reference number. By default the label and the number are in the same font as the caption text and a colon follows the number to divide the label from the text. This caption format is not suitable for all document formats.

To change the default caption format, load the \LaTeX{}-package \texttt{caption} in the document preamble with this line:

\texttt{\usepackage[format definition]{caption}}

To have for example the label and the number in sans-serif bold font and the table captions always above the table like in this document, use the following command:

\texttt{\usepackage[labelfont={bf,sf}, tableposition=top]{caption}}

You can also define different caption formats for the different float types. In this case load the \texttt{caption} package without format specific options and define the different formats with the help of the command

\texttt{\captionsetup[float type]{format definition}}

in the document preamble. For example the caption formats of Figure 3.4 and Table 3.4 can be created using these commands in the document preamble:

\texttt{\usepackage[tableposition=top]{caption}}
\texttt{\captionsetup[figure]{labelfont={tt}, textfont=it, indentation=1cm,\% labelsep=period}}
\texttt{\captionsetup[table]{labelfont={bf,sf}}}  

Note: The option \texttt{tableposition=top} has no effect when a \texttt{KOMA-script} document class is used. In this case the document class option \texttt{tablecaptionabove} must be used.

For more information about the package \texttt{caption} we refer to its documentation [5].

To change the label name from e.g. “Figure” to “Image” use this preamble command:

\texttt{\renewcommand{\fnum@figure}{Image~\thefigure}}

where \texttt{\thefigure} inserts the figure number and “~” creates a protected space.

If you are using a \texttt{KOMA-script} document class (\texttt{article} (\texttt{KOMA-script}), \texttt{book} (\texttt{KOMA-script}), \texttt{letter} (\texttt{KOMA-script}), or \texttt{report} (\texttt{KOMA-script})), you can alternatively to the \texttt{caption} package use \texttt{KOMA-script}'s built-in command \texttt{\setkomafont}. For example, to have the caption label in bold, add this command to your document preamble:

\texttt{\setkomafont{captionlabel}{\bfseries}}

For more information about \texttt{\setkomafont} we refer to the \texttt{KOMA-script} documentation [11].
3. Floats

3.10. Caption Placement

The common caption placement rule is:

**Figure**: Caption is set below the figure

**Table**: Caption is set above the table

Having the caption above the table is unfortunately not supported in \LaTeX{}’s standard classes. That means if you are using the document classes article, book, letter, or report there will be no space between the caption and the table. To insert the needed space, add the following option to the load command of the \LaTeX-package **caption** in your document preamble:

```
\tableposition=top
```

If you are using a KOMA-script document class (article (KOMA-script), book (KOMA-script), letter (KOMA-script), or report (KOMA-script)), you can alternatively to the **caption** package set the document class option **tablecaptionabove**.

It is also possible to set the caption beside a figure or table. To get this the \LaTeX-package **sidecap** has to be loaded in the document preamble with the line:

```
\usepackage[option]{sidecap}
```

If you set no option, the caption is placed on the side of the outer page margin – to the right on odd pages, to the left on even pages. You can change the placement to inner margin with the option **innercaption**. To force the placement always to the right or left, use the option **rightcaption** or **leftcaption**, respectively.

To place in LyX the caption of a float on the side, it is necessary to add these commands to the document preamble:

```
\newcommand{\TabBesBeg}{\%
  \let\MyTable\table
  \let\MyEndtable\endtable
  \renewenvironment{table}{\begin{SCtable}}{\end{SCtable}}\%}
\newcommand{\TabBesEnd}{\%
  \let\table\MyTable
  \let\endtable\MyEndtable
  \newcommand{\FigBesBeg}{\%
    \let\MyFigure\figure
    \let\MyEndfigure\endfigure
    \renewenvironment{figure}{\begin{SCfigure}}{\end{SCfigure}}\%}
\num
```

\footnote{See section 3.9 for more information of the package **caption**.}

326
3.11. Listings of Floats

The commands allow you to redefine the floats so that the caption is set on the side. For figure floats use the command

\FigBesBeg

in \LaTeX-code before the float. Behind the float insert the command

\FigBesEnd

in \LaTeX-code to get back to the original float definition.

For table floats use the corresponding commands

\TabBesBeg and \TabBesEnd

Figure 3.5 and Table 3.5 are examples where the caption is set beside.

You can see in the examples that the caption text appears at the top of the floats for table floats and at the bottom for figure floats. To change this, you can use the command

\sidecaptionvpos{float type}{placement}

in the document preamble or in \LaTeX-code before the float. The float type is either figure or table, the placement can be “t” for top, “c” for center, or “b” for bottom. To have for example the caption of figure floats vertically centered, use the command

\sidecaptionvpos{figure}{c}

This was used for Figure 3.6.

For more information about the package \texttt{sidecap} we refer to its documentation [15].

\textbf{Note:} The \LaTeX-package \texttt{hypcap}, described in section 3.4.3, has no effect on floats with the caption set beside.

3.11. Listings of Floats

Similar to the the table of contents where the sections of the document are listed, there are listings for all float types, like the figures of the documents. You can insert them via the \textit{Insert} $\rightarrow$ \textit{List / TOC} sub menus.

The list entries are the float captions or its short title, the float number, and the page number where they appear in the document.

You can find the list of figures and tables at the end of this document.
3. Floats

Figure 3.4.: This is an example figure caption that is longer than one line to show the different caption format. Here a self-defined caption format is used.

Table 3.4.: This is an example table caption that is longer than one line to show the different caption format. Here the standard caption format for tables in this document is used.

\[
\begin{array}{ccccc}
  a & b & c & d & e \\
\end{array}
\]

Figure 3.5: This is a caption beside a figure.

Table 3.5: This is a caption beside a table.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>e</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>g</td>
<td>h</td>
</tr>
<tr>
<td>i</td>
<td>j</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.6: This is a vertically centered caption beside a figure.
4. Notes

4.1. LyX Notes

Notes are inserted with the toolbar button or the menu Insert > Note. There are three types of notes:

**LyX Note** This note type is for internal notes that won’t appear in the output. Its note-box looks like this:

```
This is text in a note box that doesn't appear in the output.
```

**Comment** This note also doesn’t appear in the output but it appears as LaTeX-comment, when you export the document to TeX via the menu File > Export > LaTeX (pdflatex) / (plain). Its note-box looks like this:

```
This is text in a note box that only appears as comment in LaTeX-files.
```

**Greyed Out** This note will appear in the output as grey text. Its note-box looks like this:

```
This is text of a comment that appears in the output as grey text.
```

This is text\(^1\) of a comment that appears in the output as grey text.

As you can see in the example, the first line of greyed out notes is a bit indented and greyed out notes can have footnotes.

When you use the toolbar button to insert notes, a LyX Note is inserted. You can switch between the five note types by right-clicking on the note-box. If you want to turn existing text into a note, mark it and click on the note toolbar button. To change a note to text, press the backspace key when the cursor is in the first position of a note, or press the deletey key when the cursor is in the very last position of the note, respectively.

\(^1\)This is an example footnote within a greyed out note.
4. Notes

You can change the text color of the greyed out notes in the preamble with the following command:

\renewenvironment{lyxgreyedout}
{\textcolor{color}\bgroup}{\egroup}

The available colors and the method to define own colors is explained in section 2.11. Notes that appear in blue in this document are set using greyed out notes with blue text.

4.2. Footnotes

Footnotes can be inserted using the toolbar button or the menu Insert>Footnote. You’ll see then the following footnote-box: where you can enter the footnote text. If you want to turn existing text into a footnote, mark it and click on the footnote toolbar button. To change a footnote to text, press the Backspace key when the cursor is in the first position of a footnote, or press the Delete key when the cursor is in the very last position of the footnote, respectively.

Here is an example footnote:\footnote{The footnote will appear in the output as a superscript number at the text position where the footnote box is placed. The footnote text is placed at the bottom of the current page. The footnote number is calculated by \LaTeX, the numbers are consecutive. It depends on your document-class, if the footnote number is reset for every chapter.

Footnotes can be referenced like floats: Insert a label into the footnote and cross-reference this label in the text as described in section 3.4. This is a cross-reference of Footnote 2.

To use footnotes within tables, you have to use minipages, see section 5.4. Footnotes within longtables are described in section 2.6.1.

To create only a mark for a footnote, use the command \footnotemark[number] in \LaTeX-Code. This is used when you have the same annotation several times in a text but doesn’t want to print the footnote text every time.

As you don’t know the number of the repeating footnote while you are writing the text, you have to store its number. For the following footnote mark example, these commands were inserted in \LaTeX-Code behind Footnote 2 to store the footnote number:

\newcounter{MyRepeatFoot}
\setcounter{MyRepeatFoot}{\thefootnote}

\footnotemark{\thefootnote}

\footnotetext{This is an example footnote.}
4.2. Footnotes

The footnote mark was then created with this command:
\footnotemark[\theMyRepeatFoot]
Here is an example footnote mark:\footnotemark[2]

4.2.1. Footnote Numbering

To reset the footnote number back to 1 after each section, add this command to your document preamble:
\@addtoreset{footnote}{section}

The following preamble command changes the footnote numbering style to small roman numerals:
\renewcommand{\thefootnote}{\roman{footnote}}
This is a footnote with roman numbering.\footnote{iii}

To change the numbering style to capital roman numerals replace in the command above \roman by \Roman. To “number” footnotes with capital or small Latin letters use \Alph or \alph, respectively. To “number” footnotes with symbols use \fnsymbol.

Note: You can only number 26 footnotes with Latin letters, because this numbering is limited to single letters.

Note: You can only number 9 footnotes with symbols.

To return to the default numbering style when you changed to another one, use \arabic instead of \roman in the command above.

If you want to have footnotes numbered in the scheme “chapter.footnote”, add the following command to your document preamble:
\numberwithin{footnote}{chapter}

To be able to use the command \numberwithin, set in the tab Math Options in the document settings the option Use AMS math package.

This is another example footnote:\footnote{4.4}

Note: \numberwithin always prints out the footnote number as arabic number; previous redefinitions to get non-arabic numbers are overwritten.

So to get for example the scheme “chapter.\Roman{footnote}”, use this command instead of \numberwithin:
\renewcommand{\thefootnote}{\thechapter.\Roman{footnote}}

\footnote{iii}This is an example footnote with roman numbering.
\footnote{4.4}This is a footnote numbered in the scheme “chapter.footnote”.

331
4. Notes

4.2.2. Footnote Placement

If you have several footnotes in one page, they appear without vertical space between them at the bottom of the page. To make them better readable you can e.g. add 1.5 mm space with the following preamble command:

\let\myFoot\footnote
\renewcommand{\footnote}[1]{\myFoot{#1}\vspace{1.5mm}}

In a two-column document the footnotes appear at the bottom of every column, see Figure 4.1. If the footnotes should only appear at the bottom of the right column, as in Figure 4.2, use the \LaTeX-package \texttt{ftnright} with this command in the document preamble:

\usepackage{ftnright}

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
Sei nun $S$ unser normiertes Ausgangssignal und $P$ die Phasenverteilungsfunktion, so ergibt sich die Beziehung \\
$S(t) = S_0(t) \int_{-\infty}^{\infty} P(\phi, t) e^{i\phi} d\phi$ (2) \\
wobei $S_0$ das Signal ohne Gradient ist und die Normierungsbedingung
\hline
\end{tabular}
\end{table}

1. Fourier transformation  \\
2. Phase distribution function

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{footnote_placement.png}
\caption{Standard footnote placement in two-column documents.}
\end{figure}

In some scientific literature it is usual to collect the footnotes and print them in a separate paragraph at the end of a section, like in Figure 4.3. They are then so called “endnotes”. To use endnotes instead of footnotes in your document, load the \LaTeX-package \texttt{endnotes} with the document preamble lines

\usepackage{endnotes}
\let\footnote\endnote

To insert the collected footnotes, insert the command

\theendnotes

in \TeX-Code at the end of a section or chapter.

The paragraph heading for the endnotes isn’t automatically translated into the document language, this must be done manually. The following preamble command
man über alle Phasen integrieren muss. Sei nun $S$ unser normiertes Ausgangssignal und $P$ die Phasenverteilungsfunktion, so ergibt sich die Beziehung

$$S(t) = S_0(t) \int_{-\infty}^{\infty} P(\phi, t)e^{i\phi} d\phi$$  \hspace{1em} (2)

wobei $S_0$ das Signal ohne Gradient ist und die Normierungsbedingung $\int_{-\infty}^{\infty} P(\phi, t) d\phi = 1$ gilt. Nun dürfen

Das Spektrum wird fouriertransformiert. Die Fouriertransformation wird verwendet, um die überlagerten Signale (Netzwerk, Lösungsmittel) zu trennen. Nachdem wir die Phasenverschiebung bestimmen konnten, interessiert uns nun das Aussehen des

1. Fourier transformation
2. Phase distribution function
3. Fourier transformation
4. Phase distribution function

Figure 4.2.: Footnote placement in two-column documents when the LaTeX-package \texttt{ftnright} is used.

Die F
couriertransformation wird verwendet, um die überlagerten Signale (Netzwerk, Lösungsmittel) zu trennen. Nachdem wir die Phasenverschiebung bestimmen konnten, interessiert uns nun das Aussehen des

1. Fourier transformation
2. Phase distribution function
3. Fourier transformation
4. Phase distribution function

Figure 4.3.: Endnotes – footnotes are printed in a separate paragraph at the end of sections or chapters.
4. Notes

Footnotes can also be placed in the page margin and the footnote text alignment can be changed, see the \LaTeX-package \texttt{footmisc}, [8] for more information about this. For various further footnote formatting issues have a look at \LaTeX-books, [1, 2, 3].

4.3. Margin Notes

Margin notes look and behave in \LaTeX like footnotes. They are inserted via the menu \texttt{Insert} > \texttt{Marginal Note} or the toolbar button $\equiv$. A grey box with the red label “margin” appears where you can enter the text of the margin note.

At the side is an example margin note. Margin notes appear at the right side in single-sided documents. In double-sided documents they appear in the outer margin – left on even pages, right on odd pages. The text of margin notes is aligned opposite to the outer margin – right-aligned when the note appears in the left margin. The first line of the margin note is placed at the position of the text line where it is inserted in the document.

To place the margin note in the inner margin, add the command

$$\texttt{\textbackslash reversemarginpar}$$

in \TeX-Code before a margin note. The new placement is valid for all following margin notes.

\textbf{Note:} There is often not enough space in the inner margin so that the notes are not correctly displayed in the output.

To return to the default placement insert the command

$$\texttt{\textbackslash normalmarginpar}$$

in \TeX-Code. \textbf{Note:} The command is ignored when it is within a paragraph where also the command \texttt{\textbackslash reversemarginpar} is inserted.

Similar to the case described in section 2.8.1, long words cannot be hyphenated when they are the first word in a margin note. To avoid this, insert 0pt horizontal space before the word.

\textbf{Note:} Margin notes can normally not be used inside tables, floats, and footnotes.

This restriction can be evaded by using the \LaTeX-package \texttt{marginnote}. By adding these two lines to your document preamble, the command used by \LaTeX for margin notes is redefined to use the command provided by the \texttt{marginnote}-package:

$$\texttt{\usepackage\{marginnote\}}$$

$$\texttt{\let\marginpar\marginnote}$$

334
This is also used in this document because \texttt{marginnote} has another useful feature: You can set a vertical offset for the note. This is often needed when too many margin notes are too close together or for a better page layout. The offset is set in \LaTeX as \TeX-Code directly behind the margin note in the scheme

\[\text{offset}\]

where the offset is a length with one of the units listed in Table A.1. A negative value shifts the note up, a positive value shifts it down. For example the margin note beside this text line is shifted up 1.5 cm with the \TeX-Code-command \texttt{“[-1.5cm]”}

With \texttt{marginnote} you can also change the alignment of the text in the margin note. For example the commands

\begin{verbatim}
\renewcommand*{\raggedleftmarginnote}{\centering}
\renewcommand*{\raggedrightmarginnote}{\centering}
\end{verbatim}

set the alignment to centered. \texttt{\raggedleftmarginnote} denotes margin notes that appear at the left side. The default is

\begin{verbatim}
\renewcommand*{\raggedleftmarginnote}{\raggedleft}
\renewcommand*{\raggedrightmarginnote}{\raggedright}
\end{verbatim}

For the other features of \texttt{marginnote} we refer to its documentation [13].

You can adjust the layout of margin notes by changing its definition. To create for example a header for all margin notes with the underlined, sans-serif, and bold header text \texttt{“Attention!”}, add this to your document preamble:

\begin{verbatim}
\let\myMarginpar\marginpar
\renewcommand{\marginpar}{\myMarginpar%\hspace{0pt}\textsf{\textbf{\underline{Attention!}}}}%
\vspace{1.5mm}\%\#1}
\end{verbatim}

This is a margin note with a defined heading.
4. Notes
5. Boxes

5.1. Introduction

Boxes are used to format a block of text. Boxes can be used to write documents with multiple languages, see section 5.4, to frame texts, see section 5.2.3, to prevent words to be hyphenated, see section 5.6.1, to align text, see section 5.6.2, or to set the background color of texts, see section 5.7.

Boxes can be inserted with the menu Insert $\Rightarrow$ Box or the toolbar button $\Rightarrow$. A grey box with the label Box (Minipage): $\Rightarrow$ will be inserted. The box type can be specified by right-clicking on the box. The appearing box dialog offers the Inner Box types Parbox and Minipage. The type Minipage is the default for new boxes and is explained in section 5.4; the type Parbox is described in section 5.5.

Boxes aren’t numbered and can therefore not be referenced like floats or footnotes.

**Note:** Boxes must not be the item in an Itemize or Description environment.

**Note:** For an unknown reason you can only set the Inner Box type to None when you use a framed box. Boxes without an Inner Box type and without frames are explained in section 5.6.1.

5.2. Box Dialog

5.2.1. Size

In the box dialog you can adjust the box geometry in the fields Width and Height. The available units for the geometry are explained in Table A.1. The field Height offers the following additional sizes:

**Depth** This is the plain text “height”. It ignores the total depth when there are multiple text lines in the box:

```
<table>
<thead>
<tr>
<th>Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>height set to</td>
</tr>
<tr>
<td>1 Depth</td>
</tr>
</tbody>
</table>
```
5. Boxes

**Height** This is the height of the text that is inside the box. A value of e.g. 2 for this size will set the box height to 2 times the text height:

![Box height set to 2 Height](image)

**Total Height** This is the Height + Depth:

![Box height set to 1 Total Height](image)

**Width** This sets the width of the box as height:

![Box height set to 1 Width](image)

### 5.2.2. Alignment

When you have chosen an *Inner Box*, the vertical box alignment can be:

**Top** This is an example text line. This is an example text line.

![This box is top-aligned.](image)

**Middle** This is an example text line. This is an example text line.

![This box is middle-aligned.](image)

**Bottom** This is an example text line. This is an example text line.

![This box is bottom-aligned.](image)

**Note:** The vertical box alignment can be lost in the output when you have two boxes in a line and one has e.g. a shadow and the other one not.

The horizontal box alignment can be set via \LaTeX{}’s paragraph dialog when you set the box into its own paragraph.

When you have chosen an *Inner Box*, the box content can be vertical aligned to:

**top** This is an example text line. This is an example text line.

![This box text is top-aligned.](image)
5.2. Box Dialog

**middle** This is an example text line.

This box text is middle-aligned.

This is an example text line.

**bottom** This is an example text line.

This box text is bottom-aligned.

This is an example text line.

**stretch** This is an example text line.

This box text is stretched.

This is an example text line.

To stretch the box content, it must consist of more than one paragraph. In the example above every text line is in an own paragraph.

To align the box content horizontally you can use \LaTeX’s paragraph dialog when you have chosen an *Inner Box*.

This box text is horizontally stretched.

If you haven’t set an *Inner Box*, you can align the box content horizontally in the box dialog.

5.2.3. Decoration

The type of the box can be specified in the box-dialog in the drop-down list *Decoration*. The following types are possible:

**Simple rectangular frame** This draws a rectangle frame around the box. The frame line thickness has the size of \fboxrule. 

**Rectangular box**
allow page break} When you use the decoration simple rectangular frame and no inner box, you can allow page breaks within a box. Note that then, in contrary to other framed boxes, the frame always uses the whole column width, the box is set as its own paragraph, and \fboxrule and \fboxsep has no effect on this box type. The frame line thickness has the size of \FrameRule.

**Oval box, thin** This draws an oval frame around the box. The frame line thickness has the size 0.4 pt.

**Oval box, thick** This draws an oval frame around the box. The frame line thickness has the size 0.8 pt.

**Drop shadow** This draws a rectangle frame with a shadow around the box. The frame line thickness has the size of \fboxrule, the shadow has a width of 4 pt.

**Shaded background** This draws a box with red background color. In contrary to colored boxes\textsuperscript{1}, it always uses the whole column width and the box is set as its own paragraph.

**Double rectangular frame** This draws a double-line rectangle frame around the box. The line thickness of the inner frame is 0.75 \fboxrule, the thickness of the outer frame is 1.5 \fboxrule. The distance between the lines is 1.5 \fboxrule + 0.5 pt.

\textsc{LXe}'s box label will reflect the different frame types. To be able to use all types, the \LaTeX-package fancybox must be installed.

### 5.3. Box customization

The default value for the size \fboxrule is 0.4 pt. It can be changed with the following command in \TeX-Code to e.g. 2 pt:

\begin{verbatim}
\setlength{\fboxrule}{2pt}
\end{verbatim}

\textsc{LXe}'s box label will reflect the different frame types. To be able to use all types, the \LaTeX-package fancybox must be installed.

\begin{verbatim}
\setlength{\fboxrule}{2pt}
\end{verbatim}

\textsuperscript{1}see sec. 5.7

340
The space between the frame and the box content is for all frame styles by default 3 pt. You can change it by setting the length \texttt{\fboxsep} to another value. For example the command
\begin{verbatim}
\setlength{\fboxsep}{10pt}
\end{verbatim}
sets the value to 10 pt, like for the following box:

\begin{verbatim}
Rectangular box
with
\fboxsep = 10 pt
\end{verbatim}

The diameter of the round corners of the oval boxes can be set with the command \texttt{\cornersize}. The command
\begin{verbatim}
\cornersize*{1cm}
\end{verbatim}
sets the diameter to 1 cm. The command
\begin{verbatim}
\cornersize{num}
\end{verbatim}
sets the diameter to \(num \times \text{minimum(width and height of box)}\). The default is \texttt{\cornersize{0.5}}.

\begin{verbatim}
Oval box with \cornersize = 1.5 cm
\end{verbatim}

The size of the shadow can be adjusted by changing the length \texttt{\shadowsize}. It is set to 2 pt for the following box by this command:
\begin{verbatim}
\setlength{\shadowsize}{2pt}
\end{verbatim}

\begin{verbatim}
Shadow box with \shadowsize = 2 pt
\end{verbatim}

The default value for the size \texttt{\FrameRule} is 0.4 pt. The default space between the note content and the frame is 9 pt and can be changed with the value of \texttt{\FrameSep}. For example the frame appearance of the following box is set with the \TeX-Code commands
\begin{verbatim}
\setlength{\FrameRule}{5pt}
\setlength{\FrameSep}{0.5cm}
\end{verbatim}

\begin{verbatim}
This is text in an allow page break box.
\end{verbatim}
5. Boxes

For shaded background boxes the default space between the box content and the box border is 3 pt and can be changed with the value of \texttt{\fboxsep}. The default background color red can either be changed locally with the command \texttt{\definecolor{shadebox}} or globally with the menu \texttt{Tools} \texttt{Preferences} \texttt{Colors} \texttt{shaded box}. The scheme of the \texttt{\definecolor} command is explained in section 2.11.\footnote{Note that \texttt{\definecolor} requires the \LaTeX-package \texttt{color} in the preamble, see section 5.7.} For example the appearance of the following shaded background note is set with the \LaTeX-Code commands

\begin{verbatim}
\setlength{\fboxsep}{0.5cm}
\definecolor{shadecolor}{cmyk}{0.5,0,1,0.5}
\end{verbatim}

This is yellow text in a shaded background box with dark green background.

Changed lengths and widths are valid for all boxes following the commands that change them.

5.4. Minipages

Minipages are treated by \LaTeX as pages within pages and can therefore for example have their own footnotes.

Minipages are useful when you write documents with different languages.

Below are two example minipages side by side. Their width is set to 45\texttt{col} and they are separated by a horizontal fill, that was inserted via the menu \texttt{Insert} \texttt{Special Formatting} \texttt{Horizontal Fill}.

\begin{verbatim}
\end{verbatim}

\footnote{Dies ist eine deutsche Fußnote.}

This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text. This is an English Text.

\footnote{This is an English footnote.}

Another application for minipages are footnotes within tables. Due to a \TeX restriction footnotes within tables doesn’t appear at the bottom of the current page. But when you put the table with the footnote to a minipage, the footnote will appear at its bottom, numbered with Latin letters. The footnote number is reset to 1 in every minipage but not outside the minipages.

The footnote of this table doesn’t appear:

\begin{table}
\begin{tabular}{cccc}
1 & 2 & 3 & 4 \\
a & b & c & d \\
e & f & g & h
\end{tabular}
\end{table}

This is a footnote within a table.

\begin{table}
\begin{tabular}{cccc}
1 & 2 & 3 & 4 \\
a & b & c & d \\
e & f & g & h
\end{tabular}
\end{table}

\footnote{This is a footnote within a table.}

The document-wide paragraph settings are ignored within minipages. That means that there will be no space between paragraphs in minipages although you set it to e.g. \texttt{MedSkip} in the document settings.

Minipages can also be used to set a background color for text parts, see section 5.7.2.

\textbf{Note:} You cannot have floats or margin notes inside minipages but minipages can be used inside tables, floats, and other boxes.
5.5. Parboxes

Parboxes are very similar to minipages with the difference that they cannot have footnotes. The main difference to minipages is that minipages are in contrary to parboxes no real boxes but $\LaTeX$-environments.

This a text within a parbox.
This a text within a parbox.
This footnote won’t appear: $^4$

5.6. Boxes for Words and Characters

5.6.1. Prevent Hyphenation

You can use a special kind of boxes to prevent words or text to be hyphenated. Here is an example text:

This line is an example to show how you can prevent the hyphenation of “veryverylongword”.

To prevent the hyphenation of the word “veryverylongword”, add the command
\mbox{

in $\LaTeX$-Code before the word. Behind the word insert a closing brace “}” in $\LaTeX$-Code.

This is the result:

This line is an example to show how you can prevent the hyphenation of “veryverylongword”.

You can alternatively set the command “\-” as $\LaTeX$-Code directly before the word:

This line is an example to show how you can prevent the hyphenation of “veryverylongword”.

Of course the word now protrudes over the side margin. To avoid this, add via the menu Insert $\Rightarrow$ Special Formatting $\Rightarrow$ Line Break (shortcut Ctrl+Return) a line break before the word:

This line is an example to show how you can prevent the hyphenation of “veryverylongword”.

5.6.2. Vertical Alignment

With the help of the command $\texttt{\raisebox}$ you can align words, characters or other boxes vertically to the surrounding text. $\texttt{\raisebox}$ is used with the following scheme:
\raisebox{\depth}{\depth}\{box content\}

The lift can be a positive value to raise the box or a negative value to lower the box. To align for example the word “preventing” so that the bottom of the “deepest” character “p” is at the baseline, insert the command

\raisebox{\depth}\{ in \TeX-Code before the word. Behind the word insert a closing brace “}” in \TeX-Code.

This is the result:

This is a text line with the word “preventing” as raised word.

When you raise or lower characters in a line, the line distance will be spread:

This is a text line with the word “preventing” as lowered word.

This is a text line with the word “testing” as raised word.

If you want to prevent this for a certain reason, set the box height to a zero value. For example use

\raisebox{-\depth}[0pt]\{

This is a text line with the word “testing” as lowered word.

This is a text line with the word “preventing” as raised word.

5.7. Colored Boxes

5.7.1. Color for Text

To color the background of text the text must be put into a so called “colorbox”. This requires that the \LaTeX-package \texttt{color} is loaded in the document preamble with the command

\@ifundefined{\textcolor}{\usepackage{color}}{}

The package \texttt{color} will be loaded automatically by \LaTeX when you color text\textsuperscript{5}.

Colorboxes are created with the command \texttt{\colorbox}. This will be used with the following scheme:

\colorbox{color}\{box content\}

The box content can also be a box and colorboxes can also be within other boxes.

\textsuperscript{5}To avoid that it is loaded twice the command \texttt{\@ifundefined} is used.
The following colors are predefined: black, blue, cyan, green, magenta, red, white, and yellow.
You can also define your own color as described in section 2.11.

To have e.g. a red background for a word, insert the command
\colorbox{red}{
before the word in \TeX-Code. Behind the word insert a closing brace “}” in \TeX-Code.
This is the result:
This is a line where the word “Attention!” has a red background.

If you would have the box frame in a different color, you can use the command \fcolorbox with the following scheme:
\fcolorbox{frame color}{box color}{box content}
\fcolorbox is an extension to \colorbox. The frame thickness and the space between the frame and the box content can be adjusted with the lengths \fboxrule and \fboxsep, respectively, as described in section 5.2.3.

For the following example the command
\fcolorbox{cyan}{magenta}{
was used.

Here is an example where the frame line thickness was set to 1 mm:
This is text within a colored, framed box.

Of course you can also have colored text inside a colorbox:
This is colored text within a colored, framed box.

Note: Text in colorboxes cannot have line breaks. To color multiple text lines or paragraphs, use a box inside a colorbox as described in the following.

5.7.2. Color for Paragraphs

To set the background color for more than one text line, put the text into a minipage. Before the minipage insert the \colorbox command
\colorbox{color}{
in \TeX-Code. Behind the minipage insert a closing brace “}” in \TeX-Code.
5.8. Rotated and Scaled Boxes

To use the the commands described in this section, the \texttt{\LaTeX}-package \texttt{graphicx} needs to be loaded in the document preamble with the command

\begin{verbatim}
@ifundefined{rotatebox}
  \{\usepackage{graphicx}\}\}
\end{verbatim}

\textbf{Note:} Some DVI-viewers can't display rotated or scaled material.

\textbf{Note:} Floats mustn't be inside a rotated or scaled box.

5.8.1. Rotated Boxes

To rotate material, you can put it into a rotated box. Such a box is created using the command \texttt{\rotatebox} in \LaTeX-Code with the following scheme:

\texttt{\rotatebox[rotation origin]{rotation angle}{box content}}

The rotation origin is specified in the form \texttt{origin=position}. The following positions are possible: \texttt{c} (center), \texttt{l} (left), \texttt{r} (right), \texttt{b} (bottom), \texttt{t} (top), and also expedient combinations of the four base positions. For example \texttt{lt} means, that the rotation origin is at the top left corner of the box. When no rotation origin is specified, the position \texttt{l} will be used. The rotation angle is a number that can be negative that specifies the angle in degrees. The rotation direction is counterclockwise.

In the following example the command \texttt{\rotatebox[origin=c]{60}{}} \ was inserted as \LaTeX-Code before the text “with rotated”, after the text the box was closed by a closing brace } in \LaTeX-Code.

This is a line with rotated text.

The box content can also be another box or an inline formula:
5. Boxes

This is a line of text and a formula.

or an image or table:

\[ \int A \, dx = B \]

This is a line of text with a rotated image and table.

5.8.2. Scaled Boxes

To scale material the commands \texttt{\textbackslash scalebox} and \texttt{\textbackslash resizebox} can be used as \TeX-Code.

\texttt{\textbackslash scalebox} is used with the following scheme:

\texttt{\textbackslash scalebox\{horizontal\}[vertical\}{box content}}

Horizontal and vertical are the corresponding scaling factors. If no vertical scaling factor is given, the horizontal factor will also be used as vertical one.

The command \texttt{\textbackslash scalebox\{2\}{Hello}} creates for example a double size \texttt{Hello}, compared to the document text size.

\texttt{\textbackslash scalebox\{2\}[1\}{Hello}} in contrary distorts the \texttt{Hello}.

If a the scaling factor is negative, the box content will be mirrored. Therefore the command \texttt{\textbackslash scalebox\{-1\}[1\}{Hello}} can be used to create mirror writing: \texttt{olleH}

\texttt{\textbackslash scalebox\{1\}[-1\}{Hello}} reflects the \texttt{Hello} at the base line.

Equivalent to \texttt{\textbackslash scalebox\{-1\}[1\}{box content}} there exists the command \texttt{\textbackslash reflectbox\{box content\}}.

The command \texttt{\textbackslash resizebox} is used to scale the box to a defined width and height. The command scheme is:

\texttt{\textbackslash resizebox\{width\}\{height\}\{box content\}}
5.8. Rotated and Scaled Boxes

Is one of the two command arguments given as exclamation mark !, the size is set so that the aspect ratio of the box content is kept.

The command \resizebox{2cm}{1cm}{Hello} produces: \text{Hello}
The command \resizebox{2cm}{!}{Hello} produces: \text{Hello}

\textbf{Note:} When arguments of \texttt{\scalebox} or \texttt{\resizebox} are set to zero, no \LaTeX- errors occur when exporting the document but the exported files can not or only partly be displayed.

The boxes can be combined in any order. E.g. the command \rotatebox[origin=c]{-45}{\resizebox{2cm}{!}{\reflectbox{Hello}}}
produces:

Hello

Images, tables, and inline formulas are allowed as box content:

When the global formula style \texttt{fleqn} is used in the document\footnote{When “fleqn” is added to the document class options.}, also display formulas can be scaled.

\footnote{When “fleqn” is added to the document class options.}
5. Boxes
6. External Document Parts

With the menu Insert▷File you can insert external material to your document. This can be:

- **LyX Document**: Another LyX document; its content is directly inserted to your document.
- **Plain Text**: A text document; every of its text lines is inserted to your document as own paragraph.
- **Plain Text, Join Lines**: A text document; its text lines are inserted as they are. Empty text lines creates a new paragraph in your document.
- **External Material**: Files in various formats.
- **Child Document**: LyX or \LaTeX-documents.

### 6.1. External Material

The external material feature allows you to insert files to your document without converting them previously to a format that can be read by the document output format because LyX takes care of needed conversions. This is similar to images that can be inserted in various image formats to LyX documents. When you have enabled Instant Preview in LyX’s preferences under Look and feel▷Graphics, the external material types Dia and Xfig is directly shown in LyX.

External material can be inserted via the External Material dialog that is accessible with the menu Insert▷File▷External Material. Currently the following file types (Templates) are allowed:

- **ChessDiagram**: This template supports chess position diagrams made with the program XBoard.
- **Date**: This inserts the date in the form Day-Month-Year. This is a date inserted as external material: 17-08-2009. The date is not shown within LyX, only in the output. There are two other methods of inserting a date: Via menu Insert▷Date and with the \LaTeX command \today as \LaTeX-Code. The different methods are compared in Table 6.1.
- **Dia**: This template supports diagrams created with the program Dia.
6. External Document Parts

LilyPond This template is used for music notation typeset with the program LilyPond.

PDFPages With this template you can insert PDF documents to your document. To insert certain or all pages of a PDF, use the pages option in the Option field in the \LaTeX{} and \texttt{lyx} options tab according to the template description in the dialog. When no pages option is given, only the first page of the PDF will be inserted.

RasterImage This can be used for bitmap images. Nearly all popular image formats are supported. The image can be treated in the External material dialog like the images that are usually included via the Graphics dialog as described in section 1.1. The difference is that only raster images are allowed, that means that PDF and EPS-images are not supported.

XFig This template supports images created with the program Xfig.

<table>
<thead>
<tr>
<th>Document format</th>
<th>External Material</th>
<th>Insert</th>
<th>Date</th>
<th>command $\today$</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{LyX}</td>
<td>as inset box</td>
<td>as date</td>
<td>as T	exttt{\LaTeX}-Code inset box</td>
<td></td>
</tr>
<tr>
<td>\texttt{B\LaTeX}</td>
<td>as date</td>
<td>as date</td>
<td>as command</td>
<td></td>
</tr>
<tr>
<td>DVI, PDF, PS</td>
<td>as date</td>
<td>as date</td>
<td>as date</td>
<td></td>
</tr>
</tbody>
</table>

When you use the option Draft in the File tab of the External Material dialog, only the path to the inserted file is shown in the output.

External material is displayed in \texttt{LyX} either as box like this: 

\begin{tikzpicture}
\node[draw, thick, fill=cyan] at (0,0) {catalan};
\end{tikzpicture}

or as image, depending on the option Show in \texttt{LyX} in the \LaTeX{} and \texttt{lyx} options tab of the dialog.

The Customization manual explains how you can define your own templates.

6.2. Child Documents

Child documents are used when you have a long document consisting of several larger parts or sections. For maintenance it is often useful and sometimes even required to split the document in several files that can be revised separately. The different documents are then the so called child documents and a master document connects them to print the full document or parts of it. A child documents inherits contents of its master, for example the \LaTeX{} preamble, the bibliography, and labels for cross-references.

To be able to work on child documents without the need to open its master, specify in the child document the master in the menu Document >> Settings >> Document Class.
This master document will then be used in the background by \LaTeX when you edit the child document.

Included documents are displayed in \LaTeX as a box like this: Include. 
To include child documents to a master document use the menu Insert $\Rightarrow$ File $\Rightarrow$ Child Documents. A dialog pops up where you can choose between four include methods:

**Include** You can include \LaTeX and \TeX -documents. When you press the Load button in the Child Document dialog, the included documents will be opened in \LaTeX in a new file tab so that you can modify it.

Here is a child document inserted using **Include**:
6. External Document Parts

6.2.1. External Subsection 1

This is a small dummy child document to show how files can be inserted to another document.
The section numbering includes the sections of the included files in the order they are inserted in the master document. The included example document has for example a subsection that is numbered as subsection of this section. Labels of included documents can be referenced: Subsection 6.2.1

The preamble of the child document is ignored, only the preamble of the master document is used. Branches in child documents will be ignored by the master document when the master document does not have a branch with the same name. Included documents are inserted starting on a new page and end with a page break.

With the \texttt{\LaTeX}-command \texttt{\includeonly} you can specify which included child documents are processed when the output is generated. This is useful when you are perhaps only working on a certain chapter of your large document as this saves compiling time. \texttt{\includeonly} is inserted to the master document preamble. It takes a comma-separated list of the filenames as argument, e.g.

\texttt{\includeonly{chapter1,chapter5}}

will only process the included files named “chapter1.lyx” (or “chapter1.tex”) and “chapter5.lyx”.

\textbf{Note:} When you have included a \texttt{LYX}- or \texttt{\LaTeX}-file, you are warned when you export/view the document in case that the child document uses another document class than the master document as this will lead to unexpected outputs.

\textbf{Input} This method is very similar to the Include method. The differences are:

- Input files don’t start with a new page and don’t end with a page break.
- Input files can be previewed in \texttt{LYX} when \texttt{Instant Preview} is enabled in \texttt{LYX}’s preferences under \texttt{Look and feel} $\rightarrow$ \texttt{Graphics}.
- The \texttt{\LaTeX}-command \texttt{\includeonly} cannot be used.

Here is a child document inserted using \texttt{Input}:

\textbf{6.2.2. External Subsection 2}

This is another small dummy child document to show how files can be inserted to a document.

\textbf{Verbatim} With this method every text file can be included. The file is shown in the output with its source code, no command used in the text is invoked. You can use the option \texttt{Mark spaces in output} that displays the character “␣” for every space character in the source code. The difference to the method via the menu \texttt{Insert} $\rightarrow$ \texttt{File} $\rightarrow$ \texttt{Plain Text} is that the document content is not shown in \texttt{LYX}.

Here is a child document inserted as Verbatim:

This is a small dummy child text document to show how files can be inserted to a document.
6. *External Document Parts*

Here is a child document inserted as Verbatim using the **Mark spaces in output** option:

This is a small dummy child text document to show how files can be inserted to a document.

**Note:** As you can see in the examples above, the text of the documents included as *verbatim* is not broken at the end of the document lines.

**Listings** This type is described in chapter 7.

**Note:** Including the same document twice in a document using different methods could cause \LaTeX-problems.
7. Program Code Listings

To include and typeset program code you can use the Listings inset that can be inserted via the menu Insert > Program Listing. The \LaTeX-package listings provides a powerful and flexible way to insert program source code to your document.

Right-clicking on a listings inset opens the context menu containing where you can set the listings format.

By default, a listing starts a new paragraph in the output. The placement option Inline listing prints the listing inline like this: \texttt{int a=5;}
The option Float creates a listings float where you can specify the placement options “h”, “t”, “b”, and “p” corresponding to the float placement options described in section 3.5. The placement options can be mixed and are inserted without any separation, e.g. “htbp”. The placement “h” has sometimes no effect, but you don’t need to use the Float option in this case as also non-float listings can have captions and be referenced.

You can add captions to listings with the menu Insert > Caption. Listings can be referenced like floats: Listing 7.1

**Listing 7.1:** Example Listing float

```python
# Example listing float
def func(param):
    'this is a python function'
    pass
```

When you have set a programming language in the listings dialog, the keywords of this language will be recognized and specially typeset in the output. In the example listings the Python keyword “def” is recognized and printed bold in the output.

**Note:** If you don’t get bold keywords when using typewriter fonts, your typewriter font probably doesn’t provide a bold shape. In this case select a different one in the menu Document > Settings > Fonts. (The fonts LuxiMono, BeraMono and Courier provide bold shapes.)

In section Line numbering of the listings dialog you can specify the line numbering style. You can insert a number to specify which lines are numbered to the field Step. When you insert e.g. “3”, only every 3rd line will be numbered.

You can furthermore specify a range of lines, only these will then appear in the output. The option Extended character table should be used when you use national
7. Program Code Listings

characters like the German umlauts in the listing.
Here is an example listing with left line numbering, step “3”, language “Python”,
options “Extended character table” and “Space as symbol”, range lines 3-8:

```python
    pass
2   def func(param):
      'This is a German word: Tschüß'
    pass
5   def func(param):
      'this is a python function'
```

When you have tabulators in your listing, you can specify the number of characters
that are spanned by a tabulator in the field **Tabulator size**.

**Note:** Due to a bug in the *listings* package the line numbering is shifted by a line
by every previous listing. That’s the reason why the lines 2 and 5 are numbered in
the above listing and not the lines 3 and 6.

It is also possible to print lines from a file as listing. To do this, use the menu **Insert»File»Child Document** and choose the type **Listings**. In the child document dialog you can specify the listing parameters in a text box. To show a list of all available parameters, type in a question mark “?” in the text box.

To reference child document listings, write a label text into the corresponding field of the child document dialog. The label can then be referenced using the menu **Insert»Cross-Reference**.

Listing 7.2 is an example for a listing of a file; there the lines 10-15 of this LyX file are listed.

**Listing 7.2:** Lines 10 - 15 of this LyX file

```latex
\% set fonts for nicer pdf view
\IfFileExists{lmodern . sty}
   {\usepackage{lmodern}}{}
\fi \% end if pdflatex is used
```

Global listings settings can be set in the **Document»Settings»Text Layout** dialog. To get there a list of available options, type in a question mark “?” in the **Listings settings** field.

For more information about the *listings* package, we refer to its documentation [12].

---

1The other child document types are described in section 6.2.
A. Units available in LyX

To understand the units described in this documentation, Table A.1 explains all units available in LyX.

Table A.1: Units

<table>
<thead>
<tr>
<th>unit</th>
<th>name/description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm</td>
<td>millimeter</td>
</tr>
<tr>
<td>cm</td>
<td>centimeter</td>
</tr>
<tr>
<td>in</td>
<td>inch</td>
</tr>
<tr>
<td>pt</td>
<td>point (72.27 pt = 1 in)</td>
</tr>
<tr>
<td>pc</td>
<td>pica (1 pc = 12 pt)</td>
</tr>
<tr>
<td>sp</td>
<td>scaled point (65536 sp = 1 pt)</td>
</tr>
<tr>
<td>bp</td>
<td>big point (72 bp = 1 in)</td>
</tr>
<tr>
<td>dd</td>
<td>didot (1 dd ≈ 0.376 mm)</td>
</tr>
<tr>
<td>cc</td>
<td>cicero (1 cc = 12 dd)</td>
</tr>
<tr>
<td>Scale%</td>
<td>% of original image width</td>
</tr>
<tr>
<td>text%</td>
<td>% of text width</td>
</tr>
<tr>
<td>col%</td>
<td>% of column width</td>
</tr>
<tr>
<td>page%</td>
<td>% of paper width</td>
</tr>
<tr>
<td>line%</td>
<td>% of line width</td>
</tr>
<tr>
<td>theight%</td>
<td>% of text height</td>
</tr>
<tr>
<td>pheight%</td>
<td>% of paper height</td>
</tr>
<tr>
<td>ex</td>
<td>height of letter $x$ in current font</td>
</tr>
<tr>
<td>em</td>
<td>width of letter $M$ in current font</td>
</tr>
<tr>
<td>mu</td>
<td>math unit (1 mu = 1/18 em)</td>
</tr>
</tbody>
</table>
A. Units available in LyX
B. Output File Formats with Graphics

B.1. DVI

This file type has the extension “.dvi”. It is called “device-independent” (DVI), because it is completely portable; you can move them from one machine to another without needing to do any sort of conversion. At the time when this file-format was developed, this was no matter of course. DVIs are used for quick previews and as pre-stage for other output formats, like PostScript.

Note: DVI-files doesn’t contain images, they will only be a linked. So don’t forget this, if you move your .dvi file to another computer. This property can also slow down your computer when you view the DVI. Because the DVI-viewer has to convert the image in the background to make it visible when you scroll in the DVI. So we recommend to use PDF for files with many images.

You can export your document to DVI by using the menu File > Export > DVI. You can view your document as DVI via the View menu or by using the toolbar button .

B.2. PostScript

This file type has the extension “.ps”. PostScript was developed by the company Adobe as printer language. The file contains therefore commands that the printer uses to print the file. PostScript can be seen as “programming language”; you can calculate with it and draw diagrams and images . Due to this ability, the files are often bigger than PDFs.

PostScript can only contain images in the format “Encapsulated PostScript” (EPS, file extension “.eps”). As \LaTeX{} allows you to use any known image format in your document, it has to convert images in the background to EPS. If you have e.g 50 images in your document, \LaTeX{} has to do 50 conversions whenever you view or export your document. This will slow down your work flow with \LaTeX{} drastically. So if you plan to use PostScript, you can insert your images directly as EPS to avoid this problem.

\[\text{If you are interested to learn more about this, have a look at the \LaTeX{}-package PSTricks [14].}\]
B. Output File Formats with Graphics

You can export your document to PostScript using the menu File $\rightarrow$ Export $\rightarrow$ Postscript. You can view your document as PostScript via the View menu or by using the toolbar button $\mathbb{F}_S$.

B.3. PDF

This file type has the extension “.pdf”. The “Portable Document Format” (PDF) is developed by Adobe as derivative from PostScript. It is more compressed and it uses much less commands than PostScript. As the name “portable” implies, it can be processed at any computer system and the printed output looks exactly the same.

PDF can contain images in its own PDF format, in the format “Joint Photographic Experts Group” (JPG, file extension “.jpg” or “.jpeg”), and in the format “Portable Network Graphics” (PNG, file extension “.png”). Nevertheless you can use any other image format, because \LaTeX{} converts them in the background to one of these formats. But as described in the section about PostScript, the image conversion will slow down your work flow. So it is recommended to use images in one of the three mentioned formats.

You can export your document to PDF via the menu File $\rightarrow$ Export in three different ways:

**PDF (ps2pdf)** This uses the program ps2pdf that creates a PDF from a PostScript-version of your file. The PostScript-version is produced by the program dvips which uses a DVI-version as intermediate step. So this export variant consist of three conversions.

**PDF (dvipdfm)** This uses the program dvipdfm that converts your file in the background to DVI and in a second step to PDF.

**PDF (pdflatex)** This uses the program pdftex that converts your file directly to PDF.

It is recommended to use PDF (pdflatex) because pdftex supports all features of actual PDF-versions, is quick and works stable without problems. The program dvipdfm is not under development and therefore a bit outdated.

You can view your document as PDF via the View menu or by using the toolbar button $\mathbb{F}_S$ (that uses PDF (pdflatex)).
C. Explanation of Equation (2.1)

The total width of \( n \) table cells \( W_{\text{tot}\, n} \) can be calculated to

\[
W_{\text{tot}\, n} = n \cdot (W_{\text{g}\, n} + 2 \cdot \texttt{\textbackslash tabcolsep}) + (n + 1) \cdot \texttt{\textbackslash arrayrulewidth} \quad (C.1)
\]

Where \( W_{\text{g}\, n} \) is the given width of all cells. \texttt{\textbackslash tabcolsep} is the \LaTeX-length between the cell text and the cell border, its default value is 6 pt. \texttt{\textbackslash arrayrulewidth} is the thickness of the cell border line, the default is 0.4 pt.

Following equation (C.1), the total width of a multicolumn \( W_{\text{tot}\, \text{mult}} \) is

\[
W_{\text{tot}\, \text{mult}} = W_{\text{g}\, \text{mult}} + 2 \cdot \texttt{\textbackslash tabcolsep} + 2 \cdot \texttt{\textbackslash arrayrulewidth} \quad (C.2)
\]

By setting equation (C.1) and (C.2) equal we can calculate the needed given width \( W_{\text{g}\, n} \) when \( n \) columns are spanned, so that each column has a total width of \( W_{\text{tot}\, \text{mult}} / n \):

\[
W_{\text{g}\, n} = (W_{\text{g}\, \text{mult}} + (1 - n) \cdot (2 \cdot \texttt{\textbackslash tabcolsep} + \texttt{\textbackslash arrayrulewidth})) / n \quad (C.3)
\]
C. Explanation of Equation (2.1)
Bibliography


[4] Documentation of the \TeX-package *booktabs*

[5] Documentation of the \TeX-package *caption*

[6] Documentation of the \TeX-package *endfloat*

[7] Documentation of the \TeX-package *wrapfig*

[8] Documentation of the \TeX-package *footmisc*

[9] Documentation of the \TeX-package *hycap*

[10] Documentation of the \TeX-package *hyperref*

[11] Documentation of the \TeX-package *KOMA-script*

[12] Documentation of the \TeX-package *listings*

[13] Documentation of the \TeX-package *marginnote*

[14] Web page of the \TeX-package *PSTricks*

[15] Documentation of the \TeX-package *sidecap*

[16] Wiki page about new features in *\LYX 1.6.0.*
Bibliography
Index

Boxes
   Alignment, 338
   Box Dialog, 337
   Color, 345
   Customization, 340
   Decoration, 339
   for Characters, 344
   for Vertical Alignment, 344
   Introduction, 337
   Minipages, 342
   Parboxes, 344
   Raiseboxes, 344
   rotated, 347
   scaled, 348
   Size, 337
   to Prevent Hyphenation, 344

Caption
   Formatting, 325
   Placement, 326

Color
   for Paragraphs, 346
   for Table Cells, 305
   for Table Lines, 307
   for Text, 345

DVI, see File Formats

Endnotes, 332
EPS, see Image formats
External Document Parts, 351
   Child Documents, 352
   External Material, 351

Figures, 280
   Floats, 282
   Graphics Dialog, 280

Image Formats, 284
   rotated, 280, 347
   scaled, 280, 348
   Settings grouping, 281

File Formats
   DVI, 361
   PDF, 362
   PostScript, 361

Files
   Include, 352

Floats, 315
   Algorithms, 315
   Caption Formatting, 325
   Caption Placement, 326
   Figures, 282
   Float Lists, 327
   Introduction, 315
   Listings, 357
   Numbering, 317
   Placement, 320
   References, 318
   Rotating, 322
   Side by side, 322
   Subfloats, 322
   Tables, 287
   Wrap Floats, 316

Footnotes, 330
   Numbering, 331
   Placement, 332

GIF, see Image formats

Graphics, see Figures

Image Formats, 284

JPG, see Image formats
Index

\LaTeX\-packages

\begin{itemize}
  \item arydshln, 313
  \item booktabs, 304, 365
  \item calc, 301
  \item caption, 288, 292, 293, 325, 365
  \item color, 345
  \item colortbl, 305, 314
  \item dcolumn, 310
  \item endfloat, 321, 365
  \item endnotes, 332
  \item fancybox, 340
  \item footmisc, 334, 365
  \item ftntight, 332
  \item graphicx, 347
  \item hypercap, 320, 327
  \item hyperref, 292, 319, 320, 365
  \item KOMA-script, 325, 326, 365
  \item listings, 357, 365
  \item marginnote, 334, 365
  \item multirow, 301
  \item oberdiek, 320
  \item PSTricks, 365
  \item sidecap, 326, 365
  \item wrapfig, 316, 365
\end{itemize}

Listings, 357

Longtables, 288

\begin{itemize}
  \item Alignment, 291
  \item Calculation, 295
  \item Caption Width, 292
  \item Captions, 291
  \item Different Captions for Pages, 293
  \item Floats, 296
  \item Footnotes, 291
  \item Forced Page Breaks, 296
  \item References, 292
\end{itemize}

Multicolumns, 299

\begin{itemize}
  \item Calculations, 300
  \item Multiple Lines in Table Cells, 299
  \item Multirows, 301
\end{itemize}

Notes

\begin{itemize}
  \item Footnotes, 330
  \item LyX Notes, 329
\end{itemize}

Margin Notes, 334

PDF, 284, 362

PNG, see Image formats

PostScript, see File Formats

Program Code, 357

References

\begin{itemize}
  \item Automatic Reference Naming, 319
  \item Formats, 318
  \item Reference Position, 320
  \item to Figures, 282
  \item to Floats, 318
  \item to Tables, 288
\end{itemize}

Rotated material, 347

Scaled material, 348

SVG, see Image formats

Table, 285

\begin{itemize}
  \item Alignment, 304
  \item Color, 305
  \item Customization, 308
  \item Dialog, 285
  \item Edit Menu, 287
  \item Floats, 287
  \item Formal, 302
  \item Introduction, 285
  \item Linebreaks, 299
  \item Longtables, 288
  \item Multicolumns, 299
  \item Multirows, 301
  \item Toolbar, 286
\end{itemize}

Table Color

\begin{itemize}
  \item for Cells, 305
  \item for Lines, 307
\end{itemize}

Table Customization, 308

\begin{itemize}
  \item Cell/Column Format, 310
  \item Dashed Lines, 313
  \item Line Thickness, 312
  \item Rotating, 347
  \item Row Spacing, 308
  \item Scaling, 348
  \item Special Cell Alignment, 310
\end{itemize}

Units, 359
List of Figures

1.1. A severely distorted platypus in a float. .......................... 282
1.2. M.C. Escher on acid. .............................................. 283
1.3. Two distorted images. Both images are in the image settings group named “distorted”. .............................................. 283

3.1. This is a figure wrap float. .......................................... 316
3.2. Float on the left side. .................................................. 324
3.3. Float on the right side. ............................................... 324
3.4. This is an example figure caption that is longer than one line to show the different caption format. Here a self-defined caption format is used. 328
3.5. This is a caption beside a figure. ............................... 328
3.6. This is a vertically centered caption beside a figure. .......... 328

4.1. Standard footnote placement in two-column documents. ...... 332
4.2. Footnote placement in two-column documents when the \LaTeX-package \texttt{ftnright} is used. ........................................ 333
4.3. Endnotes – footnotes are printed in a separate paragraph at the end of sections or chapters. ........................................ 333
List of Figures
# List of Tables

2.1. A table float. ................................................. 288
2.2. Longtable ..................................................... 291
2.3. Referenced longtable ......................................... 292
2.4. caption with default width .................................. 293
2.5. caption with width = 5 cm ................................... 293
2.6. Example Phone List ............................................ 294
2.7. Table with forced page break in table cell ................ 296
2.8. Table with multiple lines in cells .......................... 299
2.9. Table with and without hyphenation ......................... 299
2.10. Perfect multicolumn table ................................... 300
2.11. Imperfect multicolumn table ................................ 300
2.12. Example booktabs-table ..................................... 303
2.13. Special booktabs-table ...................................... 304
2.14. Table without colortbl ...................................... 305
2.15. Table with colortbl .......................................... 307
2.16. Table with colored vertical lines ......................... 307
2.17. Table with colored horizontal lines ....................... 308
2.18. Table with colored lines .................................... 308
2.19. Vertical alignment of text with large font sizes .......... 309
2.20. Table cells of a column aligned with the decimal separator. 310
2.21. Several table cell alignments ............................... 310
2.22. Alignments when \LaTeX-package dcolumn is used. For all column alignments tricks have to be used to get the output. 311
2.23. Table using user-defined table formats .................... 312
2.24. Table with 1.5 pt thick lines ................................ 312
2.25. Table with 1.5 pt thick horizontal lines ................ 313
2.26. Table with 1.5 pt thick vertical lines .................... 313
2.27. Table with dashed lines ..................................... 313
3.1. Rotated table .................................................. 323
3.2. Two subtables placed side by side ......................... 324
3.3. Two subtables placed one upon the other. (a) table with 4 cells, (b) table with 5 cells. 324
3.4. This is an example table caption that is longer than one line to show the different caption format. Here the standard caption format for tables in this document is used. 328
## List of Tables

3.5. This is a caption beside a table. ........................................... 328  
6.1. Comparison of the date input methods. ................................. 352  
A.1. Units .................................................................................. 359
List of Algorithms

3.1. Example Algorithm float .......................... 316
LyX’s detailed Math manual

by the LyX Team*

Version 1.6.x

August 17, 2009

*If you have comments or error corrections, please send them to the LyX Documentation mailing list: lyx-docs@lists.lyx.org
Contents

1. Introduction 374

2. General Instructions 374

3. Basic Functions 377
   3.1. Exponents and Indices 377
   3.2. Fractions 377
   3.3. Roots 378
   3.4. Binomial Coefficients 379
   3.5. Case Differentiations 379
   3.6. Negations 380
   3.7. Placeholders 380
   3.8. Lines 381
   3.9. Ellipses 381

4. Matrices 382

5. Brackets and Delimiters 384
   5.1. Vertical Brackets and Delimiters 384
      5.1.1. Manual Bracket Size 384
      5.1.2. Automatic Bracket Size 385
   5.2. Horizontal Brackets 386

6. Arrows 387
   6.1. Horizontal Arrows 388
   6.2. Vertical and diagonal Arrows 388

7. Accents 389
   7.1. Accents for one Character 389
   7.2. Accents for Operators 389
   7.3. Accents for several Characters 390

8. Space 391
   8.1. Predefined Space 391
   8.2. Variable Space 392
   8.3. Space besides inline Formulas 392

9. Boxes and Frames 393
   9.1. Boxes with Frame 393
   9.2. Boxes without Frame 394
   9.3. Colored Boxes 395
   9.4. Paragraph Boxes 396
1. Introduction

This document explains LyX’s math features and is furthermore a collection of \LaTeX-commands used for mathematical characters and constructs. The explanations are designed for the usage of commands. It is therefore required that you have read the section Mathematical Formulas of the User’s Guide.

Most of the characters and many constructs explained in this manual are also accessible via the menu Insert $\rightarrow$ Math, or the math toolbar. But everybody who has to write lots of formulas will notice that it is much faster to use commands instead of the math toolbar. Therefore this manual is focused on commands but also mentions the corresponding toolbar buttons when available.

If not specially mentioned the commands are only available within formulas. To be able to use all commands explained in this document, the option Use AMS math package must be used in the document settings (menu Document $\rightarrow$ Settings $\rightarrow$ Math Options).\(^1\)

This document doesn’t list all $\mathcal{AMS}$-math commands\(^2\) for lucidity reasons.

2. General Instructions

To create an inline formula that is embedded into a text line, use one of the shortcuts Ctrl+M, Alt+C M, Alt+M M or the toolbar button $\frac{a+b}{c}$.

To create a display style formula that will appear bigger and in an own paragraph, use one of these shortcuts: Ctrl+Shift+M, Alt+M D.

To change a display style formula to an inline formula, set the cursor into the formula and use one of the shortcuts Ctrl+M, Alt+C M, Alt+M M or the menu Edit $\rightarrow$ Math $\rightarrow$ Change formula type. The same way is used to change an inline formula to a display style formula.

To display parts of an inline formula in the size of a display style formula, enter the command \texttt{\textbackslash displaystyle} to a formula. Then a new blue box appears in which the desired formula part is inserted.

Only inline formulas are allowed inside tables.

The math toolbar can be turned on in the menu View $\rightarrow$ Toolbars. When you click there on “Math” the toolbar will be shown permanently at the bottom; this state is visualized in the Toolbars menu with a checkmark. When you click in this state again

\(^1\)The option Use AMS math package automatically only uses $\mathcal{AMS}$-math when math constructs are found that are supported by LyX.

\(^2\)A list with all $\mathcal{AMS}$-math commands is in the file amsguide.ps, which is part of every \LaTeX standard installation.
on “Math” in the Toolbars menu, the math toolbar is only shown when the cursor is within a formula; this state is visualized by the renaming of the menu entry from “Math” to “Math (auto)”. The \TeX-mode is invoked by pressing the toolbar button $\TeX$ or by using the menu Insert $\Rightarrow$ TeX Code (shortcut Ctrl+L).

To change the $\LaTeX$-preamble, use the menu Document $\Rightarrow$ Settings $\Rightarrow$ LaTeX Preamble.

To edit matrices, case differentiations, and multiline formulas subsequently, the menus Edit $\Rightarrow$ Math and Edit $\Rightarrow$ Rows & Columns, or the table toolbar can be used. When lines and columns are swapped via the menu, the column or line where the cursor is in is exchanged with the column to the right or the line below, respectively. Is the cursor in the last column or row, the exchange is done with the column to the left or the line above.

To write text in formulas\(^3\) mathematical text is used. This mode is invoked with the the shortcut Alt+C Space or by the insertion of the command $\text$. The text appears black in LYX and can therefore be distinguished from the other formula parts that appear blue. In the output mathematical text is set upright, in contrary to other formula parts.

### Command Scheme

Most of the $\LaTeX$-commands for math constructs have the following scheme:

\[\text\textless\text{commandname}\text{[optional argument]}\text{\{}\text{required argument}\text{\}}\]

A command starts always with a backslash “\”. To omit optional arguments, also omit the associated brackets. The braces around the required arguments are named in this document as $\TeX$-braces. If you add in a formula a left brace to a command name, LYX creates automatically a $\TeX$-brace. In all other cases $\TeX$-braces are created in formulas with the command $\text\{\$. $\TeX$-braces appear red in LYX, in contrary to normal braces that appear blue. In $\TeX$-mode no command is needed to get $\TeX$-braces. $\TeX$-braces don’t appear in the output.

When commands without arguments, like commands for symbols are entered in $\TeX$-mode, a space character must always be behind the command to end it. This space doesn’t appear in the output. When the space should appear in the output, the space must be followed by a protected space in normal text.

A protected space is inserted with Ctrl+Space.

\(^3\)For multiline formulas the command $\text\intertext$ is used, see sec.18.7.
Syntax Explanation

- The symbol \( \mathbf{\text{\_\_}} \) denotes a space character to be input.
- An arrow like \( \rightarrow \) denotes the usage of the corresponding arrow key on the keyboard.

Available units

Table 1: Available units

<table>
<thead>
<tr>
<th>Unit</th>
<th>Name / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm</td>
<td>Millimeter</td>
</tr>
<tr>
<td>cm</td>
<td>Centimeter</td>
</tr>
<tr>
<td>in</td>
<td>Inch (1 in = 2.54 cm)</td>
</tr>
<tr>
<td>pt</td>
<td>Point (72.27 pt = 1 in)</td>
</tr>
<tr>
<td>pc</td>
<td>Pica (1 pc = 12 pt)</td>
</tr>
<tr>
<td>sp</td>
<td>scaled point (65536 sp = 1 pt)</td>
</tr>
<tr>
<td>bp</td>
<td>big point (72 bp = 1 in)</td>
</tr>
<tr>
<td>dd</td>
<td>Didot (1 dd ≈ 0.376 mm)</td>
</tr>
<tr>
<td>cc</td>
<td>Cicero (1 cc = 12 dd)</td>
</tr>
<tr>
<td>ex</td>
<td>Height of letter “x” in the current font</td>
</tr>
<tr>
<td>em</td>
<td>width of letter “M” in the current font</td>
</tr>
<tr>
<td>mu</td>
<td>math unit (1 mu = ( \frac{1}{18} ) em)</td>
</tr>
</tbody>
</table>

This visible space character can be created with the command \textvisiblespace, inserted in \TeX-mode.
3. Basic Functions

3.1. Exponents and Indices

Indices are created with an underscore “_” or via the math toolbar button $\text{□}$, exponents with a caret “^” or via the math toolbar button $\text{□}$.

<table>
<thead>
<tr>
<th>command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_V$</td>
<td>$B_V$</td>
</tr>
<tr>
<td>$B^V$</td>
<td>$B^V$</td>
</tr>
<tr>
<td>$B^\alpha A$</td>
<td>$B^A$</td>
</tr>
</tbody>
</table>

As the caret is in some languages an accent, vowels will be accentuated in this case and not set as exponents\(^5\). To get in this case exponents, press \text{Space} after the caret as in the last example.

3.2. Fractions

Fractions are generated with the command $\frac{}{}$ or via the math toolbar button $\text{□}$. The font size is adjusted automatically, depending on whether the fraction is in an inline or display style formula. With the math toolbar button $\text{□}$ you can select different fraction types.

With the command $\dfrac{}{}$ a fraction can be created that has in any case the size of a display style formula. With $\tfrac{}{}$ the fraction appears always with the size of an inline formula. An example:

A line with the fraction $\frac{1}{2}$ that was created with the command $\frac{}{}$.

A line with the fraction $\frac{1}{2}$ that was created with the command $\dfrac{}{}$.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{A}{B}$</td>
<td>$A \div B$</td>
</tr>
<tr>
<td>$\dfrac{A}{B}$</td>
<td>$A \div B$</td>
</tr>
<tr>
<td>$\dfrac{e^2}{3}$</td>
<td>$e^2 \div 3$</td>
</tr>
</tbody>
</table>

\(^5\)Depending on the used keyboard settings this can also happen for other characters than vowels.
For nested fractions the command \cfrac can be used. Here an example:

\[
\begin{array}{c}
\cfrac{A}{B + \cfrac{C + E}{D}} \\
\frac{A}{B + \cfrac{C + E}{D}}
\end{array}
\]

The command for the example above is:

\cfrac\cfrac\cfrac\cfrac

\cfrac sets the fraction always in the size of a displayed formula, also when it is part of another fraction.

It is possible to specify the alignment of the numerator. The command \cfracleft is used to left align it, the command \cfracright to right-align it. \cfrac centers the numerator. These fractions demonstrate the different alignments:

\[
\begin{array}{c}
\cfracleft A \cfracleft B + \cfracright C + \cfracleft E \\
\cfracleft A \cfracleft B + \cfracright C + \cfracright F \\
\cfracleft A \cfracleft B + \cfracleft C
\end{array}
\]

Note: \cfracleft and \cfracright are no real LATEX commands but represent the command \cfrac[alignment]{numerator}{denominator}. Therefore you cannot use them in TEX code.

It is often advantageous to combine \cfrac and \frac:

\[
\frac{A}{B + \frac{C + E}{D}}
\]

For inline fractions with a sloped fraction stroke you can use the command \nicefrac: 5/31 or \unitfrac: 5/31. There is furthermore the command \unitfracthree that offers to write a fraction in combination with a number: 2 \frac{1}{3}

Note: \unitfracthree is not a real LATEX command but the command \unitfrac[number]{numerator}{denominator} . Therefore you cannot use it in TEX code.

How to define own fractions where the fraction stroke can be changed, is explained in sec. 23.4.

3.3. Roots

Square roots are created with \sqrt or the math toolbar button √, all other roots with the command \root or with the math toolbar button √.
A square root can also be created with \( \sqrt{ } \) when the root index field is left empty.

With certain indices the distance to the root is too small, like in this formula: \( \sqrt[\beta]{B} \)

The \( \beta \) touches the root. To avoid this, the commands \( \leftroot \) and \( \uproot \) are used with the following scheme:

\[ \leftroot{distance} \text{ and } \uproot{distance} \]

Distance is the number of Big Points (unit bp; 72 bp = 1 inch), that the index should be moved to the left or top, resp.. The commands are written to the index. This way the command

\( \sqrt[\leftroot{-1} \uproot{2}]{\beta} \) \quad \text{produces a correct typeset formula:} \quad \sqrt[\beta]{B} \)

### 3.4. Binomial Coefficients

Binomial coefficients are inserted with the command \( \binom \) or with the submenu of the math toolbar button \( \binom \). Analog to fractions (\( \frac{A}{B} \)) there are besides \( \binom \) the commands \( \dbinom \) and \( \tbinom \). For other brackets around binomial coefficients there are the commands \( \brace \) and \( \brack \).

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \binom{A}{B} )</td>
<td>( \binom{A}{B} )</td>
</tr>
<tr>
<td>( \dbinom{A}{B} )</td>
<td>( \dbinom{A}{B} )</td>
</tr>
<tr>
<td>( \tbinom{A}{B} )</td>
<td>( \tbinom{A}{B} )</td>
</tr>
<tr>
<td>( \brack{A}{B} )</td>
<td>( \brack{A}{B} )</td>
</tr>
<tr>
<td>( \brace{A}{B} )</td>
<td>( \brace{A}{B} )</td>
</tr>
</tbody>
</table>

### 3.5. Case Differentiations

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \cases ) ( A \rightarrow B &gt; 0 ) \quad { A \quad B &gt; 0 )</td>
<td>( \cases ) ( A \rightarrow B &gt; 0 ) \quad { A \quad B &gt; 0 )</td>
</tr>
<tr>
<td>( \cases ) ( Ctrl+Return ) \quad { A \quad \text{for } x &gt; 0 ) \quad { B \quad \text{for } x = 0 )</td>
<td>( \cases ) ( Ctrl+Return ) \quad { A \quad \text{for } x &gt; 0 ) \quad { B \quad \text{for } x = 0 )</td>
</tr>
</tbody>
</table>
After inserting \texttt{cases} or the usage of the math toolbar button \textbullet, you can create new lines with the shortcut Ctrl+Return or the table toolbar button \textbullet. The command \texttt{cases} is also available via the menu Insert $\rightarrow$ Math $\rightarrow$ Cases-Environment.

### 3.6. Negations

By inserting of \texttt{not} every character can be displayed canceled. The characters are quasi accentuated by a slash.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\not=</td>
<td>$\neq$</td>
</tr>
<tr>
<td>\not \le</td>
<td>$\not\le$</td>
</tr>
<tr>
<td>\not \parallel</td>
<td>$\not\parallel$</td>
</tr>
</tbody>
</table>

The last example shows, that not all negations look good. Therefore there are for some negations special commands (see sec. 13.1 and sec. 14).

### 3.7. Placeholders

When displaying e.g. isotopes\textsuperscript{6} the following problem occurs:

Indices created with sub- and superscripts: $^{19}_9\text{F}$

\texttt{correct indices:} $^{19}_9\text{F}$

The shorter index is by default placed below or above the first character of the longer index. To avoid this there is the command \texttt{\phantom} or the math toolbar button \textbullet that creates one or more phantom characters. When inserting \texttt{\phantom} a small blue box appears that is superposed with two red arrows. The arrows indicate that the complete width and height of the box content will be created as placeholder. Phantom characters are accordingly placeholders with the size of the characters.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\prescript{19}_9\text{F}$</td>
<td>$^{19}_9\text{F}$</td>
</tr>
<tr>
<td>$\prescript{235}_9\text{F}$</td>
<td>$^{235}_9\text{F}$</td>
</tr>
<tr>
<td>$\prescript{\Lambda}{\text{M}}{t}$</td>
<td>$\Lambda^t_{\text{M}M_t}$</td>
</tr>
</tbody>
</table>

\textsuperscript{6}Typesetting isotopes and chemical symbols is described in sec. 20.

\textsuperscript{7}can be found in the submenu of the toolbar button \textbullet.

380
Furthermore there are the commands `\vphantom` (toolbar button \[1\]) and `\hphantom` (toolbar button \[2\]). `\hphantom` creates only space for the maximal height of the characters in the box but not for its width. `\vphantom` creates only space for the width of the box content. Therefore the boxes of both commands have only one red arrow.

For example creates `\vphantom a\int` space for the height of the integral sign,\(^8\) because this is the larger character. An example application is in sec. 18.1.4.

### 3.8. Lines

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\overline{A+B}</code></td>
<td>$A+B$</td>
</tr>
<tr>
<td><code>\underline{A+B}</code></td>
<td>$A+B$</td>
</tr>
<tr>
<td><code>\overline{\underline{A+B}}</code></td>
<td>$A+B$</td>
</tr>
</tbody>
</table>

In the last example it doesn’t matter if first `\overline` or `\underline` is inserted.

To double underline e.g. results, one uses `\underline` twice.

It is possible to place up to 6 lines above or below characters.

Custom lines can be created using the command `\rule` which has the following scheme:

`\rule[vertical offset]{length}{thickness}`

The optional vertical offset shifts the line upwards (or downwards, when the value is negative). The units listed in Table 1 can be used for the values. Here are two example lines created with the commands

`\rule[-2ex]{3cm}{2pt}` and `\rule{2cm}{1pt}`:

This is a sentence _____________ with two lines.

`\rule` can also be used for text when it is inserted in \TeX-mode.

### 3.9. Ellipses

There are different types of ellipses available.\(^9\) For listings dots at the baseline are used (\`\ldots\`), while for operations dots are needed that are on the same height as the operators (\`\cdots\`). When using the command `\dots`, \LaTeX decides on the basis of the next character what type is used.

---

\(^8\)The command `\int` creates an integral sign, see sec. 10.1.

\(^9\)In the math toolbar in the submenu of the button \[\cdots\]
<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A_1, \ldots, A_n)</td>
<td>(A_1, \ldots, A_n)</td>
</tr>
<tr>
<td>(A_1 + \cdots + A_n)</td>
<td>(A_1 + \cdots + A_n)</td>
</tr>
<tr>
<td>(A_1, \ldots, A_n)</td>
<td>(A_1, \ldots, A_n)</td>
</tr>
<tr>
<td>(A_1 + \cdots + A_n)</td>
<td>(A_1 + \cdots + A_n)</td>
</tr>
</tbody>
</table>

\(\vdots\)
\(\ddots\)

3×3 matrix with the different dots
\[
\begin{pmatrix}
A_{11} & \cdots & A_{1m} \\
\vdots & \ddots & \vdots \\
A_{n1} & \cdots & A_{nm}
\end{pmatrix}
\]

The ellipses available in menu \textbf{Insert} > \textbf{Special Character} are \texttt{\ldots}.

Specially for matrices there are ellipses that span over several columns. They are created with the command \texttt{\hdotsfor}, that has the following scheme:

\texttt{\hdotsfor[distance]{number of columns}}

The number of columns specifies how many columns should be spanned. Distance is a factor for the distance between the dots.

In the following matrix the command \texttt{\hdotsfor[2]{4}} was inserted in the first box of the second line, to get an ellipsis with a dot distance twice as long as with the command \texttt{\dots}:

\[
\begin{pmatrix}
A & B & C & D \\
q & w & e & r
\end{pmatrix}
\]

Note that the matrix fields that should be spanned must be empty, otherwise you get \textsc{LATEX}-errors.

Furthermore you can fill with the command \texttt{\dotfill} the rest of a line with dots. The effect of these commands is the same like with \texttt{\hfill}, see sec.8.2.

For example the command \texttt{A\dotfill B} produces

\[
A................................................................................B
\]

Analog to \texttt{\dotfill} there is for a line the command \texttt{\hrulefill}:

\[
A\hrulefillB
\]

To use the commands for text, they have to be inserted in \textsc{TEX}-mode.

4. Matrices

Matrices can be inserted via the math toolbar button \textbf{Matematik} or the menu \textbf{Insert} > \textbf{Math} > \textbf{Matrix}. You will be asked for the number of matrix columns and rows, and the
alignment. The vertical alignment is hereby only of importance for matrices in inline formulas:

\[
\begin{array}{cccc}
A & D & G & J \\
B & E & H & K \\
C & F & I & L
\end{array}
\]

The first matrix is top aligned, the second middle aligned, and the third bottom aligned.

The horizontal alignment specifies how the column entries should be aligned. It is set by entering a letter for every column. \(l\) denotes left aligned, \(c\) centered, and \(r\) right aligned. To create for example a \(4 \times 4\) matrix where the first column is left aligned, the second and third are centered, and the last one is right aligned, one enters for the horizontal alignment \(lccr\). Normally are in a matrix all columns centered, therefore the default is for every column is a \(c\).

Horizontal alignment:

\[
\begin{array}{cccc}
10000 & D & G & 10000 \\
B & 10000 & H & 10000 \\
C & F & 10000 & C & F & 10000 \\
\end{array}
\]

To add or delete rows and columns subsequently, the math toolbar buttons, etc., or the menu Edit > Rows & Columns can be used. New rows can also be created with \(\text{Ctrl}+\text{Return}\).

Parentheses around a matrix can can either be created with the commands \(\left\) and \(\right\) (shortcut Alt+M Parenthesis), see sec.5.1.2, or by using the following commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
</table>
| \(\text{bmatrix}\_2\times2\text{matrix}\) | \[
\begin{bmatrix}
0 & -i \\
i & 0
\end{bmatrix}
\] | \(\text{vmatrix}\_2\times2\text{matrix}\) | \[
\begin{vmatrix}
0 & -i \\
i & 0
\end{vmatrix}
\] |
| \(\text{Bmatrix}\_2\times2\text{matrix}\) | \[
\begin{Bmatrix}
0 & -i \\
i & 0
\end{Bmatrix}
\] | \(\text{Vmatrix}\_2\times2\text{matrix}\) | \[
\begin{Vmatrix}
0 & -i \\
i & 0
\end{Vmatrix}
\] |
| \(\text{pmatrix}\_2\times2\text{matrix}\) | \[
\begin{pmatrix}
0 & -i \\
i & 0
\end{pmatrix}
\] | \(\text{matrix}\_2\times2\text{matrix}\) | \[
\begin{pmatrix}
0 & -i \\
i & 0
\end{pmatrix}
\] |

When e.g. \(\text{vmatrix}\) is inserted, a blue box appears between two vertical lines where the matrix is inserted.

As all multiline formulas are matrices, the length \(\text{arraycolsep}\) that is described in sec.18.1.2 can also be used to change the column separation of matrices.
To change the row separation, the command \textbackslash arraystretch is used. It is used as follows:

\renewcommand{\arraystretch}{stretch factor}

The command \textbackslash renewcommand assigns the stretch factor to the predefined command \textbackslash arraystretch. To double e.g. the row separation, use the factor 2. This is then used for all following matrices. To go back to the original separation, assign the factor 1 to \textbackslash arraystretch.

To set matrices into a text line, the command \textbackslash smallmatrix is used. When it is inserted a blue box with two dashed lines appears. In this box the matrix is inserted. This is a matrix $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ in a text line.

5. Brackets and Delimiters

5.1. Vertical Brackets and Delimiters

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>(</td>
<td>)</td>
<td>)</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
<td>}</td>
<td>}</td>
</tr>
<tr>
<td>\langle</td>
<td>(</td>
<td>\rangle</td>
<td>)</td>
</tr>
<tr>
<td>\lceil</td>
<td>(</td>
<td>\rceil</td>
<td>)</td>
</tr>
<tr>
<td>\lfloor</td>
<td>(</td>
<td>\rfloor</td>
<td>)</td>
</tr>
<tr>
<td>/</td>
<td>/</td>
<td>|</td>
<td>|</td>
</tr>
</tbody>
</table>

Note: In \textbackslash TeX-mode the command \textbackslash textbackslash must be used for the backslash, because the command \textbackslash produces there a line break.

For all characters listed above the size can be adjusted with the commands described in the following two subsections. When using these commands, the characters < and > can directly be used instead of the commands \textbackslash langle and \textbackslash rangle.

5.1.1. Manual Bracket Size

The bracket size can be specified manually by the \textbackslash LaTeX-commands \textbackslash big, \textbackslash Big, \textbackslash bigg, and \textbackslash Bigg. \textbackslash big denotes the smallest and \textbackslash Bigg the largest bracket size.

These commands are used to emphasize levels of brackets:

all brackets in the same size: $((A + B)(A - B))^C$

this looks better: $\left( (A + B)(A - B) \right)^C$
For the second formula the command $\Big((A+B)(A-B)\Big)^C$ has been used.

Here is an overview about all bracket sizes:

$$\Bigg(\exp\bigg<\Big\{\ln(3x)\big)^2\sin(x)\Big\}^A\bigg>\Bigg)^{0,5}$$

Besides the $\big$-commands there is the variant $\bigm$ that adds a bit more space between the bracket and its content, and the variant $\bigl$-$\bigr$, that don’t add additional space. The $l$ at the end of the command $\bigl$ is for a left bracket; for a right bracket this will be replaced by an $r$. A left or right bracket can each be an opening or closing bracket.

In the following table is a comparison of the variants:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Bigm\bigm(\ln(3x)\bigm)^2\Bigm)$</td>
<td>$\left(\ln(3x)\right)^2$</td>
</tr>
<tr>
<td>$\Big(\big(\ln(3x)\big)^2\Big)$</td>
<td>$\left(\ln(3x)\right)^2$</td>
</tr>
<tr>
<td>$\Bigl\bigl(\ln(3x)\bigr)^2\Bigr)$</td>
<td>$\left(\ln(3x)\right)^2$</td>
</tr>
<tr>
<td>$\big\ln(3x)\bigr(\ln(3x)\big)$</td>
<td>$\ln(3x)\ln(3x)$</td>
</tr>
</tbody>
</table>

5.1.2. Automatic Bracket Size

Brackets with variable size can be inserted with the commands $\left$ and $\right$ or via the math toolbar button . Directly behind $\left$ and $\right$ the wanted bracket must be inserted. The bracket size will then automatically be calculated for the output.

normal bracket: The command $\ln(\frac{A}{C})$ creates

$$\ln\left(\frac{A}{C}\right)$$

multiline bracket: The command $\ln\left(\frac{A}{C}\right)$ creates

$$\ln\left(\frac{A}{C}\right)$$
Instead of \texttt{\left} and \texttt{\right} the shortcut Alt+M \texttt{Bracket} can be used. This has the advantage that you can see in LyX immediately the real bracket size and that the matching right bracket will be created too. The command for the last example would then be: \texttt{\ln Alt+M \left(\frac{A}{C}\right)}

To omit a left or right bracket, a dot is inserted for the omitted bracket. For example the command \texttt{\left.\frac{A}{B}\right\}} creates:

\[
\frac{A}{B}
\]

The commands \texttt{\left} and \texttt{\right} will be converted by LyX to brackets in the right size when the document is reloaded and an omitted bracket will appear as dashed line.

Because all popular \LaTeX-Distributions use e\TeX, an extension to \LaTeX, the command \texttt{\middle} is additionally available for all brackets and limits. With this command the height of the following character is adapted to the one of the surrounding brackets, what is e.g. needed for physical vectors:

\[
\langle \phi \mid J = \frac{3}{2}, M_J \rangle
\]

For physical vectors there is a special \LaTeX-package, described in sec.23.3.

### 5.2. Horizontal Brackets

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\overbrace{A+B}_{\hat{3}}</td>
<td>\overbrace{A+B}_{\hat{3}}</td>
</tr>
<tr>
<td>\underbrace{A+B}_{\hat{5}}</td>
<td>\underbrace{A+B}_{\hat{5}}</td>
</tr>
<tr>
<td>\overbrace{\underbrace{A+B}<em>{\hat{7}}}</em>{\hat{C}}</td>
<td>\overbrace{\underbrace{A+B}<em>{\hat{7}}}</em>{\hat{C}}</td>
</tr>
</tbody>
</table>

In the last example it doesn’t matter if \texttt{\overbrace} or \texttt{\underbrace} is inserted at first.

When brackets are needed that overlap each other, multiline formulas, as described in sec.18, must be used:

\[
A = \underbrace{g_{ggg} + b_{bqq} + d_{ddd}}_r + \underbrace{s}_{s}
\]
In the first row the formula is inserted together with the first brace. It is hereby important that the space command\[10\] \: is inserted before the first \(d\), because the brace that ends behind the \(q\) prevents that the following “+” is surrounded by space.\[11\] In the second row the second brace is inserted. As it should begin before the \(b\), first the command \(\text{\texttt{\textbackslash hphantom{gggg+\:}}}\) is inserted.\[12\] This space is needed because the “+” is also surrounded by space in the formula. The brace is placed under the command \(\text{\texttt{\textbackslash hphantom{bbqq+dddd}}}\).

It gets more complicated when brackets overlap each other, like in the following example:

\[
A = \frac{gggg + bbqq + dddd}{r}
\]

The first formula row is the same as the second row of the previous example, with the difference that the brace is above. The second row contains the formula together with the second brace. To avoid that there is space between the upper brace in the first row and the formula, the row spacing need to be reduced. This is not easily possible due to a bug in LyX\[13\]. As solution for the problem, the global formula row separation \(\text{\texttt{jot}}\) must be changed to -6 pt before the formula with the command \(\text{\texttt{\setlength{\jot}{-6pt}}}\) in \(\text{\texttt{\LaTeX}}\)-mode. \(\text{\texttt{jot}}\) is set back after the formula to the standard value of 3 pt using the same command. More about the row separation in formulas is explained in sec.18.1.1.

6. Arrows

Arrows can be inserted via the math toolbar button \(\leftarrow\) or the commands listed in the following subsections.

---

\[10\]Space commands are explained in sec.8.1.

\[11\]because a bracket is not handled as character, see sec.10.3

\[12\]more about \(\text{\texttt{\textbackslash hphantom}}}\) see sec.3.7

\[13\]LyX-bug #1505
### 6.1. Horizontal Arrows

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\gets</td>
<td>←</td>
</tr>
<tr>
<td>\Leftarrow</td>
<td>⇐</td>
</tr>
<tr>
<td>\longleftarrow</td>
<td>←−</td>
</tr>
<tr>
<td>\Longleftarrow</td>
<td>⇐</td>
</tr>
<tr>
<td>\leftharpoonup</td>
<td>⇐</td>
</tr>
<tr>
<td>\leftharpoondown</td>
<td>⇐</td>
</tr>
<tr>
<td>\hookleftarrow</td>
<td>⇐</td>
</tr>
<tr>
<td>\to</td>
<td>→</td>
</tr>
<tr>
<td>\Rightarrow</td>
<td>⇒</td>
</tr>
<tr>
<td>\longrightarrow</td>
<td>−→</td>
</tr>
<tr>
<td>\Longrightarrow</td>
<td>⇔</td>
</tr>
<tr>
<td>\leftharpoonup</td>
<td>⇐</td>
</tr>
<tr>
<td>\leftharpoondown</td>
<td>⇐</td>
</tr>
<tr>
<td>\hookrightarrow</td>
<td>⇐</td>
</tr>
</tbody>
</table>

Arrows used as accent like e.g. vector arrows are listed in sec. 7.

Furthermore there are the labeled arrows \( \xleftarrow{} \) and \( \xrightarrow{} \). When inserting one of these commands in a formula, an arrow with two blue boxes appear where the label can be inserted. The length of the arrow adapts to the label width.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F(a) \xleftarrow{x=a} x&gt;0 \rightarrow F(x) )</td>
<td>( F(a) \xrightarrow{x=a} x&gt;0 \rightarrow F(x) )</td>
</tr>
</tbody>
</table>

### 6.2. Vertical and diagonal Arrows

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\uparrow</td>
<td>↑</td>
</tr>
<tr>
<td>\Uparrow</td>
<td>↑</td>
</tr>
<tr>
<td>\updownarrow</td>
<td>↓</td>
</tr>
<tr>
<td>\Updownarrow</td>
<td>↓</td>
</tr>
<tr>
<td>\Downarrow</td>
<td>↓</td>
</tr>
</tbody>
</table>
| \narrow       | /
| \nearrow      | /
| \searrow      | /
| \swarrow      | /
| \nwarrow      | /

Vertical arrows can be used also as delimiter together with the commands described in sec. 5.1.1 and sec. 5.1.2.
7. Accents

Accents can be inserted via the math toolbar button \(\mathord{\text{\textasciicircum}}\) or the commands listed in the following subsections.

7.1. Accents for one Character\(^{14}\)

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\dot{A})</td>
<td>Ā</td>
</tr>
<tr>
<td>(\ddot{A})</td>
<td>Ă</td>
</tr>
<tr>
<td>(\dddot{A})</td>
<td>Ā̂</td>
</tr>
<tr>
<td>(\dddot{A})</td>
<td>Ā̂</td>
</tr>
<tr>
<td>(\vec{A})</td>
<td>Ā̭</td>
</tr>
<tr>
<td>(\bar{A})</td>
<td>Ā̧</td>
</tr>
<tr>
<td>(\mathring{A})</td>
<td>Ā̆</td>
</tr>
<tr>
<td>(\tilde{A})</td>
<td>Ā̃</td>
</tr>
<tr>
<td>(\hat{A})</td>
<td>Ā̂</td>
</tr>
<tr>
<td>(\check{A})</td>
<td>Ā̊</td>
</tr>
<tr>
<td>(\acute{A})</td>
<td>Ā́</td>
</tr>
<tr>
<td>(\grave{A})</td>
<td>Ā̀</td>
</tr>
<tr>
<td>(\breve{A})</td>
<td>Ā̍</td>
</tr>
</tbody>
</table>

You can directly insert accents like \(\text{é}\) to formulas. \texttt{LyX} will transform them to the corresponding accent command. For umlauts it is better to insert a quotation mark before the vowel. These two characters are then treated by \texttt{LaTeX} as one character when the formula part with the umlaut is marked as German. In contrary to \(\ddot{A}\), with this method “real” umlauts are created as demonstrated in the following example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>“i&quot;</td>
<td>ī</td>
</tr>
<tr>
<td>(\ddot{.i})</td>
<td>ī̂</td>
</tr>
</tbody>
</table>

Another advantage to \(\ddot{A}\) is that umlauts can directly be converted to mathematical text because the accent commands above are not allowed in mathematical text. To convert an accented character to mathematical text, only the character under the accent may be converted. This applies also for all other conversions, e.g. to italic or bold.

In mathematical text, umlauts and other accented characters can directly be inserted.

7.2. Accents for Operators

With the commands \(\overset\) and \(\underset\) characters can be placed above or below an operator, respectively, to accent it. With the command \(\sideset\) characters can be set before and behind an operator. The command scheme is:

\(^{14}\)accents in text see sec. 16.2
\sideset{character before}{character behind}

\sideset must always be before the operator that should be accented. You can accent with several characters and even with other operators and symbols. To place with \sideset for example only characters behind an operator, write nothing between the first braces but don’t omit the braces.

For example the command \( \sideset{\rightarrow}{\leftarrow} \sum_{k=1}^{n} \) produces:

\[ \sum_{k=1}^{n} \]

The command \( \overset{\maltese}{a} \) produces:

\[ a \]

As seen in the last example, with \( \overset \) and \( \underset \) also symbols and characters can be accented; with \( \sideset \) this is not possible.

### 7.3. Accents for several Characters

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\overleftarrow ( A=B )</td>
<td>( \overleftarrow{A=B} )</td>
<td>\oversetrightarrow ( A=B )</td>
<td>( \oversetrightarrow{A=B} )</td>
</tr>
<tr>
<td>\underleftarrow ( A=B )</td>
<td>( \underleftarrow{A=B} )</td>
<td>\underrightarrow ( A=B )</td>
<td>( \underrightarrow{A=B} )</td>
</tr>
<tr>
<td>\overleftrightharpoon ( A=B )</td>
<td>( \overleftrightharpoon{A=B} )</td>
<td>\widetilde ( A=B )</td>
<td>( \widetilde{A=B} )</td>
</tr>
<tr>
<td>\underleftrightharpoon ( A=B )</td>
<td>( \underleftrightharpoon{A=B} )</td>
<td>\widehat ( A=B )</td>
<td>( \widehat{A=B} )</td>
</tr>
</tbody>
</table>

With these commands as many characters as you like can be accented. But the accents \( \widetilde \) and \( \widehat \) will only be set in the output with a length of three characters, as shown in the following example:

\[ A + \overleftarrow{B} = \overleftarrow{C} - D \]

With the commands \( \overset \) and \( \underset \) described in the previous subsection it is also possible to accent several characters. The command \( \underset{\ldots}{A=B} \) creates:

\[ A = B \]

390
8. Space

8.1. Predefined Space

Sometimes it is necessary to insert horizontal space to a formula. This is done by inserting a protected space (shortcut Ctrl+Space). A “␣” appears and by pressing Space several times one can select one of eight different space sizes. The spaces can also be inserted using the math toolbar button ♦ or special commands. Independent from the inserted command, one can select the size again by pressing Space afterwards.

<table>
<thead>
<tr>
<th>Command</th>
<th>,</th>
<th>:</th>
<th>;</th>
<th>\quad</th>
<th>\qquad</th>
<th>!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Space keystrokes after inserting the protected space</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Result</td>
<td>\ A B</td>
<td>\ A B</td>
<td>\ A B</td>
<td>\ A B</td>
<td>\ A B</td>
<td>\ A B</td>
</tr>
</tbody>
</table>

The last size seem to produce no space. It is displayed red in LyX contrary to the other sizes, because it is a negative space. There are two more negative spaces:

<table>
<thead>
<tr>
<th>Command</th>
<th>\negmedspace</th>
<th>\negthickspace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Space keystrokes after inserting the protected space</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Result</td>
<td>\ A B</td>
<td>\ A B</td>
</tr>
</tbody>
</table>

Negative spaces can lead to characters overlapping each other. Thus they can be used to enforce ligatures, what is e.g. useful for summation operators:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\sum \sum f_{kl}</td>
<td>\sum \sum f_{kl}</td>
</tr>
<tr>
<td>\sum \negmedspace \sum f_{kl}</td>
<td>\sum \negmedspace \sum f_{kl}</td>
</tr>
</tbody>
</table>

Relations like for example equal signs, are always surrounded by space. To suppress this, the equal sign is placed into a \TeX-brace. The following example demonstrates this:

normal equation \quad A = B

equation without space \quad A = B

The command for the last formula is: \( A \{=\to B \)

Spaces are needed for physical units, because the space between the value and the unit is the smallest one and not a normal space. For units in text, the smallest space is inserted via the menu Insert > Formatting > Thin Space (shortcut Ctrl+Shift+Space).

An example to visualize the difference:

24 kW·h space between value and unit

24 kW·h smallest space between value and unit
8.2. Variable Space\(^{15}\)

Space with a defined length can be inserted with the command \texttt{\hspace}. Then a long “␣” appears. The length can be specified by left-clicking on the “␣”. The length may also be negative. To insert so many space that the formula uses all available space, the command \texttt{\hfill} is used.

\begin{center}
\begin{tabular}{|c|c|}
\hline
Command (\texttt{\hspace length}) & Result \\
\hline
\texttt{A=B\hspace␣\rightarrow A\not=C \ (3 \text{ cm})} & \texttt{A = B} \quad A \neq C \\
\texttt{A\hspace␣\rightarrow A\not=A \ (-1 \text{ mm})} & \texttt{A \neq A} \\
\texttt{A=A\hfill␣B=B} & \texttt{A = A} \quad B = B \\
\hline
\end{tabular}
\end{center}

In the last example the available space is given by the longest column entry of the table. In an inline formula the space depends on the length of the line in which \texttt{\hfill} is inserted. Thus, when the line uses the full width, no space will be created. \texttt{\hfill} only has an effect on displayed formulas when the formula style \texttt{Indented} is used. (Formula styles are explained in sec. 17.)

Besides \texttt{\hfill}, there are the commands \texttt{\dotfill} and \texttt{\hrulefill} that fill the space with a pattern, see sec. 3.9 for an example.

For text, variable space can be inserted via the menu Insert $>$ Formatting $>$ Horizontal Space:

This is a line with \hspace{2cm} space.

This is a line with \hspace{max} space.

8.3. Space besides inline Formulas

The space that surrounds inline formulas can be adjusted with the length \texttt{\mathsurround}. The value of a length is set with the command \texttt{\setlength} that has the following scheme:

\texttt{\setlength{\texttt{length name}}{\texttt{value}}} \\

To set \texttt{\mathsurround} to the value 5mm, the command

\texttt{\setlength{\mathsurround}{5mm}}

is inserted in \TeX-mode. 5mm space will now be set around all inline formulas:

This is a line with an inline formula $A = B$ with 5mm surrounding space.

To return to the predefined value, \texttt{\mathsurround} is set to the value 0pt.

\(^{15}\)for vertical space in formulas see sec. 18.1.1
9. Boxes and Frames

Boxes for text are described in chapter Boxes in the Embedded Objects manual.

9.1. Boxes with Frame

It is possible to frame formulas or parts of it with the commands \fbox and \boxed.

When one of these commands is inserted to a formula, a blue box appears within a frame to enter formula parts. For \fbox an additional formula has to be created by Ctrl+M within this box, because the box content will otherwise be treated as mathematical text. When \boxed is used, a new formula is automatically created inside the frame.

The command \fbox is not suitable to frame displayed formulas because the formula will always be set in the size of the text. \boxed is in contrary not suitable to frame inline formulas, because the formula will always be set in the size of a displayed formula.

As extension to \fbox there is the command \framebox where additionally the frame width and the alignment can be specified. \framebox is used in the following scheme:

\framebox[frame width][position]{box content}

The position can either be l or r. l left aligns, r right aligns the formula in the box. When no position is given, the formula will be centered.

Is no width given, also no position can be given. In this case the frame width is adjusted to the box content like for \fbox.

When the command \framebox is inserted, a box appears containing three blue boxes. The first two boxes are surrounded by brackets and denote the two optional arguments. The third box is for formula parts like for \fbox.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\fbox_{Ctrl+M}\int_{A=B}</td>
<td>[ A = B ]</td>
</tr>
<tr>
<td>\boxed_{\int_{A=B}}</td>
<td>[ A = B ]</td>
</tr>
<tr>
<td>A+\fbox_{B}</td>
<td>A + $B$</td>
</tr>
<tr>
<td>\framebox_20mm\rightarrow\int_{A=B}</td>
<td>$\frac{A}{B}$</td>
</tr>
</tbody>
</table>

The frame thickness can also be adjusted. To do this the following commands have to be inserted in \TeX-mode before the formula.
“distance” specifies the minimal distance between the frame and the first character in the box. An example for this is the following framed formula:

\[
\begin{array}{c}
A + B = C
\end{array}
\]

Before this formula the commands
\fboxrule 2mm \fboxsep 3mm
were inserted in \TeX-mode. The given values are used for all following boxes. To return to the standard frame size, the command
\fboxrule 0.4pt \fboxsep 3pt
is inserted in \TeX-mode before the next formula.

### 9.2. Boxes without Frame

For boxes without a frame there are the following box commands: \texttt{\mbox}, \texttt{\makebox}, and \texttt{\raisebox}

With \texttt{\raisebox} a box can be super- or subscripted. But in contrary to normal super- and subscripting, the characters in the box keep their font size. \texttt{\raisebox} is used in the following scheme:

\texttt{\raisebox{height}\{box content\}}

When the box should contain a formula, an extra formula is needed like for \texttt{\fbox}. \textbf{Note:} For \texttt{\raisebox} this extra formula is created by pressing Ctrl+M twice instead of once because \LaTeXX doesn’t yet support \texttt{\raisebox} directly.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>H\raisebox{2mm}{al→lo}</td>
<td>H^{al}_{lo}</td>
</tr>
<tr>
<td>H\raisebox{-2mm}{al→lo}</td>
<td>H^{al}_{lo}</td>
</tr>
<tr>
<td>A=\raisebox{-2mm}{\text{Ctrl+M Ctrl+M }sqrt_{B}}</td>
<td>\text{A = } \sqrt{B}</td>
</tr>
</tbody>
</table>

The command \texttt{\mbox} is equivalent to \texttt{\fbox} and \texttt{\makebox} is equivalent to \texttt{\framebox}, with the difference that there is no frame.
9.3. Colored Boxes

To be able to use all commands explained in this section, the \texttt{\LaTeX}-package \texttt{color}\textsuperscript{16} has to be loaded in the \texttt{\LaTeX}-preamble with the line\textsuperscript{17}

\texttt{\usepackage{color}}

To color boxes, the command \texttt{\colorbox} is used in the following scheme:

\texttt{\colorbox{color}{box content}}

The box content can also be a box and a \texttt{\colorbox} can also be part of another box (see the 2nd and 3rd example). When the box should contain a formula, an extra formula has to be created, the same way as for \texttt{\raisebox}.\textsuperscript{18}

One of the following predefined colors can be chosen: \texttt{black}, \texttt{blue}, \texttt{cyan}, \texttt{green}, \texttt{magenta}, \texttt{red}, \texttt{white}, and \texttt{yellow}

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{\colorbox{yellow}{{A=B}}</td>
</tr>
<tr>
<td>\texttt{\colorbox{green}{{\fbox A=B}}</td>
</tr>
<tr>
<td>\texttt{\fbox\colorbox{green}{{Ctrl+M Ctrl+M \int C=D}}</td>
</tr>
</tbody>
</table>

\texttt{\colorbox} only colors the box but not the characters in the box. To color all characters, the whole formula is highlighted and the wanted color is chosen in the \texttt{Text Style} dialog. The dialog can be called with the toolbar button \texttt{ab} or the menu \texttt{Edit>Text Style>Customized}. The formula number has then the same color as the formula. When the formula number should get another color than the formula characters, the color must be changed within the formula.

An example:

\begin{align*}
\int A &= B \\
\int A &= B
\end{align*}

\textsuperscript{16}The \texttt{\LaTeX}-package \texttt{color} is part of every \texttt{\LaTeX} standard installation.

\textsuperscript{17}When text is colored somewhere in the document with a predefined color, \texttt{\LaTeX} loads the \texttt{\LaTeX}-package \texttt{color} automatically. Thus it is possible that the package is loaded twice, but this doesn’t arise problems.

\textsuperscript{18}This also applies for the command \texttt{\fcolorbox}. 395
Formula (1) is completely colored red.
Formula (2) was first completely colored green to set the color for the formula number. Subsequently the formula characters were colored red.

To color the frame different than the rest of the box, the command \textcolor{color}{box content} is used in the following scheme:
\textcolor{frame color}{color}{box content}
So \textcolor is an extension of the command \colorbox. The frame width is set, like for \framebox, with \fboxrule and \fboxsep. An example:
\textcolor{cyan}{magenta}{A=B}

This formula was created with the command\textcolor{cyan}{magenta}{A=B}.

To use other colors than the predefined ones, they have to be defined first.
One can for example define the color “darkgreen” with the \LaTeX-preamble line:
\definecolor{darkgreen}{cmyk}{0.5, 0, 1, 0.5}
cmyk is the color space that denotes the colors cyan, magenta, yellow, and black. The four comma separated numbers are the portion factor for the corresponding colors of the color space. The factors can be in the range of 0-1. Instead of cmyk also the color space rgb can be used for definitions. rgb denotes red, green, and blue, so that there are in this case three portion factors for the corresponding colors. Furthermore there is the color space gray with one portion factor for the gray value.
As example a framed box with the new defined color darkgreen where the characters have been colored yellow:
\textcolor{darkgreen}{yellow}{\int A \, dx = \frac{\sqrt{B}}{\ln \left( \frac{3}{2} \right)}}
Self-defined colors can also be used for text with the help of the command \textcolor: This sentence is “darkgreen”.
\textcolor is used in the scheme \textcolor{color}{characters to color}.

9.4. Paragraph Boxes
A box that can contain several lines and paragraphs, a so called paragraph box (parbox), can be created with the menu Insert>Box or the toolbar button.
The following example shows a framed parbox in a line:

This is a line

This is a paragraph box.
It is exactly 5 cm long and can also contain formulas:
\[ \int A \, ds = C \]

with a parbox.

Such a box is created by right-clicking on the gray box inset. A dialog pops up showing the box properties. In our case set: Decoration: Recangular box, Inner Box: Parbox, Width: 5 cm, Vertical Box Alignment: Middle

In \LaTeX{} a parbox is created with the command \texttt{\parbox} that has the following scheme:

\texttt{\parbox[position]{width}{box content}}

The positions \texttt{b} and \texttt{t} are possible. \texttt{b} for bottom means that the box is aligned within the surrounding text with its last line. With \texttt{t} for top this is done with the first line. When no position is given, the box will be vertically centered, see section \textit{Boxes} of the \textit{Embedded Objects} manual for examples.

To frame formulas completely, including the formula number, the formula must be set into a parbox. To do this, the command \texttt{\fbox{\parbox{\linewidth-2\fboxsep-2\fboxrule}{box content}}} is inserted in \TeX{}-mode before the formula. \texttt{\linewidth} is hereby the line width set for the document. Because the frame is outside the parbox, 2 times the frame separation and the frame thickness must be subtracted from the line width. As this is not automatically done by LyX due to a bug\textsuperscript{19}, \TeX{}-mode has to be used. To be able to multiply and subtract in arguments, the \LaTeX{}-package \texttt{calc}\textsuperscript{20} must be loaded in the \LaTeX{}-preamble with the line

\texttt{\usepackage{calc}}

Behind the formula both boxes are closed by entering \texttt{}}} in \TeX{}-mode. Here is an example:

\[ \int A \, dx = \frac{\sqrt[5]{B}}{\ln \left( \frac{1}{3} \right)} \] (4)

As a parbox is used as argument of \texttt{\fbox}, there is in this case no difference between \texttt{\fbox} and \texttt{\boxed}.

\textsuperscript{19}LyX-bug \#4483

\textsuperscript{20}calc is part of every \LaTeX{} standard installation.
Paragraph boxes are very useful to comment formulas directly. To do this, \texttt{parbox} is used in combination with the command \texttt{tag}. (more about \texttt{tag} see sec.19.4)

An example of a formula commented with \texttt{parbox}:

\[
5x - 7b = 3b
\]

This is a description. It is distinctly separated from the formula and multiline.

Such a formula must be inserted completely in \TeX-mode because LyX does not yet support the command \texttt{parbox} in formulas. The formula is created with the following command sequence:

The command \texttt{[5x-7b=3b\tag*\{parbox{5cm}\}} is inserted in \TeX-mode. Then the description follows as normal text, and finally} \texttt{]} is inserted in \TeX-mode. The commands \texttt{[} and \texttt{]} hereby create a displayed formula.

The advantages of \texttt{parbox} can be seen in this example that was “commented” using the mathematical textmode:

\[
5x - 7b = 3b\text{This is a description. It is not separated from the formula ...}
\]

10. Operators

10.1. Big Operators

To be able to use all integral operators listed here, the option \texttt{Use esint package automatically} must be set in the document settings under Math Options.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\int</td>
<td>$\int$</td>
</tr>
<tr>
<td>\oint</td>
<td>$\oint$</td>
</tr>
<tr>
<td>\ointclockwise</td>
<td>$\oint$</td>
</tr>
<tr>
<td>\sqint</td>
<td>$\sqint$</td>
</tr>
<tr>
<td>\fint</td>
<td>$\fint$</td>
</tr>
<tr>
<td>\landupint</td>
<td>$\landupint$</td>
</tr>
<tr>
<td>\landdownint</td>
<td>$\landdownint$</td>
</tr>
<tr>
<td>\bigcap</td>
<td>$\bigcap$</td>
</tr>
<tr>
<td>\bigcup</td>
<td>$\bigcup$</td>
</tr>
</tbody>
</table>

All big operators can also be inserted via the math toolbar button $\int$.\textsuperscript{21}

\textsuperscript{21}When the formula style \texttt{Indented} is used, \texttt{\tag*\{} can also be replaced by \texttt{hfill}. (formula styles see sec.17)
The operators are called big because they are bigger than the sometimes equal looking binary operators. All big operators can have limits as described in the next subsection.

For all integral operators there is a second version available, ending on \texttt{op:} \verb|\intop|, \verb|\ointop| etc. These operators are different from \verb|\int| etc. in the style the operator limits are displayed, see sec. 10.2.

\textbf{Advice for Integrals}

The letter \texttt{d} in an integral is an operator, that therefore has to be set upright. This is done by highlighting the \texttt{d} and using the keyboard shortcut Alt+C R. Finally the smallest space is inserted before the \texttt{d}, as this is usual for operators. An example:

incorrect: \( \int A(x)dx \)

correct: \( \int A(x) \, dx \)

For multiple integrals there are the following commands:

\begin{center}
\begin{tabular}{|c|c|}
\hline
\texttt{Command} & \texttt{Result} \\
\hline
\texttt{\intint} & \intint \\
\texttt{\ointoint} & \ointoint \\
\texttt{\sqintsqint} & \sqintsqint \\
\hline
\end{tabular}
\end{center}

\begin{center}
\begin{tabular}{|c|c|}
\hline
\texttt{Command} & \texttt{Result} \\
\hline
\texttt{\iiintiint} & \iiintiint \\
\texttt{\iiintiiint} & \iiintiiint \\
\texttt{\dotsintdotsint} & \dotsintdotsint \\
\hline
\end{tabular}
\end{center}

\textbf{10.2. Operator Limits}

Limits are created by super- and subscripts:

\begin{center}
\begin{tabular}{|c|c|}
\hline
\texttt{Command} & \texttt{Result} \\
\hline
\texttt{\prod\prod\prod} & \prod_{0}^{\infty} A(x) \\
\hline
\end{tabular}
\end{center}

Limits of inline formulas are set right beside the operator. Limits in displayed formulas are set above or below the operator, except for integral limits.

To force that the limits are set beside the operator, the cursor is set directly behind the operator and the limits type is changed with the menu \texttt{Edit} $\Rightarrow$ \texttt{Math} $\Rightarrow$ \texttt{Change Limits Type} to \texttt{Inline} (shortcut Alt+M L). An example:

The default limits type is this:

\[
\sum_{x=0}^{\infty} \frac{1}{x^2}
\]

\textsuperscript{22}Font styles see sec. 11.1
This is how it looks when the limits type was changed to **Inline**:

$$\sum_{x=0}^{\infty} \frac{1}{x^2}$$

For integrals, except those ending with \texttt{\textbackslash intop}, \texttt{\textbackslash ointop} etc., the limits are by default set beside the operator. But for multiple integrals the limits are often set below the operator. In the following example the limits type was therefore set to **Display** and so set below the integrals:

$$\iiint_V X \, dV = U \quad (5)$$

To specify conditions for limits, the commands \texttt{\textbackslash subarray} and \texttt{\textbackslash substack} are used. To create for example this expression

$$\sum_{\substack{0<k<1000 \\ k \in \mathbb{N}}} n \kern-0.5pt k^{-2} \quad (6)$$

the following has to be done:

First the command \texttt{\textbackslash sum} is typed in. One is now in a blue box under the summation operator and insert there the command \texttt{\textbackslash subarray}. The blue box is now within a purple box and now several lines can be written among each other. A new line is created by inserting a line break (Ctrl+Return). When now \texttt{0<k<1000 Ctrl+Return} is typed in, a new box appears below for the new line.

The alignment of the lines can be changed to left aligned with the **Table toolbar** or the menu **Edit \rightarrow Rows & Columns**. To get right alignment, \texttt{\textbackslash hfill} is inserted at the beginning of the line.

The command \texttt{\textbackslash substack} is equivalent to \texttt{\textbackslash subarray} with the difference that the lines are always centered.

Like in formula (6) there can be too much space beside an operator, because the characters following the operator are set beside the limits.

To avoid this, the following macro can be used in the \texttt{\textbackslash \LaTeX}-preamble:

\begin{verbatim}
\def\clap#1{\hbox to 0pt{\hss #1\hss}}
\def\mathclap #1#2{\clap{$\mathsurround =0pt #1{#2}$}}
\end{verbatim}

This defines the command \texttt{\textbackslash mathclap} that sets the width of the limit to 0 pt. The command scheme is

\texttt{\mathclap{limit}}
where the limit can consist of several conditions.

Applied on formula (6), one uses the command

\[
\sum_{\substack{0<k<1000} \text{Ctrl+Return}}
\]

to create the lower limit. The summand is now directly behind the summation operator:

\[
\sum_{0<k<1000 \atop k \in \mathbb{N}} k^{-2}
\]

How to use one limit for several operators is described in sec. 10.4.

10.3. Binary Operators

Binary operators are surrounded by space when there is a character before and behind them.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>\pm</td>
<td>±</td>
</tr>
<tr>
<td>\mp</td>
<td>±</td>
</tr>
<tr>
<td>\cdot</td>
<td>·</td>
</tr>
<tr>
<td>\times</td>
<td>×</td>
</tr>
<tr>
<td>\div</td>
<td>÷</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>\star</td>
<td>•</td>
</tr>
<tr>
<td>\circ</td>
<td>⋄</td>
</tr>
<tr>
<td>\diamond</td>
<td>⋆</td>
</tr>
<tr>
<td>\bullet</td>
<td>•</td>
</tr>
<tr>
<td>\nabla</td>
<td>∇</td>
</tr>
<tr>
<td>\bigtriangledown</td>
<td>▽</td>
</tr>
<tr>
<td>\bigtriangleup</td>
<td>Δ</td>
</tr>
<tr>
<td>\Box</td>
<td>□</td>
</tr>
<tr>
<td>\cap</td>
<td>∩</td>
</tr>
<tr>
<td>\cup</td>
<td>∪</td>
</tr>
<tr>
<td>\dag</td>
<td>†</td>
</tr>
<tr>
<td>\ddag</td>
<td>‡</td>
</tr>
<tr>
<td>\wr</td>
<td>⨿</td>
</tr>
<tr>
<td>\bigcirc</td>
<td>⃝</td>
</tr>
<tr>
<td>\wedge</td>
<td>∧</td>
</tr>
<tr>
<td>\vee</td>
<td>∨</td>
</tr>
</tbody>
</table>

All binary operators can also be inserted via the math toolbar button ±.

To typeset the Laplace operator also \(\Delta\) or \(\nabla^2\) can be used instead of \(\bigtriangleup\).

The character Menu Separator from the menu Insert ▶ Special Character is the operator \(\triangleright\).

10.4. Self-defined Operators

With the help of the command \(\DeclareMathOperator\) custom operators can be defined in the \LaTeX\-preamble. Its command scheme is:
\DeclareMathOperator{new command}{display}

Display can be characters or symbols that define how the operator looks in the output. To define a big operator a * is set behind the command. All self-defined big operators can have limits as described in sec. 10.2.

For example the \LaTeX-preamble line
\[\DeclareMathOperator*{\Lozenge}{\blacklozenge}\]
defines the command \Lozenge, that inserts a big operator consisting of the lozenge symbol from sec. 13.2:
\[\lozenge\]

The command for this formula is: \Lozenge^{\text{\infty}}_{n=1}

When self-defined operators are not used several times in the document, they can also be defined with the commands \texttt{\mathop} and \texttt{\mathbin}, which have the following scheme:

\[\mathop{display}\text{ and }\mathbin{display}\]

\texttt{\mathop} defines big operators, \texttt{\mathbin} binary operators.
\texttt{\mathop} can e.g. be used to use one limit for several operators:
\[\sum\sum_{i,j=1}^{N}\]

The command for the formula above is:
\[\text{\mathop}\sum\negmedspace\sum_{i,j=1}^{N}\]

11. Fonts

11.1. Font Styles

Latin letters in formulas can be set in one of the following font styles:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
<th>shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>\mathbb_ABC</td>
<td>\textit{ABC}</td>
<td>Alt+C C</td>
</tr>
<tr>
<td>\mathbf AbC</td>
<td>\textbf{AbC}</td>
<td>Ctrl+B</td>
</tr>
<tr>
<td>\textbf{\textit{AbC}}</td>
<td>Ctrl+Alt+B, Alt+C B</td>
<td></td>
</tr>
<tr>
<td>\textsc{ABC}</td>
<td>\textsc{ABC}</td>
<td>Ctrl+E</td>
</tr>
<tr>
<td>\mathfrak ABC</td>
<td>\textfrak{ABC}</td>
<td>-</td>
</tr>
<tr>
<td>\textfrak{AbC}</td>
<td>\textfrak{AbC}</td>
<td>-</td>
</tr>
</tbody>
</table>

402
<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
<th>shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{AbC}</td>
<td>AbC</td>
<td>-</td>
</tr>
<tr>
<td>\textbf{AbC}</td>
<td>AbC</td>
<td>Alt+C R</td>
</tr>
<tr>
<td>\textsl{AbC}</td>
<td>AbC</td>
<td>Alt+C S</td>
</tr>
<tr>
<td>\texttt{AbC}</td>
<td>AbC</td>
<td>Ctrl+Shift+P</td>
</tr>
</tbody>
</table>

**Note:** The styles \textbb and \textcal can only be used for big letters.

Predefined is the style \textnormal.

The style commands work also for letters in mathematical constructs:

\[ \mathfrak{A} = \frac{b}{c} \]

Characters in mathematical text don’t appear in a math font style but in the text font style \texttt{\textnormal}. That their style can’t be set correctly via the text style dialog is a bug in LyX.\(^\text{23}\)

Instead of the style commands the dialog Edit -> Math -> Text Style or the toolbar button can be used.

### 11.2. Bold Formulas

To make a complete formula bold, the command \textbf from the previous subsection cannot be used, because it doesn’t work for small Greek letters. Furthermore it prints Latin letters always upright, like in the following equation:

\[ \int_n^2 f(\theta) = \Gamma \quad \text{equation with \textbf} \]

To display the formula correctly, the command \textbf{\textbf} is used:

\[ \int_n^2 f(\theta) = \Gamma \quad \text{equation with \textbf{\textbf}} \]

It is also possible to set the formula in a boldmath environment. This environment is created by inserting the command \textbf{\textbf} in \texttt{\texttt} mode. To end the environment, the command \textbf{\textbf} is inserted in \texttt{\texttt} mode.

\[ \int_n^2 f(\theta) = \Gamma \quad \text{equation in a boldmath environment} \]

\(^\text{23}\)LyX-bug \#4629
11.3. Colored Formulas

Formulas can be colored like normal text: Highlight a formula or a formula part and use the Text Style dialog. Here is a formula in magenta:

\[ \int A \, dx = \frac{\sqrt{B}}{\ln \left( \frac{1}{3} \right)} \]

You can also define your own colors as described in sec. 9.3. They can be used with the \textcolor{color}{characters or formula} command in the scheme.

\textcolor{color}{characters or formula}

The following example was colored completely dark green and partly red:

\[ \int A \, dx = \frac{\sqrt{B}}{\ln \left( \frac{1}{3} \right)} \]

Due to a bug in LyX only complete formulas can be colored with self-defined colors.\textsuperscript{24}

11.4. Font Sizes

For characters in formulas there are, analog to characters in text, the following size commands:\n
\texttt{\Huge, \huge, \LARGE, \Large, \large, \normalsize, \small,}\n\texttt{\footnotesize, \scriptsize, and \tiny}\n
The size produced by the commands depends on the document font size, that corresponds with the command \texttt{\normalsize}. The other commands produce smaller or larger sizes than \texttt{\normalsize}. The font size can however not exceed a certain value. Is for example the document font size 12pt, the command \texttt{\Huge} switches to the same size as \texttt{\huge}.

A size command is inserted in \TeX-mode before the formula and sets the size for all following formula and text characters. To switch back to the initial size, the command \texttt{\normalsize} is inserted behind the formula in \TeX-mode.

Within a formula the size can only be changed for symbols or letters in mathematical text. To do this, the size command is inserted in mathematical text. All following characters until the end of the mathematical text or until another size command will have the selected size. Two examples:

\textsuperscript{24}LyX-bug #5269
Before both formulas the command \texttt{\huge} was inserted. The command for the second formula is:
```
\texttt{\maltese A} \texttt{Alt+M M \Large \texttt{\maltese \textit A}}
```

If a symbol cannot be displayed in different sizes, it will always be displayed in the default size.

## 12. Greek Letters

All Greek letters can also be inserted via the toolbar button χ.

### 12.1. Small Letters

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\alpha</td>
<td>α</td>
</tr>
<tr>
<td>\beta</td>
<td>β</td>
</tr>
<tr>
<td>\gamma</td>
<td>γ</td>
</tr>
<tr>
<td>\delta</td>
<td>δ</td>
</tr>
<tr>
<td>\epsilon</td>
<td>ε</td>
</tr>
<tr>
<td>\varepsilon</td>
<td>ε</td>
</tr>
<tr>
<td>\zeta</td>
<td>ζ</td>
</tr>
<tr>
<td>\eta</td>
<td>η</td>
</tr>
<tr>
<td>\theta</td>
<td>θ</td>
</tr>
<tr>
<td>\vartheta</td>
<td>θ</td>
</tr>
</tbody>
</table>

How to create upright Greek letters is explained in sec. 23.9.
12.2. Big Letters

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\Gamma</td>
<td>Γ</td>
</tr>
<tr>
<td>\Delta</td>
<td>∆</td>
</tr>
<tr>
<td>\Theta</td>
<td>Θ</td>
</tr>
<tr>
<td>\Lambda</td>
<td>Λ</td>
</tr>
<tr>
<td>\Xi</td>
<td>Ξ</td>
</tr>
<tr>
<td>\Pi</td>
<td>Π</td>
</tr>
<tr>
<td>\Sigma</td>
<td>Σ</td>
</tr>
<tr>
<td>\Upsilon</td>
<td>Υ</td>
</tr>
<tr>
<td>\Phi</td>
<td>Φ</td>
</tr>
<tr>
<td>\Psi</td>
<td>Ψ</td>
</tr>
<tr>
<td>\Omega</td>
<td>Ω</td>
</tr>
</tbody>
</table>

That the big Greek letters appear upright is caused by a design bug when TeX was developed. To get correct italic big letters, begin every command with var. For example the command \texttt{varGamma} produces: Γ

12.3. Bold Letters

Greek letters cannot be set with different font styles like Latin letters. They can only be made bold with the command \texttt{boldsymbol}.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\Upsilon\texttt{boldsymbol}\Upsilon</td>
<td>Υ</td>
</tr>
<tr>
<td>\theta\texttt{boldsymbol}\theta</td>
<td>θ</td>
</tr>
</tbody>
</table>

13. Symbols\textsuperscript{25}

Many of the symbols listed in this section can also be inserted via the toolbar buttons $\nabla$ and $\vec{F}$.

13.1. Mathematical Symbols

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\neg</td>
<td>¬</td>
</tr>
<tr>
<td>\Im</td>
<td>ℑ</td>
</tr>
<tr>
<td>\Re</td>
<td>ℜ</td>
</tr>
<tr>
<td>\aleph</td>
<td>ℵ</td>
</tr>
<tr>
<td>\partial</td>
<td>∂</td>
</tr>
<tr>
<td>\infty</td>
<td>∞</td>
</tr>
<tr>
<td>\wp</td>
<td>ϕ</td>
</tr>
<tr>
<td>\imath</td>
<td>𝕟</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\forall</td>
<td>∀</td>
</tr>
<tr>
<td>\exists</td>
<td>∃</td>
</tr>
<tr>
<td>\nexists</td>
<td>∄</td>
</tr>
<tr>
<td>\emptyset</td>
<td>∅</td>
</tr>
<tr>
<td>\varnothing</td>
<td>∅</td>
</tr>
<tr>
<td>\dag</td>
<td>†</td>
</tr>
<tr>
<td>\ddag</td>
<td>‡</td>
</tr>
<tr>
<td>\complement</td>
<td>ℭ</td>
</tr>
<tr>
<td>\Bbbk</td>
<td>ℌ</td>
</tr>
</tbody>
</table>

\textsuperscript{25}A list with all symbols of most of the L\TeX-packages can be found in [5].
13.2. Miscellaneous Symbols

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
<th>Command</th>
<th>Result</th>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\flat</td>
<td>♭</td>
<td>\hbar</td>
<td>h</td>
<td>\diamondsuit</td>
<td>♦</td>
</tr>
<tr>
<td>\natural</td>
<td>♪</td>
<td>\hslash</td>
<td>ℏ</td>
<td>\Diamond</td>
<td>♦</td>
</tr>
<tr>
<td>\sharp</td>
<td>♯</td>
<td>\clubsuit</td>
<td>♠</td>
<td>\heartsuit</td>
<td>♦</td>
</tr>
<tr>
<td>\surd</td>
<td>√</td>
<td>\spadesuit</td>
<td>♦</td>
<td>\P</td>
<td>¶</td>
</tr>
<tr>
<td>\checkmark</td>
<td>✓</td>
<td>\bigstar</td>
<td>★</td>
<td>\copyright</td>
<td>©</td>
</tr>
<tr>
<td>\yen</td>
<td>¥</td>
<td>\blacklozenge</td>
<td>♦</td>
<td>\circledR</td>
<td>®</td>
</tr>
<tr>
<td>\pounds</td>
<td>$</td>
<td>\blacktriangle</td>
<td>▲</td>
<td>\maltese</td>
<td>✠</td>
</tr>
<tr>
<td>$</td>
<td>$</td>
<td>\blacktriangledown</td>
<td>▼</td>
<td>\diagup</td>
<td>\diagdown</td>
</tr>
</tbody>
</table>

More symbols are listed in sec. 16.4.
Some symbols can be displayed in different sizes, see sec. 11.4.

13.3. The Euro-Symbol €

To use the Euro symbol in formulas, the \LaTeX-package eurosym must be installed and loaded with the \LaTeX-preamble line
\usepackage[gennarrow]{eurosym}

The Euro symbol can now be inserted with the command \euro. The Euro symbol can directly be inserted with the € key in mathematical text, without having eurosym installed. When eurosym is installed, \euro can also be inserted in \LaTeX-mode. The official currency symbol can then be inserted with the command \officialeuro, that is only available in \LaTeX-mode.

An overview about the different Euro symbols:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>formula</td>
<td>\euro</td>
</tr>
<tr>
<td>mathematical text</td>
<td>€</td>
</tr>
<tr>
<td>\LaTeX-mode</td>
<td>\officialeuro</td>
</tr>
</tbody>
</table>
14. Relations

All relations can also be inserted via the toolbar button ≤.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
<th>Command</th>
<th>Result</th>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>=</td>
<td>=</td>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>\le</td>
<td>≤</td>
<td>\not=</td>
<td>≠</td>
<td>\ge</td>
<td>≥</td>
</tr>
<tr>
<td>\ll</td>
<td>≪</td>
<td>\equiv</td>
<td>≡</td>
<td>\gg</td>
<td>≫</td>
</tr>
<tr>
<td>\prec</td>
<td>≪</td>
<td>\sim</td>
<td>≍</td>
<td>\succ</td>
<td>≫</td>
</tr>
<tr>
<td>\preceq</td>
<td>≪</td>
<td>\simeq</td>
<td>≍</td>
<td>\succeq</td>
<td>≫</td>
</tr>
<tr>
<td>\subset</td>
<td>⊂</td>
<td>\approx</td>
<td>≈</td>
<td>\supset</td>
<td>⊃</td>
</tr>
<tr>
<td>\subseteq</td>
<td>⊆</td>
<td>\bowtie</td>
<td>⋈</td>
<td>\supseteq</td>
<td>⊑</td>
</tr>
<tr>
<td>\sqsubseteq</td>
<td>⊑</td>
<td>\in</td>
<td>∈</td>
<td>\ni</td>
<td>∋</td>
</tr>
<tr>
<td>\vdash</td>
<td>⊦</td>
<td>\propto</td>
<td>∝</td>
<td>\dashv</td>
<td>⊵</td>
</tr>
<tr>
<td>\smile</td>
<td>⊂</td>
<td>\asymp</td>
<td>≍</td>
<td>\frown</td>
<td>⊵</td>
</tr>
<tr>
<td>\lhd</td>
<td>△</td>
<td>\doteq</td>
<td>≐</td>
<td>\rhd</td>
<td>△</td>
</tr>
<tr>
<td>\unlhd</td>
<td>△</td>
<td>\circeq</td>
<td>≍</td>
<td>\unrhd</td>
<td>△</td>
</tr>
<tr>
<td>\gtrless</td>
<td>≻</td>
<td>\models</td>
<td>⊩</td>
<td>\lessgtr</td>
<td>△</td>
</tr>
<tr>
<td>\mid</td>
<td>{</td>
<td>\widehat =</td>
<td>≈</td>
<td>\parallel</td>
<td></td>
</tr>
<tr>
<td>\nmid</td>
<td>}</td>
<td></td>
<td></td>
<td>\nparallel</td>
<td>{</td>
</tr>
</tbody>
</table>

The characters \lhd and \rhd are bigger than the equal looking operators \triangleleft and \triangleright, respectively.

Relations are, in contrary to symbols, always surrounded by space.

Relations with labels can be created with the command \stackrel:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(r)\stackrel␣r\to\infty␣\approx␣␣B</td>
<td>A(r) \approx B</td>
</tr>
</tbody>
</table>

15. Functions

15.1. Predefined Functions

In general, variables are set italic in mathematical expressions, but not function names, because sin could be misunderstood as s · i · n. Therefore there are predefined functions, that are additionally a bit separated from prefactors. They are inserted as commands starting with a backslash before their name.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asin(x)+B</td>
<td>Asin(x) + B</td>
<td>A\sin(x)+B</td>
<td>A\sin(x) + B</td>
</tr>
</tbody>
</table>
The following functions are predefined:

<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
<th>Command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>\sin</td>
<td>\sinh</td>
<td>\arcsin</td>
<td>\sup</td>
</tr>
<tr>
<td>\cos</td>
<td>\cosh</td>
<td>\arccos</td>
<td>\inf</td>
</tr>
<tr>
<td>\tan</td>
<td>\tanh</td>
<td>\arctan</td>
<td>\lim</td>
</tr>
<tr>
<td>\cot</td>
<td>\coth</td>
<td>\arg</td>
<td>\liminf</td>
</tr>
<tr>
<td>\sec</td>
<td>\min</td>
<td>\deg</td>
<td>\limsup</td>
</tr>
<tr>
<td>\csc</td>
<td>\max</td>
<td>\det</td>
<td>\Pr</td>
</tr>
<tr>
<td>\ln</td>
<td>\exp</td>
<td>\dim</td>
<td>\hom</td>
</tr>
<tr>
<td>\lg</td>
<td>\log</td>
<td>\ker</td>
<td>\gcd</td>
</tr>
</tbody>
</table>

They can also be inserted with the math toolbar button \exp \tan.

### 15.2. Self-defined Functions

To use a function that is not predefined, like for example the sign function \( \text{sgn}(x) \), there are two possibilities:

- Define the function by inserting the following line to the \LaTeX-preamble

  \[ \DeclareMathOperator{\text{sgn}}{\text{sgn}} \]

  Now the new defined function can be called with the command \text{sgn}.

- Write the the formula as usual, mark the formula name, in our example the letters \text{sgn}, and change it to mathematical text. At last a space is inserted between prefactor and function.

The result is the same with both methods as with a predefined function:\[A \text{sgn}(x) + B\] :

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A \text{sgn}(x) + B )</td>
<td>( A \text{sgn}(x) + B )</td>
</tr>
<tr>
<td>( A \underbrace{\text{sgn}(x)}_{\text{Alt+M M}} + B )</td>
<td>( A \text{sgn}(x) + B )</td>
</tr>
</tbody>
</table>

The first method is more suitable when the self-defined function should be used several times.

---

\[26\text{For more about } \text{\texttt{\textbackslash DeclareMathOperator}} \text{ see sec. 10.4.}\]

\[27\text{In } \text{\LaTeX} \text{ self-defined functions are displayed red, predefined ones black.}\]
15.3. Limits

For limits there are defined besides $\lim$, $\liminf$ and $\limsup$ furthermore the following functions:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varliminf$</td>
<td>$\lim$</td>
</tr>
<tr>
<td>$\varlimsup$</td>
<td>$\lim$</td>
</tr>
<tr>
<td>$\varprojlim$</td>
<td>$\lim$</td>
</tr>
<tr>
<td>$\varinjlim$</td>
<td>$\lim$</td>
</tr>
</tbody>
</table>

The limit is created by inserting a subscript. It is set right beside the function in an inline formula:

$$\lim_{x \to A} x = B$$

In a displayed formula the limit is set below the formula, as usual:

$$\lim_{x \to A} x = B$$

15.4. Modulo-Functions

The modulo-function is special, because it exists in four variants. The variants in a displayed formula:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \mod b$</td>
<td>$a \mod b$</td>
</tr>
<tr>
<td>$a \pmod b$</td>
<td>$a \pmod b$</td>
</tr>
<tr>
<td>$a \bmod b$</td>
<td>$a \bmod b$</td>
</tr>
<tr>
<td>$a \pod b$</td>
<td>$a \pmod b$</td>
</tr>
</tbody>
</table>

In an inline formula less space is set before the function names for all variants.
16. Special Characters

16.1. Special Characters in Mathematical Text

The following commands can only be used in mathematical text or in \TeX-mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\oe</td>
<td>ë</td>
</tr>
<tr>
<td>\OE</td>
<td>Õ</td>
</tr>
<tr>
<td>\ae</td>
<td>æ</td>
</tr>
<tr>
<td>\AE</td>
<td>Æ</td>
</tr>
<tr>
<td>\aa</td>
<td>å</td>
</tr>
<tr>
<td>\AA</td>
<td>Å</td>
</tr>
<tr>
<td>\i</td>
<td>î</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\o</td>
<td>ø</td>
</tr>
<tr>
<td>\O</td>
<td>Ø</td>
</tr>
<tr>
<td>\l</td>
<td>ł</td>
</tr>
<tr>
<td>\L</td>
<td>Ł</td>
</tr>
<tr>
<td>!`</td>
<td>¡</td>
</tr>
<tr>
<td>?`</td>
<td>¿</td>
</tr>
<tr>
<td>\j</td>
<td>ĵ</td>
</tr>
</tbody>
</table>

The characters Å and Ø can also be inserted via the math toolbar button \mathcal{F}.

An exception are the commands !' and ?', because they can be inserted in \LaTeX directly to text.

16.2. Accents in Text

With the following commands all letters can be accented. The commands must be inserted in \TeX-mode.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;e</td>
<td>ë</td>
</tr>
<tr>
<td>\e</td>
<td>è</td>
</tr>
<tr>
<td>^e</td>
<td>ê</td>
</tr>
<tr>
<td>=e</td>
<td>¯è</td>
</tr>
<tr>
<td>u.e</td>
<td>ë</td>
</tr>
<tr>
<td>b.e</td>
<td>ë</td>
</tr>
<tr>
<td>t.ee</td>
<td>ee</td>
</tr>
<tr>
<td>\H.e</td>
<td>ě</td>
</tr>
<tr>
<td>\e</td>
<td>é</td>
</tr>
<tr>
<td>¬e</td>
<td>¯è</td>
</tr>
<tr>
<td>\v,e</td>
<td>ë</td>
</tr>
<tr>
<td>\d,e</td>
<td>ě</td>
</tr>
<tr>
<td>\c,e</td>
<td>ě</td>
</tr>
</tbody>
</table>

With the command \t also two different characters can be accented. The command \t \sz creates: ñz.

The accents ' , ' , and ^ can in combination with vowels directly be inserted with the keyboard without using \TeX-mode. The same applies for the tilde in combination with a , n , or o.

The commands \b , \c , \d , \H , \t , \u , \v , and accents inserted directly with the keyboard are also available in mathematical text. For the other accents there are special math commands to be used in formulas, see sec. 7.1.

\(^{28}\)This only applies for keyboards where the tilde is defined as accent.
Furthermore, with the command \textcircled{} all numbers and letters can be set into a circle, quasi accented with a circle, similar to the copyright symbol.

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textcircled{w}</td>
<td>( w )</td>
</tr>
<tr>
<td>\Large \textcircled{\normalsize\protect\raisebox{-1.5pt}{W}}</td>
<td>( W )</td>
</tr>
</tbody>
</table>

One has to take care that the character fits in the circle. \Large^{29} specifies thereby the size of the circle. With the help of \raisebox\(^{30}\) the character can be centered.

### 16.3. Minuscule Numbers

Minuscule numbers are created with the command \oldstylenums. The command can be used in formulas and in \TeX-mode. The command scheme is:

\oldstylenums{number}

The command \oldstylenums{0123456789} produces: 0123456789

### 16.4. Miscellaneous special Characters

The following characters can only be inserted to formulas by using commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>^ _</td>
<td>^ 😄</td>
</tr>
<tr>
<td>^ \circ</td>
<td>° ☺</td>
</tr>
</tbody>
</table>

The degree sign ° can nevertheless be directly inserted if the \TeX-preamble contains the following line\(^{31}\):

\DeclareInputtext{176}{\ifmmode^{\circ}\else\textdegree\fi}

---

\(^{29}\)see sec. 11.4

\(^{30}\)see sec. 9.2

\(^{31}\)More about this is described in sec. 23.10.
17. Formula Styles

- There are two different alignment styles:

  **Centered** is the predefined standard

  **Indented** for this the option `fleqn` must be inserted in the menu **Document > Settings under Document Class**

  When **Indented** is used, the indentation can be adjusted with the length `\mathindent`. Should the distance be 15mm, the following command line is inserted in the \LaTeX-preamble

  `\setlength{\mathindent}{15mm}`

  When no length is specified, the predefined value of 30 pt will be used.

- And two different numbering styles:

  **Right** is the predefined standard

  **Left** for this the option `leqno` must be inserted in the menu **Document > Settings under Document Class**

  `fleqn` and `leqno` can also be used together. In this case both options are inserted, separated by a comma.

  The chosen styles are used for all displayed formulas of the document. When both, centered and indented formulas should be created in a document, the style **Centered** is used. The indented formulas are then set in a `flalign` environment, see sec. 18.2.3.

18. Multiline Formulas

18.1. General

In \LaTeXX multiline formulas are created by pressing **Ctrl+Return** inside a formula. This creates either an **eqnarray environment** that is described in sec. 18.3 or, when the option **Use AMS math package** in the document settings is selected, an **align environment** that is described in sec. 18.2.1.

There are other multiline formula environments that can be created via the menu **Insert > Math**. These environments are described in the following sections.

In all multiline formula environments a new line is created by pressing **Ctrl+Return**. To add or delete lines, the math toolbar buttons or , respectively, or the menu **Edit > Rows & Columns** can be used.
18.1.1. Line Separation

There is sometimes not enough space in multiline formulas between the lines:

\[
B^2 (B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + 4x_0^2 x^2 + 4x_0 x D = -4x^2 B^2 + 4x_0 x B^2
\]
\[
4x^2 \left( B^2 + x_0^2 \right) + 4x_0 x \left( D - B^2 \right) + B^2 \left( B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2 \right) = 0
\]

In \LaTeX\ additional line space is specified as optional argument of the new line command. This is not yet possible in LyX\textsuperscript{32}, therefore the whole formula must be inserted in \TeX\-mode. To add in our example space, the command \textbackslash \texttt{[3mm]} is inserted at the end of the first line. One gets:

\[
B^2 (B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + 4x_0^2 x^2 + 4x_0 x D = -4x^2 B^2 + 4x_0 x B^2
\]
\[
4x^2 \left( B^2 + x_0^2 \right) + 4x_0 x \left( D - B^2 \right) + B^2 \left( B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2 \right) = 0
\]

To set the the line separation for all lines in a formula, the length \textbackslash \texttt{jot} is changed. The definition is: line separation = 6 pt + \textbackslash \texttt{jot}. Predefined for \textbackslash \texttt{jot} is the value 3 pt. To create 3 mm additional line separation as in the previous example, the command

\texttt{\setlength{\textbackslash jot}{3mm+3pt}}

is inserted in \TeX\-mode before the formula. This requires that the \LaTeX\-package \texttt{calc}\textsuperscript{33} was loaded in \LaTeX\-preamble with the line

\texttt{\usepackage{calc}}

One gets:

\[
B^2 (B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + 4x_0^2 x^2 + 4x_0 x D = -4x^2 B^2 + 4x_0 x B^2
\]
\[
4x^2 \left( B^2 + x_0^2 \right) + 4x_0 x \left( D - B^2 \right) + B^2 \left( B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2 \right) = 0
\]

To get back to the predefined distance, \textbackslash \texttt{jot} is set to the value 3 pt.

18.1.2. Column Separation

Multiline formulas form a matrix. A formula in the \texttt{eqnarray} environment is for example a matrix with three columns. By changing the column separation in this environment, the space beside the relation sign can be changed.

\textsuperscript{32} \texttt{see LyX-bug \#1505}

\textsuperscript{33} \texttt{calc} is part of every \LaTeX\ standard installation.
The column separation is specified with the length \texttt{\arraycolsep} according to:
\[ \text{column separation} = 2 \texttt{\arraycolsep} \]
Thus, the command
\begin{verbatim}
\setlength{\arraycolsep}{1cm}
\end{verbatim}
inserted in \TeX\-mode, sets for all following formulas a column separation of 2 cm. To get back to the predefined distance, \texttt{\arraycolsep} is set to 5 pt.

A formula with 2 cm column separation:
\[
A = B \\
C \neq A
\]
A formula with the predefined column separation for matrices of 10 pt:
\[
A = B \\
C \neq A
\]

\subsection*{18.1.3. Long Formulas}

Long formulas can be typeset using these methods:
\begin{itemize}
  \item When one side of the equation is much shorter than the line width, this one is chosen for the left side and the right side is typeset over two lines:
  \begin{equation}
  H = W_{SB} + W_{mv} + W_D - \frac{\hbar^2}{2m_0} \Delta - \frac{\hbar^2}{2m_1} \Delta_1 - \frac{\hbar^2}{2m_2} \Delta_2 - \frac{e^2}{4\pi\varepsilon_0 |r - \mathbf{R}_1|} \\
  - \frac{e^2}{4\pi\varepsilon_0 |r - \mathbf{R}_2|} + \frac{e^2}{4\pi\varepsilon_0 |\mathbf{R}_1 - \mathbf{R}_2|}
  \end{equation}
  The minus sign at the beginning of the second line does normally not appear as operator because it is the first character of the line. Thus it would not be surrounded by space and could not be distinguished from the fraction bar. To avoid this, 3 pt space was inserted behind the minus sign with the command \texttt{\hspace}.\footnote{more about \texttt{\hspace} see sec. 8.2}
  \item When both sides of the equation are too long, the command \texttt{\lefteqn} is used. It is inserted to the first column of the first line and effects that all further insertions overwrite the following columns:
  \begin{align}
  4x^2 (B^2 + x_0^2) + 4x_0 x (D - B^2) + B^2 (B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + D^2 \\
  - B^2 - 2B \sqrt{r_g^2 - x^2 + 2x_0 x - x_0^2} + r_g^2 - x^2 + 2x_0 x - x_0^2 \\
  = B^2 + 2 \left(r_g^2 + 2x_0 x - x_0^2 - r_k^2\right) + \frac{\left(r_g^2 + 2x_0 x - x_0^2 - r_k^2\right)^2}{B^2}
  \end{align}
\end{itemize}

\footnote{more about \texttt{\hspace} see sec. 8.2}
After the insertion of \texttt{\textbackslash lefteqn}, the cursor is in a purple box that is a bit shifted to the left from the blue one. In this the formula is inserted.

The content of the further lines is inserted to the second or another formula column. The greater the column number where it was inserted, the larger the indentation.

Note the following when using \texttt{\textbackslash lefteqn}:

* The formula doesn’t use the full page width. When e.g. the term $-B^2$ is added to the first line in the above example, it would have been outside the page margin. To better use the width, negative space can be inserted at the beginning of the first line.

* Due to a bug in \texttt{LyX} the cursor cannot be set with the mouse into the first line.\footnote{LyX-bug \#1429} One can only set the cursor at the beginning of the line and move it with the arrow keys.

* Other methods to set long formulas are offered by the environments described in sec. 18.5 and sec. 18.6.

\subsection*{18.1.4. Multiline Brackets}

For brackets spanning multiple lines the following problem occurs:

\begin{align*}
  A &= \sin(x) \left[ \prod_{R=1}^{\infty} \frac{1}{R} + \cdots \\
  &\quad \cdots + B - D \right]
\end{align*}

The closing bracket is smaller than the opening bracket because brackets with variable size may not span multiple lines.

To set the bracket size for the second line correctly, the first line is ended with \texttt{\textbackslash right}. and the second line with \texttt{\textbackslash left}.\footnote{for more about \texttt{\textbackslash left} and \texttt{\textbackslash right} see sec. 5.1.2} After \texttt{\textbackslash left.} the command \texttt{\vphantom{\prod_{\infty}^{1}}\textbackslash infty_{\textbackslash R=1}} is inserted, because the multiplication operator with its limits is the largest symbol in the first line and this should be the size for the bracket in the second line.

The result is this:

\begin{align*}
  A &= \sin(x) \left[ \prod_{R=1}^{\infty} \frac{1}{R} + \cdots \\
  &\quad \cdots + B - D \right]
\end{align*}

\footnote{for more about \texttt{\textbackslash left} and \texttt{\textbackslash right} see sec. 5.1.2}
18.2. Align Environments

Align environments can be used for every kind of multiline formulas. They are specially useful to set several formulas side by side.

Align environments consist of columns. The odd columns are right aligned, the even ones left aligned. Every line in an Align environment can be numbered.

Align environments are created via the menu Insert \textgreater Math. With the menu Edit \textgreater Math \textgreater Change Formula Type already existing formulas can be converted to Align environments.

To add or delete columns, the math toolbar buttons or , respectively, or the menu Edit \textgreater Rows & Columns can be used.

18.2.1. Standard align Environment

This Align environment is created by pressing Ctrl+Return in a formula or by the menu Insert \textgreater Math \textgreater AMS align Environment.

An example for two formulas set side by side, that are created with a four column align environment:

\[
A = \sin(B) \quad C = D
\]
\[
C \neq A \quad B \neq D
\]

As it can be seen, the formulas in this environment are placed so as if there would be a \textbackslash hfill before the first and after every even column. When the formula style Indented \textsuperscript{38} is used, the formula is set without the \textbackslash hfill before the first column.

18.2.2. Alignat Environment

The alignat environment has no predefined column separation. It can be inserted manually with the spaces that are described sec. 8.

The above example in the alignat environment where 1 cm space was inserted at the beginning of the second formula:

\[
A = \sin(B) \quad C = D
\]
\[
C \neq A \quad B \neq D
\]

Because the column separation can be set separately for every column, this environment is especially suitable to set three and more formulas side by side.

\textsuperscript{37} more about \textbackslash hfill see sec. 8.2
\textsuperscript{38} formula styles see sec. 17
18.2.3. Flalign Environment

In this environment the first two columns are always set as much as possible to the left and the last two ones to the right. An example:

\[
\begin{align*}
A &= 1 & B &= 2 & C &= 3 \\
X &= -1 & Y &= -2 & Z &= 4
\end{align*}
\]

By creating a flalign environment with an odd number of columns where an empty \TeX-brace is inserted to the last column, several formulas in a document can be set to the left, although the formula style Centered is used. As example the indented formula (5):

\[
\int_X dV = U
\]  \hfill (9)

The first two columns contain the formula. To indent it as with the formula style Indented, 30 pt space was inserted at the beginning of the first column.

18.3. Eqnarray Environment

When this environment has been created, three blue boxes appear. The content of the first box is right aligned, the content of the last one left aligned. The content of the middle box appears centered and a bit smaller, because it is designed to insert there only relation characters.

\[
\begin{align*}
ABC & \quad ABC & \quad ABC \\
D & \quad D & \quad D \\
AB & \quad AB & \quad AB \\
A & = & A
\end{align*}
\]

18.4. Gather Environment

This environment consists of only one centered column. Every line can be numbered.

\[
\begin{align*}
A &= 1 \\
X &= -1
\end{align*}
\]  \hfill (10), (11)

18.5. Multline Environment

The multline environment consists, like the gather environment, of only one column. But the first line is left aligned, the last one right aligned. All other lines are centered.
Therefore this environment is suitable for long formulas. As example formula (8) in the multiline environment:

\[ 4x^2(B^2 + x_0^2) + 4x_0x(D - B^2) + B^2(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + D^2 \]
\[ - B^2 - 2B\sqrt{r_g^2 - x^2 + 2x_0x - x_0^2 + r_g^2 - x^2 + 2x_0x - x_0^2} \]
\[ = B^2 + 2\left(r_g^2 + 2x_0x - x_0^2 - r_k^2\right) + \frac{\left(r_g^2 + 2x_0x - x_0^2 - r_k^2\right)^2}{B^2} \] (12)

In the output only the last (first) line of a multiline environment appears numbered when the document numbering is right (left).\(^{39}\)

With the commands `\shoveright` and `\shoveleft` a centered line can be right or left aligned, respectively. The commands are used as follows:

`\shoveright{line content}` and `\shoveleft{line content}`

The length `\multlinegap` specifies the distance of the first line from the left page margin. Predefined is the length 0 pt.

As example the above formula where the command

`\setlength{\multlinegap}{2cm}`

was inserted in \TeX-mode before:

\[ 4x^2(B^2 + x_0^2) + 4x_0x(D - B^2) + B^2(B^2 - 2r_g^2 + 2x_0^2 - 2r_k^2) + D^2 \]
\[ - B^2 - 2B\sqrt{r_g^2 - x^2 + 2x_0x - x_0^2 + r_g^2 - x^2 + 2x_0x - x_0^2} \]
\[ = B^2 + 2\left(r_g^2 + 2x_0x - x_0^2 - r_k^2\right) + \frac{\left(r_g^2 + 2x_0x - x_0^2 - r_k^2\right)^2}{B^2} \] (13)

The second line was left aligned using `\shoveleft`.

### 18.6. Multiline Formula Parts

To display only parts of a formula with multiple lines, one of the following environments are used: `aligned`, `alignedat`, `gathered` or `split`. They can be inserted via the menu `Insert \(\rightarrow\) Math` or by using the commands described in this section.

The first three have the same properties as the corresponding multiline formula environments, but it is possible to set further formula parts beside them. An example:

\[
\begin{align*}
\Delta x \Delta p & \geq \frac{h}{2} \\
\Delta E \Delta t & \geq \frac{h}{2}
\end{align*}
\]

Uncertainty relations

\(^{39}\)numbering styles see sec. 17
To get this formula, a displayed formula is created where the command \texttt{aligned} is inserted. A purple box appears around the blue formula box where now columns and lines can be added. Outside the multiline environment other formula parts can be set, like the brace.

The aligned environment is also suitable for long formulas whose lines are horizontally aligned. Using aligned in a displayed formula has the advantage that the formula number is vertically centered behind the lines. As example formula (7) in the aligned environment:

\begin{equation}
H = W_{SB} + W_{mv} + W_D - \frac{\hbar^2}{2m_0} \Delta - \frac{\hbar^2}{2m_1} \Delta_1 - \frac{\hbar^2}{2m_2} \Delta_2 - \frac{e^2}{4\pi\varepsilon_0 |\mathbf{r} - \mathbf{R}_1|} - \frac{e^2}{4\pi\varepsilon_0 |\mathbf{r} - \mathbf{R}_2|} + \frac{e^2}{4\pi\varepsilon_0 |\mathbf{R}_1 - \mathbf{R}_2|}
\end{equation}

To use the environments \texttt{alignedat}, \texttt{gathered}, or \texttt{split}, the command \texttt{alignedat}, \texttt{gathered}, or \texttt{split} are inserted, respectively. The split environment has the same properties as the aligned environment but it can only have two columns.

\textbf{18.7. Text in multiline Formulas}

In the Align environments, and the multline and gather environment, text can be inserted that will appear in a separate line and doesn’t affect the column alignment. To do this, the command \texttt{\intertext} is used in the following scheme:

\texttt{\intertext\{text\}}

The text should not be longer than a line because it cannot be hyphenated. As LyX doesn’t yet support \texttt{\intertext} directly, the text is written as mathematical text. \texttt{\intertext} must hereby be at the beginning of a line and appears in the output above this line. An example where the text was inserted at the beginning of the second line:

\begin{equation}
I = a\sqrt{2} \int_0^{2\pi} \sqrt{1 + \cos(\phi)} \, d\phi
\end{equation}

integrand is symmetric to $\phi = \pi$, therefore

\begin{equation}
= 2a\sqrt{2} \int_0^{\pi} \sqrt{1 + \cos(\phi)} \, d\phi
\end{equation}

\textbf{19. Formula Numbering}

\textbf{19.1. General}

Numbered formulas can be created with the menu Insert $\triangleright$ Math $\triangleright$ Numbered Formula (shortcut Ctrl+Alt N). Existing formulas can be numbered with the menu Edit $\triangleright$.
Math → Toggle Numbering (shortcut Alt+M N). The formula number is displayed in \LaTeX{} behind the formula as number sign in parentheses. The number sign is replaced in the output by the formula number.

When numbering is turned on in multiline formulas, all lines will be numbered. But the numbering can be controlled with the menu Edit → Math → Toggle Numbering of Line (shortcut Alt+M Shift+N) for every line.

Except for inline formulas, all formulas can be numbered with two different styles, see sec. 17.

### 19.2. Cross-References

All labeled formulas can be cross-referenced. A label is added by the menu Insert → Label or the toolbar button \(\mathbb{L}\). The cursor must hereby be inside a displayed formula. A dialog pops up displaying the prefix \texttt{eq:} in a text field. The label is inserted there behind the prefix. The predefined prefix means “equation” and makes it easier to find labels in large documents because it marks it as formula label to divide it from e.g. section labels. To change a label, the menu Insert → Label is used again.

The name of the label is displayed in \LaTeX{} within two parentheses behind formula. A formula with a label is always numbered.

Cross-references are inserted via the menu Insert → Cross-Reference or with the toolbar button \(\mathbb{F}\). A formula cross-reference appears in the output as formula number. When in the cross-reference dialog window the format \texttt{(<reference>)} is chosen, the cross-reference appears in the output as formula number in parentheses. By right-clicking on a cross-reference in \LaTeX{}, one jumps to the formula that is referenced.

Here are as examples cross-references to formulas of the following subsections:

The equations (something) and (17b) are equivalent. In (W) big Latin letters are used for the numbering in contrary to (XXI).

When the argument of \texttt{\tag{40}} contains a box like in sec. 9.4, the formula cannot be referenced.

### 19.3. Subnumbering

With the help of the commands \texttt{\begin{subequations}} and \texttt{\end{subequations}} formulas can be subnumbered. Both commands are inserted in \TeX{}-mode.

An example:

\[ A = C - B \quad (17) \]

\(\tag{40}\) \texttt{\tag{}} is described in sec. 19.4.
\[ B = C - A \quad (17a) \]
\[ C = A + B \quad (17b) \]

To create the example, the following is done:

1. first formula is inserted
2. \texttt{\addtocounter{equation}{-1} \begin{subequations}} is inserted after the first formula
3. second formula is inserted
4. third formula is inserted
5. \texttt{\end{subequations}} is inserted after the third formula

Every formula between the commands \texttt{\begin} and \texttt{\end} is subnumbered as a, b, c, \ldots For multiline formulas every line will be subnumbered. All subnumbered formulas are treated as one numbered formula. But as every numbered formula increases the counter \texttt{equation} by one, the command \texttt{\addtocounter} is needed to decrease it. Otherwise the formulas (17), (17a), (17b) would be numbered as (17), (18a), (18b).

By inserting the commands in \TeX-mode, a space is created between the first two formulas. To revert this -5\,mm vertical space is inserted after the command \texttt{\begin{subequations}}. When the formula style \texttt{Indented} is used, -7\,mm space is inserted instead.

Here is an example for a multiline formula where the numbering was turned off for the second line:

\[ A = (B - Z)^2 = (B - Z)(B - Z) \quad (18a) \]
\[ = B^2 - ZB - BZ + Z^2 \]
\[ = B^2 - 2BZ + Z^2 \quad (18b) \]

\textbf{19.4. User-defined Numbering}

With the standard numbering parentheses are set around the formula number. To replace the parentheses for example by vertical bars, the following line is added to the \LaTeX-preamble:

\texttt{\def\tagform@#1{\maketag@@@{|#1|}}}

To use other characters, the vertical bars besides the \texttt{#1} are replaced by one ore more characters. To get only the formula number the vertical bars are omitted.

\footnote{formula styles see sec. 17}
When there should be an expression of your choice instead of the consecutive formula number in parentheses behind the formula, the command \texttt{\textbackslash tag} is used:

\[
A + B = C \quad \text{(something)}
\]

In this example the command \texttt{\textbackslash tag\_something} was inserted to the formula.

When the command \texttt{\textbackslash tag\*\_something} is inserted instead, the star prevents the parentheses around the expression:

\[
A + B = C \quad \text{something}
\]

To restart the formula numbering with new document parts or sections, the following command is used:

\texttt{\textbackslash @addtoreset{equation}\{part\}}

dep.

\texttt{\textbackslash @addtoreset{equation}\{section\}}

To be able to use these commands in L\TeX-mode, the “@” character has to be made “active” for L\TeX using the command \texttt{\textbackslash makeatletter}. The command \texttt{\textbackslash makeatother} reverts this. So the command sequence in L\TeX-mode is:

\texttt{\textbackslash makeatletter}

\texttt{\textbackslash @addtoreset{equation}\{section\}}

\texttt{\textbackslash makeatother}

In the L\TeX-preamble \texttt{\textbackslash makeatletter} and \texttt{\textbackslash makeatother} can be omitted as they are automatically internally inserted by L\TeX.

To revert \texttt{\textbackslash @addtoreset}, the file \texttt{remreset.sty}\textsuperscript{42} has to be loaded in the L\TeX-preamble with the line

\texttt{\usepackage{remreset}}

Then the command \texttt{\textbackslash @removefromreset} can be used with the same scheme as \texttt{\textbackslash @addtoreset}.

Sometimes formulas should be numbered in the following form:

\[(\text{section number} . \text{formula number})\]

The formula number should start with every section with “1”.

For this case there is the command \texttt{\textbackslash numberwithin}, which is used with the following scheme:

\texttt{\textbackslash numberwithin\{counter\}\{sectioning\}}

\textsuperscript{42}\texttt{remreset} is part of the L\TeX-package \texttt{carlisle} that is part of every L\TeX standard installation.
Counter denotes what kind of numbering is affected, sectioning denotes what number is before the dot.

Thus in our case the following \texttt{\LaTeX}-preamble or \TeX-Code line is used:
\[
\texttt{\numberwithin{equation}{section}}
\]
This is the result:
\[
A + B = C
\] (19.19)

To number e.g. tables so that the number of the part is the sectioning, \[
\texttt{\numberwithin{table}{part}}
\] is used.

To go back to the standard numbering or to prevent this kind of numbering when it is defined by the document class, the following command is inserted as \TeX-Code or to the \LaTeX-preamble:
\[
\texttt{\renewcommand{\theequation}{\arabic{equation}}}
\]
or
\[
\texttt{\renewcommand{\thetable}{\arabic{table}}}
\]
\texttt{\numberwithin} uses internally the command \texttt{@addtoreset}, described above, that also needs to be reverted.

19.5. Numbering with Roman Numbers and Letters

Formulas can also be numbered with Roman numbers and Latin letters. To number for example with small Roman numbers, the command
\[
\texttt{\renewcommand{\theequation}{\roman{equation}}}
\]
is inserted before the formula in \TeX-mode. \texttt{\renewcommand} redefines the predefined command \texttt{\theequation} to the command \texttt{\roman{equation}}.\footnote{\texttt{\renewcommand} has the same scheme like the command \texttt{\newcommand} that is described in sec. 22.1.} \texttt{equation} is the formula counter. When the command \texttt{\the} is used as prefix for a counter, the value of the counter is output as Arabic number. When a formula is numbered, \LaTeX sets internally the command \texttt{\theequation} behind the formula. \texttt{\roman{equation}} outputs the counter as small Roman number.

All formulas behind the command \texttt{\renewcommand} are now numbered Roman. To switch to numbering with big Roman numbers, the command is inserted again, but \texttt{\roman} is replaced by \texttt{\Roman}. To “number” with small Latin letters there is the command \texttt{\alph}, for big ones there is the command \texttt{Alph}.

\textbf{Note:} Only maximal 26 formulas can be numbered with Latin letters in one document.
\renewcommand{\theequation}{\arabic{equation}}

\medskip

E = Arabic

As you see, formulas are numbered serially independent from the numbering style. When then numbering should start with “1” when the style is changed, new equation counters have to be defined. A description about this can be found in the file Formula-numbering.lyx.

20. Chemical Symbols and Equations

An example text from chemistry:

The \text{SO}_4^{2-}-ion reacts with two \text{Na}^+ -ions to sodium sulfate (\text{Na}_2\text{SO}_4). The chemical equation for this is:

\begin{equation}
2 \text{Na}^+ + \text{SO}_4^{2-} \rightarrow \text{Na}_2\text{SO}_4
\end{equation}

This chemical equation can directly be created as formula. To avoid that the symbols appear italic, everything is highlighted and changed by the shortcut Alt+C R to the upright font style.\textsuperscript{44}

A more convenient way to typeset chemical formulas is to use the command \texttt{\textbackslash ce} that is available when the \LaTeX-package \texttt{mhchem} is installed. After inserting \texttt{\textbackslash ce} to a formula a new blue box appears where chemical formulas can be inserted in an intuitive way.

\textsuperscript{44}\small{font styles see sec.11.1}
<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{ce}_\text{H2CO3}$</td>
<td>$\text{H}_2\text{CO}_3$</td>
</tr>
<tr>
<td>$\text{ce}_\text{SO4}^2-$</td>
<td>$\text{SO}_4^{2-}$</td>
</tr>
<tr>
<td>$\text{ce}_\text{(NH4)2S}$</td>
<td>$(\text{NH}_4)_2\text{S}$</td>
</tr>
<tr>
<td>$\text{ce}_\text{KCr(SO4)2.12H2O}$</td>
<td>$\text{KCr(SO}_4\text{)}_2\cdot 12 \text{H}_2\text{O}$</td>
</tr>
<tr>
<td>$\text{ce}_\text{A-B_dbond_C_tbond_D}$</td>
<td>$\text{A-B=}\text{C=}\text{D}$</td>
</tr>
<tr>
<td>$\text{ce}_\text{^227_90_Th+}$</td>
<td>$^{227}_{90}\text{Th}^+$</td>
</tr>
<tr>
<td>$\text{ce}_\text{CO2_+_C_&lt;=&gt;_2CO}$</td>
<td>$\text{CO}_2 + \text{C} \rightleftharpoons 2 \text{CO}$</td>
</tr>
<tr>
<td>$\text{ce}_\text{CO2_+_C_&gt;_\alpha_beta_2CO}$</td>
<td>$\text{CO}_2 + \text{C} \xrightarrow{\alpha} \beta 2 \text{CO}$</td>
</tr>
</tbody>
</table>

**Note:** Inserting a formula to a \texttt{ce} box will lead to Li\TeX\ errors. In this case Li\TeX\ code has to be used like for \texttt{ce\{\$\mu\text{-Cl}\}}: $\mu$-Cl

Using \texttt{ce} the command for equation (25) is:
\texttt{ce\_2Na\_+\_\texttt{SO4}^2\_\rightarrow\_\texttt{Na2SO4}}

To create multiline chemical equations first a multiline formula is created as described in sec. 18. Afterwards the command \texttt{ce} is used in every small blue box of the formula. (26) and (27) are an example of a multi-stage chemical reaction where every equation has its own number.

\[
\text{TEOS} + 4 \text{O} \rightarrow \text{Si(OH)}_4 + 4 \text{C}_2\text{H}_4\text{O} \quad (26)
\]
\[
\text{Si(OH)}_4 \rightarrow \text{SiO}_2 + 2 \text{H}_2\text{O} \quad (27)
\]

Besides \texttt{ce} the \texttt{mhchem} package provides the command \texttt{cf} that has to be used for special cases. For more information about \texttt{cf} and more examples have a look at the documentation of \texttt{mhchem}, [7].

### 21. Diagrams

\LaTeX\ supports two types of commutative diagrams: \texttt{amscd} and \texttt{xymatrix} that are explained in the following.

#### 21.1. Amscd Diagrams

Diagrams of this type visualize relations by vertical and horizontal lines or arrows:

\[
\begin{array}{c}
A \rightarrow B \rightarrow C \\
\uparrow \quad \quad \quad \downarrow \\
F \leftarrow E \leftarrow D
\end{array}
\]
To get them, the command $\texttt{\CD}$ is inserted to a formula. A blue box appears with two dashed lines where further commands can be inserted. With Ctrl+Return a new line is created. Horizontal relations are inserted in odd, vertical in even formula lines.

To create the relations there are the following commands:

- @<<< creates a left arrow, @>>> a right arrow, and @= a long equal sign
- @AAA creates an up arrow, @VVV an down arrow, and @| a vertical equal sign
- @. is a placeholder for non-existent relations

All arrows can be labeled as follows:

- If text inserted between the first and second < or >, resp., it is placed above the arrow. When it is inserted between the second and third one, it appears under the arrow.
- When text for vertical arrows is inserted between the first and second A or V, resp., it is placed left beside the arrow. When it is inserted between the second and third one, it appears right beside the arrow. If the text contains an A or V, these letters must be set into a TEX-brace.

As example a diagram with all possible relations:

\[
\begin{array}{ccc}
A & j & \rightarrow & B & k & \rightarrow & C & = & F \\
D & j & \leftarrow & E & k & \rightarrow & F & \rightarrow & C \\
\end{array}
\]

The command for this is:

\[
\texttt{\CD A@>j>B@>k>C@=} F \text{ Ctrl+Return} \\
@AmAA@.VV\{V\rightarrow V\} \text{ Ctrl+Return} \\
D@<<j<E@>k>F@=} C
\]

21.2. Xymatrix Diagrams

To be able to use xymatrices, the \LaTeX-package \texttt{xypic} must be installed. A xymatrix is created by inserting the command \texttt{xymatrix} in a formula. Then you are able to add new matrix columns and rows like for normal matrices, see sec. 4.

In contrary to amscd diagrams, xymatrices supports diagonal and curved arrows, and much more. All possibilities to create commutative diagrams and decorations are explained in detail in the \texttt{XY-pic manual} that you find in the menu Help>Specific Manuals>XY-pic Manual.
22. User-defined Commands

Note: The names of user-defined commands and macros may only consist of Latin letters.

22.1. The Command \newcommand

Many \LaTeX-commands are too long to be used frequently. But it is possible to define with the command \newcommand new shorter commands.

The command scheme of \newcommand is:
\begin{quote}
\texttt{newcommand\{new\ command\ name\}[number\ of\ arguments][optional\ value]\{command\ definition\}}
\end{quote}

Note: Assure that the name of the new command is not already used in your document or by \LaTeX-packages that you use. When you for example define the command \texttt{\le} for \textbackslash \Leftarrow, you get an error message because \texttt{\le} is already defined as command for “≤”.

The number of arguments is an integer in the range 0-9 and specifies how many arguments the new command should have. With the optional value a value for an optional argument can be predefined. When this is done, the first argument of the new command is automatically an optional one.

Here are some examples:

- To define the command \texttt{\gr} for \textbackslash \Longrightarrow, the \LaTeX-preamble line is:
  \begin{quote}
  \texttt{\newcommand\{\gr\}\{\Longrightarrow\}}
  \end{quote}

- To define the command \texttt{\us} for \textbackslash \underline, the argument (that should be underlined) must be taken into account. For this the preamble line is:
  \begin{quote}
  \texttt{\newcommand\{\us\}[1]\{\underline\{#1\}\}}
  \end{quote}
  The character \# acts as argument placeholder, the 1 behind it denotes that it is the placeholder for the first argument.

- For \texttt{\framebox} one can e.g. define the command \texttt{\fb}:
  \begin{quote}
  \texttt{\newcommand\{\fb\}[3]\{\framebox\#1\#2\$\#3\$\}}
  \end{quote}
  The two Dollar signs creates the extra formula needed for \texttt{\framebox}, see sec. 9.1.

- To create a new command for \texttt{\fcolorbox} where the color for the box needn’t to be specified, the argument for the color is defined optional:
  \begin{quote}
  \texttt{\newcommand\{\cb\}[3][white]\{\fcolorbox\#2\{\#1\}\$\#3\$\}}
  \end{quote}
  When the color is not specified when using \texttt{\cb}, the predefined color \texttt{white} will be used.
A test of the new defined commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A\gr\ B</td>
<td>A \implies B</td>
</tr>
<tr>
<td>\us{ABcd}</td>
<td>ABcd</td>
</tr>
<tr>
<td>\fb[2cm]→{→{\int_A=B</td>
<td>\int A = B</td>
</tr>
<tr>
<td>\cb{red→}{\int_A=B</td>
<td>\int A = B</td>
</tr>
<tr>
<td>\cb[green]{red→}{\int_A=B</td>
<td>\int A = B</td>
</tr>
</tbody>
</table>

22.2. Math Macros

User-defined commands are especially convenient for complex expressions. When you are for example dealing in a document with quadratic equations, the same solution type occurs several times. The general form of a quadratic equation is

$$0 = \lambda^2 + p\lambda + q$$

The general form of the solution is

$$\lambda_{1,2} = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$$

To define a command for the solution formula where only the three parameters $\lambda$, $p$, and $q$ need to be specified and the index of $\lambda$ can be given optionally, the \LaTeX\-preamble line is

```
\newcommand{\qG}[4][1,\,2]{#2_{#1}=-\frac{#3}{2}\pm\sqrt{\frac{#3^{2}}{4}-#4}}
```

To create with this the solution formula, the command \qG{\lambda→\{p→\{q is inserted to a formula.

The definition of the new command is unintuitive because one has to know the schemes of all used \LaTeX\ commands, e.g. that a fraction is inserted in \LaTeX\ as \frac{numerator}{denominator}. Furthermore one can easily forget a brace in the definition and cannot see in \LYX what the new command is doing. To avoid these problems \LYX offers the possibility to use math macros instead of the command \newcommand.

A math macro is created by using the menu Insert \hspace{1em} Math \hspace{1em} Macro or the toolbar button \hspace{1em} . The math macro toolbar appears together with the following box where the macro is defined:
\newmacroname is the default name of the macro that should be changed to something sensible. The wanted formula is inserted in the first blue box. An argument placeholder is inserted with the command \#argumentnumber, e.g \#1 or by using the macro toolbar button \#. Argument placeholders are displayed red. Maximum 9 arguments are possible. Optional arguments are created with the toolbar button \$. The first non-optional argument can be transformed to an optional one with the toolbar button \$. In the second blue box the appearance of the macro in \LaTeX{} can be defined. Normally you want to see it as it is defined, so the box is kept empty. But when you have created a macro that needs lot of space on the screen, you can insert in the box for example \qG: \#1, \#2, \#3, \#4.

For the macro only the arguments with the macro name in front of them will then be displayed in \LaTeX{}, leading to a better overview. The formula appears in the output as defined in the first box.

The appearance of macros in formulas can furthermore be changed for single macros by setting the cursor in the macro and using the menu View> (Un)fold Math Macro.

To use a macro, the macro name is inserted as command to a formula, in our case \qG. Our macro looks in \LaTeX{} like this:

Here is our macro example with the arguments $x$, $\ln(x)$, and $B$:

$$x_{1,2} = -\frac{\ln(x)}{2} \pm \sqrt{\frac{\ln(x)^2}{4} - B}$$

\LaTeX{} offers in the menu Tools>Preferences>Editing>Control different styles to edit macros. To find the style that suits you the most, choose a style and set the cursor in a macro formula to see the difference.

A math macro is transformed internally to a \texttt{\newcommand} command when exporting the document. The created \texttt{\newcommand} command is not placed in the \LaTeX{}-preamble, therefore macros can only be used in formulas that are in the document below the macro definition box.

Math macros can also be directly be created from a \texttt{\newcommand} command. When writing for example the command

\texttt{\newcommand{\larrow}{2}{\xleftarrow[#2]{#1}}} in \LaTeX{} as normal text, highlighting it completely and using then the shortcut Ctrl+M,
the command will be transformed to a math macro. Using this method you need to be careful that the \newcommand command is typed correctly, otherwise you get a faulty macro leading to \LaTeX errors.

Math macros currently yet have the problem that further formulas in macro definitions are handled wrongly. Therefore the example \fb from sec.22.1 cannot be created as macro.

When the cursor is in a macro definition box, you will see the macro toolbar in \LaTeX:

![Macro Toolbar]

The macro toolbar contains from left to right the following buttons:

- Edit ▶ Math ▶ Macro Definition ▶ Remove Last Argument
- Edit ▶ Math ▶ Macro Definition ▶ Append Argument
- Edit ▶ Math ▶ Macro Definition ▶ Make First Non-Optional into Optional Argument
- Edit ▶ Math ▶ Macro Definition ▶ Make Last Optional into Non-Optional Argument
- Edit ▶ Math ▶ Macro Definition ▶ Remove Optional Argument
- Edit ▶ Math ▶ Macro Definition ▶ Insert Optional Argument
- Edit ▶ Math ▶ Macro Definition ▶ Remove Last Argument Spitting Out To The Right
- Edit ▶ Math ▶ Macro Definition ▶ Append Argument Eating From The Right
- Edit ▶ Math ▶ Macro Definition ▶ Append Optional Argument Eating From The Right
23. Tips

23.1. Negative Numbers

Negative numbers often look ugly in formulas because the minus sign before the number is set with the same length as the minus operator sign. When writing the negative number in normal text, the minus sign appears correctly. Thus, the problem disappears when converting the minus sign to mathematical text. An example to visualize the problem:

<table>
<thead>
<tr>
<th>normal text: ( x = -2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>formula: ( x = -2 )</td>
</tr>
<tr>
<td>solution: ( x = -2 )</td>
</tr>
</tbody>
</table>

23.2. Comma as decimal Separator

In \LaTeX\ a comma inside a formula is used, according to the English convention, as number group separator. So there will be space added behind all commas in formulas. To avoid this, the comma is highlighted and changed to mathematical text (shortcut Ctrl+M).

To use all formula commas in the document as decimal separator, the file `icomma.sty` is loaded with the \LaTeX-preamble line

\texttt{\usepackage{icomma}}

23.3. Physical Vectors

Predefined vectors are offered by the \LaTeX-package `braket` that is loaded with the \LaTeX-preamble line
\texttt{\usepackage{braket}}

The following commands are defined:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\Bra{\psi}</td>
<td>\langle \psi</td>
</tr>
<tr>
<td>\Ket{\psi}</td>
<td>\rangle \psi</td>
</tr>
<tr>
<td>\Braket{\psi</td>
<td>\phi}</td>
</tr>
</tbody>
</table>

---

45 Other useful math tips can be found in [3].
46 `icomma` is part of the \LaTeX-package `was`.
47 `braket` should be part of every \LaTeX standard installation.
The command \textbackslash Braket ensures that all vertical bars are set in the size of the surrounding brackets:

\[ \langle \phi \mid J = \frac{3}{2}, M_J \rangle \]

The effect of \textbackslash Braket can also be achieved using the command \textbackslash middle, that is described in sec. 5.1.2.

### 23.4. Self-defined Fractions

To define custom commands for fractions, the command \textbackslash genfrac is used in the following scheme:

\[ \text{\textbackslash genfrac\{left bracket\}\{right bracket\}\{fraction bar thickness\}\{style\}\{numerator\}\{denominator\}} \]

The style is a number in the range of 0 - 3.

<table>
<thead>
<tr>
<th>Number</th>
<th>Style (Size)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>display style formula</td>
</tr>
<tr>
<td>1</td>
<td>inline formula</td>
</tr>
<tr>
<td>2</td>
<td>small</td>
</tr>
<tr>
<td>3</td>
<td>tiny</td>
</tr>
</tbody>
</table>

When no style is given, the size is adjusted to the surrounding environment like for the command \textbackslash frac.

When no fraction bar thickness is given, the predefined value of 0.4 pt will be used.

For example, the commands \textbackslash dfrac and \textbackslash tbinom from sec. 3.2 are defined with the commands

\[ \text{\newcommand\{\textbackslash dfrac\}\{2\}\{\textbackslash genfrac\}\{}\{}\{0\}\{\#1\}\{\#2\} \]

and

\[ \text{\newcommand\{\textbackslash tbinom\}\{2\}\{\textbackslash genfrac\}\{}\{}\{0pt\}\{1\}\{\#1\}\{\#2\} \]

To define a fraction where the fraction bar thickness can be given as optional argument, the following line is inserted to the \LaTeX-preamble:

\[ \text{\newcommand\{\textbackslash fracS\}\{3\}\{}\{}\{\textbackslash genfrac\}\{}\{}\{\#1\}\{\#2\}\{\#3\} \]

433
A test:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{S[1\text{mm}]}{A \to {B} )</td>
<td>( A )</td>
</tr>
</tbody>
</table>
| \( \frac{S[5\text{mm}]}{A \to \{B} \) | \( A 
\)

As one can see, the distance of the numerator and the denominator to the fraction bar is round about three times the bar thickness.

### 23.5. Canceled Formulas

To cancel formulas or formula parts, the \LaTeX-package \texttt{cancel} has to be loaded with the \LaTeX-preamble line

\begin{verbatim}
\usepackage[samesize]{cancel}
\end{verbatim}

There are four ways to cancel formulas:

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>\cancel{\int A=B}</td>
<td>( \int A=B )</td>
</tr>
<tr>
<td>\bcancel{\int A=B}</td>
<td>( \int A=B )</td>
</tr>
<tr>
<td>\xcancel{\int A=B}</td>
<td>( \int A=B )</td>
</tr>
<tr>
<td>\cancelto{1\rightarrow{\int A=B}</td>
<td>( \int A=B^{-1} )</td>
</tr>
</tbody>
</table>

\texttt{cancelto} is especially suitable to visualize the reduction of fractions within formulas:

\[
\frac{(x_0 + bB)^2}{(1 + b^2)^d} = \frac{x_0^2 + B^2 - r_0^2}{1 + b^2}
\]

### 23.6. Formulas in Section Headings

When formulas are used in section headings, the following has to be taken into account:

\footnote{\texttt{cancel} is part of every \LaTeX standard installation.}

434
When \texttt{hyperref} support is enabled in the document settings dialog under \texttt{PDF Properties}, PDF-bookmarks are created for every section heading in the table of contents. If a section heading contains formulas, they are incorrectly displayed in the bookmark text, because formulas in bookmarks infringe the PDF conventions.

Both problems can be solved by inserting at the end of the section heading a short title with the menu \texttt{Insert} $\rightarrow$ \texttt{Short Title}. Short titles are used as alternative for multiline section headings to keep the table of contents clearly arranged. Only the short title appears in the table of contents and therefore also in the PDF-bookmark.

When formulas should be used in the table of contents but \texttt{hyperref} is used, one can use the following command in \TeX-mode:

\begin{verbatim}
\texttt{\textorpdfstring{part}{alternative}}
\end{verbatim}

Part is the part of the heading that shouldn’t appear in the PDF-bookmark. This can be characters, formulas, footnotes, but also cross-references. The alternative is used instead of the part for the bookmark.

Here are two example headings:

\begin{itemize}
  \item \textbf{23.6.1. Heading without formula in table of contents} $\sqrt{-1} = i$
  \item \textbf{23.6.2. Heading with formula in table of contents} $\sqrt{-1} = i$
\end{itemize}

In the first heading a short title was used, in the second one \texttt{\textorpdfstring}. To get the same formatting as for the other headings, the complete heading was set into a \texttt{boldmath} environment\footnote{see sec.11.2}.

\section*{23.7. Formulas in multi-column Text}

Formulas in multi-column text are often too wide to fit into a column and thus need to be set over the whole page width. This is done by using the \LaTeX-package \texttt{multicol}\footnote{\texttt{multicol} is part of every \LaTeX standard installation}, that is loaded with the \LaTeX-preamble line

\begin{verbatim}
\usepackage{multicol}
\end{verbatim}

Note hereby that the setting \texttt{Two-column document} in the menu \texttt{Document} $\rightarrow$ \texttt{Settings} under \texttt{Text Layout} must \textit{not} be selected.

Before the multi-column text the command

\begin{verbatim}
\begin{multicols}{column number}
\end{verbatim}

is written in \TeX-mode. The column number is a number in the range of 2-10. Before the formula the multi-column text is ended by inserting the command

\begin{verbatim}
\end{verbatim}
Due to the command some space is automatically added before the formula. To revert this, \(-6\text{mm}\) vertical space is inserted before the formula. When the formula style \textbf{Indented}\textsuperscript{51} is used, \(-9\text{mm}\) space is inserted instead.

As example a multi-column text with a displayed formula:

Das Spektrum wird fouriertransformiert. Die Fouriertransformation wird verwendet, um die überlagerten Signale (Netzwerk, Lösungsmittel) zu trennen. Nachdem wir die Phasenverschiebung bestimmen konnten, interessiert uns nun das Aussehen des Ausgangssignals. Im Experiment haben wir es mit sehr vielen Teilchen zu tun, so dass man über alle Phasen integrieren muss. Sei nun \(S\) unser normiertes Ausgangssignal und \(P\) die Phasenverteilungsfunktion, so ergibt sich die Beziehung

\[
S(t) = S_0(t) \int_{-\infty}^{\infty} P(\phi, t)e^{i\phi} d\phi
\]  

(28)

wobei \(S_0\) das Signal ohne Gradient ist und die Normierungsbedingung \(\int_{-\infty}^{\infty} P(\phi, t) d\phi = 1\) gilt. Nun dürfen wir aber nicht den Relaxationsprozess außer Acht lassen. Direkt nach dem \(\pi/2\)-rf-Puls beginnt sich die Magnetisierung zu entfokussieren, wodurch sich das Signal zusätzlich abschwächt. Diese Abschwächung verläuft exponentiell in Abhängigkeit der so genannten \(T_2\)-Zeit.

### 23.8. Formulas with Description of Variables

To describe variables within a formula, like in formula (29), a \(2 \times n\) matrix is used with left aligned columns for the \(n\) used variables.\textsuperscript{52} To set the description in a smaller size, before the matrix e.g. the command \texttt{\footnotesize} is inserted.\textsuperscript{53}

When the formula style \textbf{Indented}\textsuperscript{54} is used, a \texttt{\hfill} is inserted before and after the matrix to have the same separation of the matrix from the equation and the side margin.

When the formula style \textbf{Centered} is used, the method described in sec. 18.2.3 is used to indent formulas. Formula (29) consists of five columns whereas in the first

\begin{itemize}
  \item \texttt{\footnotesize} \textsuperscript{54} font sizes see sec. 11.4
  \item \texttt{\hfill} only works in formulas with the style \textbf{Indented}, see sec. 8.2.
\end{itemize}
two columns contain the equation, the third the matrix, and the last one an empty \TeX-brace.

\[ F_A = \rho \cdot V \cdot g \]

\( \rho \) density
\( V \) volume
\( g \) gravitational acceleration

(29)

23.9. Upright small Greek Letters

Most of the math fonts only provide italic small Greek letters. But for symbols of elementary particles like pions and neutrinos, upright Greek letters are needed. The file \texttt{upgreek.sty}\(^{56}\) that is loaded with the \LaTeX-preamble line

\texttt{\usepackage{upgreek}}

provides them. They are created when the command for a small Greek letters is started with \texttt{up}. For example the command \texttt{\uptau} creates this: \( \tau \)

With these commands reactions of elementary particles can be typeset:

\[ \pi^+ \rightarrow \mu^+ + \nu_\mu \]

The upright letters are more bold and wider than the italic ones. They should therefore not be used for units like “\( \mu m \)”.

23.10. Text Characters in Formulas

In some cases you might want to insert text characters directly into formulas. When for example the centered dot \( \cdot \) is often used in formulas like \( \nu = 5 \cdot 10^5 \) Hz, one would have to insert the command \texttt{\cdot} all the time, because this character is defined in all encodings as text character. But the encoding can be changed by this \LaTeX-preamble line:

\texttt{\Declare Inputtext{183}{\ifmmode\cdot\else{textperiodcentered}\fi}}

The character encoding (menu Document>Settings>Language) specifies what character appears when a keyboard key is pressed. When the key for the character ‘.’ is pressed, internally the command \texttt{textperiodcentered} is used. But this command is not available in a formula so that you would get \LaTeX-errors. With the changed encoding the right command is chosen automatically, depending on if the character was inserted into a formula or not.

The encoding of several characters is saved in definition files. For example the encoding \texttt{latin9} is defined in the file \texttt{latin9.def} that is in the installation folder of

\(^{56}\)upgreek is part of the \LaTeX-package \texttt{was}.

\(^{57}\)see sec. 10.3
\LaTeX. Encodings should only be changed via the \LaTeX-preamble and not in the definition files. Otherwise own documents could not be edited by other \LaTeX users working on other computers.

Besides the centered dot, in this document the degree sign ° is defined with the following \LaTeX-preamble line so that it can directly be inserted to a formula:

```
\DeclareInputtext{176}{\ifmmode^\circ\else\textdegree\fi}
```
A. Typographic Advice

This section is a summary of the most important typographic rules, listed in ISO norms.\footnote{This collection was partly taken from the German semi-official dictionary called “Duden” \cite{9} that lists some of the ISO rules.}

- Physical units are always set upright\footnote{done with font styles, see sec. 11.1}: \(30 \text{ km/h}\)
  Between the value and the unit is the smallest space, see sec. 8.1. This convention is automatically fulfilled when the command \texttt{\unittwo} is used. When it is entered to a formula, two boxes appear. In the first one the value is inserted, in the second one the unit, and one gets as above: \(30 \text{ km/h}\). Note that \texttt{\unittwo} is not a real \LaTeX{} command but the command \texttt{\unit[value]{unit}}, therefore you cannot use it in \LaTeX{} code.

- Percent and perthousand signs are set like physical units: \(1.2\%\text{ alcohol in blood}\)

- The degree sign follows directly on the value: \(15^\circ\), but not when it is used in units: \(15^\circ\text{C}\)

- In numbers with more than four digits the smallest space is inserted before every third digit to group them: \(18\,473\,588\)

- For dimensions like \(120\times90\times40\text{cm}\) the multiplication sign “\(\times\)” is used. It is available via the menu \textit{Insert} \textgreater{} \textit{Special Character} \textgreater{} \textit{Symbols}. With some keyboard definitions it can also be inserted directly.

- Functions with names consisting of several letters are set upright to avoid confusions, see sec. 15.1.

- Indices consisting of several letters, are set upright: \(E_{\text{kin}}\)
  Components of matrices are set italic: \(\hat{H}_{kl}\)

- The differentiation/integration operator ’d’, the Euler’s number ’e’, and the imaginary unit ’i’ should be set upright, to avoid mixing them up with other variables.
## B. Synonyms

Some characters and symbols can be created with several commands. Here is a list of the synonym commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>equivalent to</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ast</td>
<td>*</td>
</tr>
<tr>
<td>\choose</td>
<td>\binom</td>
</tr>
<tr>
<td>\geq</td>
<td>\ge</td>
</tr>
<tr>
<td>\lbrace</td>
<td>{</td>
</tr>
<tr>
<td>\lbrack</td>
<td>[</td>
</tr>
<tr>
<td>\leftarrow</td>
<td>\gets</td>
</tr>
<tr>
<td>\leq</td>
<td>\le</td>
</tr>
<tr>
<td>\lor</td>
<td>\vee</td>
</tr>
<tr>
<td>\neq</td>
<td>\not=</td>
</tr>
<tr>
<td>\slash</td>
<td>/</td>
</tr>
<tr>
<td>\vert</td>
<td>|</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>equivalent to</th>
</tr>
</thead>
<tbody>
<tr>
<td>\backslash</td>
<td>\</td>
</tr>
<tr>
<td>\dasharrow</td>
<td>\dashrightarrow</td>
</tr>
<tr>
<td>\land</td>
<td>\wedge</td>
</tr>
<tr>
<td>\rbrack</td>
<td>}</td>
</tr>
<tr>
<td>\rightarrow</td>
<td>\to</td>
</tr>
<tr>
<td>\rightarrow</td>
<td>\not=</td>
</tr>
<tr>
<td>\owns</td>
<td>\ni</td>
</tr>
<tr>
<td>\square</td>
<td>\Box</td>
</tr>
<tr>
<td>\Vert</td>
<td>|</td>
</tr>
</tbody>
</table>
References


[2] Description of \LaTeX’s math abilities

[3] \LaTeX tips and tricks-page

[4] Description of AM\TeX

[5] List of all symbols available with \LaTeX-packages

[6] Documentation of the \LaTeX-package hyperref

[7] Documentation of the \LaTeX-package mhchem

[8] Description of the command \texttt{\textbackslash mathclap}, described in sec.10.2

Index

€, 407
°, 412
Å, 411

Accents, 389
    for one character, 389
    for operators, 389
    for several characters, 390
    in text, 411
Arrows, 387
    diagonal, 388
    horizontal, 388
    labeled, 388
    vertical, 388

Binomial coefficients, 379

Boxes, 393
    as paragraph, 396
    colored, 395
    with frame, 393
    without frame, 394

Bracket size
    automatic, 385
    manual, 384
Brackets, 384
    for multiline expressions, 416
    horizontal, 386
    vertical, 384

Case differentiations, 379

Chemical characters
    Isotopes, 380
    Symbols, 425

Chemical equations, 425

Comma, 432

Commands
    @!
    @addtoreset, 423
    @!
    @removefromreset, 423
A
    \Alph, 424
    \addtocounter, 422
    \aligned, 420
    \alignedat, 420
    \alph, 424
    \arabic, 424, 425
    \arraycolsep, 383, 415
    \arraystretch, 384
B
    \big, 384
    \bigl - \bigr, 385
    \bigrm, 385
    \binom, 379
    \boldmath, 403
    \boldsymbol, 406
    \boxed, 393
    \brace, 379
    \brack, 379
C
    \cases, 379
    \CD, 427
    \cdots, 381
    \ce, 425
    \cf, 426
    \cfrac, 378
    \colorbox, 395
    \colorbox, 395
D
    \DeclareMathOperator, 401, 409
    \dbinom, 379
    \definecolor, 396
    \dfrac, 377
    \displaystyle, 374
    \dotfill, 382
    \dots, 381
E
    \euro, 407
F
    \fbox, 393
    \fcolorbox, 396
    \frac, 377
    \framebox, 393
G
placeholders, 380

relations, 408

roots, 378

space
besides inline formulas, 392
horizontal, 391
predefined, 391
variable, 392

special characters, 411
miscellaneous, 412

subscripts, see indices

sums, 398

superscripts, see exponents

symbols, 406
chemical, 425
Euro-symbol, 407
mathematical, 406
miscellaneous, 407

synonyms, 440

\TeX\-braces, 375
\TeX\-mode, 375


text
colored, 396
in formulas, 375, 420, 437


tilde, 411

tips, 432


typographic advice, 439

umlauts, 389
user-defined commands, 428
math macros, 429
Customizing LyX: Features for the Advanced User

by the LyX Team*

Version 1.6.x

August 17, 2009

*If you have comments or error corrections, please send them to the LyX Documentation mailing list, lyx-docs@lists.lyx.org. Include “[Customization]” in the subject header, and please cc the current maintainer of this file, Richard Heck <rgheck@comcast.net>.
Contents

1 Introduction 446

2 \texttt{LYX} configuration files 447
  2.1 What’s in \texttt{LYXDir}\? ................................. 447
    2.1.1 Automatically generated files ......................... 447
    2.1.2 Directories ........................................ 448
    2.1.3 Files you don’t want to modify ....................... 448
    2.1.4 Other files needing a line or two................... 449
  2.2 Your local configuration directory ...................... 449
  2.3 Running \texttt{LYX} with multiple configurations .......... 449

3 The Preferences dialog 451
  3.1 Formats ................................................. 451
  3.2 Copiers ................................................ 451
  3.3 Converters .............................................. 452

4 Internationalizing \texttt{LYX} 455
  4.1 Translating \texttt{LYX} .................................. 455
    4.1.1 Translating the graphical user interface (text messages) . . . 455
    4.1.1.1 Ambiguous messages ............................. 456
    4.1.2 Translating the documentation ....................... 456
  4.2 International Keymap Stuff .............................. 457
    4.2.1 The .kmap File ..................................... 458
    4.2.2 The .cdef File .................................... 459
    4.2.3 Dead Keys .......................................... 460
    4.2.4 Saving your Language Configuration ................. 460

5 Installing New Document Classes 461
  5.1 Installing a new \texttt{ETeX} package .................. 461
  5.2 Layouts ................................................ 462
    5.2.1 Layout modules .................................... 463
    5.2.2 Supporting new document classes .................... 464
    5.2.3 A layout for a sty file ............................ 464
    5.2.4 Layout for a cls file .............................. 465
  5.3 Declaring a new text class .............................. 465
    5.3.1 File format ....................................... 467
    5.3.2 General text class parameters ...................... 467
6 Including External Material

6.1 How does it work? ................................................. 481
6.2 The external template configuration file .................. 482
   6.2.1 The template header .................................... 483
   6.2.2 The Format section ..................................... 484
   6.2.3 Preamble definitions ................................... 485
6.3 The substitution mechanism ................................. 485
6.4 Security discussion ........................................... 487
1 Introduction

This manual covers the customization features present in LyX. In it, we discuss issues like keyboard shortcuts, screen previewing options, printer options, sending commands to LyX via the LyX Server, internationalization, installing new \LaTeX\ classes and LyX layouts, etc. We can’t possibly hope to touch on everything you can change—our developers add new features faster than we can document them—but we will explain the most common customizations and hopefully point you in the right direction for some of the more obscure ones.
2 LyX configuration files

This chapter aims to help you to find your way through the LyX configuration files. Before continuing to read this chapter, you should find out where your LyX library and user directories are by using Help > About LyX. The library directory is the place where LyX places its system-wide configuration files; the user directory is where you can place your modified versions. We will call the former \texttt{LyXDir} and the latter \texttt{UserDir} in the remainder of this document.

2.1 What's in LyXDir?

\texttt{LyXDir} and its sub-directories contain a number of files and that can be used to customize LyX's behavior. You can change many of these files from within LyX itself through the Tools > Preferences dialog. Most customization that you will want to do in LyX is possible through this dialog. However, many other inner aspects of LyX can be customized by modifying the files in \texttt{LyXDir}. These files fall in different categories, described in the following subsections.

2.1.1 Automatically generated files

The files, which are to be found in \texttt{UserDir}, are generated when you configure LyX. They contain various default values that are guessed by inspection. In general, it is not a good idea to modify them, since they might be overwritten at any time.

\texttt{lyxrc.defaults} contains defaults for various commands.

\texttt{packages.lst} contains the list of packages that have been recognized by LyX. It is currently unused by the LyX program itself, but the information extracted, and more, is made available with Help > \LaTeX Configuration.

\texttt{textclass.lst} the list of text classes that have been found in your layout/ directories, along with the associated \LaTeX document class and their description.

\texttt{lyxmodules.lst} the list of layout modules found in your layout/ directories

\texttt{*files.lst} lists of various sorts of \LaTeX-related files found on your system

\texttt{doc/\LaTeXConfig.lyx} is automatically generated during configuration from the file \texttt{\LaTeXConfig.lyx.in}. It contains information on your \LaTeX configuration.
### 2.1.2 Directories

These directories are duplicated between `LyXDir` and `UserDir`. If a particular files exists in both places, the one in `UserDir` will be used.

- **bind/** this directory contains files with the extension `.bind` that define the key-bindings used in LyX. If there exists an internationalized version of the bind file named `$LANG_xxx.bind`, that will be used first.

- **clipart/** contains graphics files that can be included in documents.

- **doc/** contains LyX documentation files (including the one you are currently reading). The file `MkXConfig.lyx` deserves special attention, as noted above. The internationalized help docs are in subdirectories `doc/xx` where “xx” is the ISO language code. See chapter 4 for details.

- **examples/** contains example files that explain how to use some features. In the file browser, press the `Examples` button to get there.

- **images/** contains image files that are used by the Document dialog. In addition, it also contains the individual icons used in the toolbar and the banners that can be shown when LyX is launched.

- **kbd/** contains keyboard keymapping files. See Chapter 4.2 for details.

- **layouts/** contains the text class and module files described in Chapter 5.

- **lyx2lyx** contains the `lyx2lyx` Python scripts used to convert between LyX versions. These can be run from the command line if, say, you want to batch-convert files.

- **scripts/** contains some files that demonstrate the capabilities of the External Template feature. Also contains some scripts used by LyX itself.

- **templates/** contains the standard LyX template files described in Chapter 5.4.

- **ui/** contains files with the extension `.ui` that define the user interface to LyX. That is, the files define which items appear in which menus and the items appearing on the toolbar.

### 2.1.3 Files you don’t want to modify

These files are used internally by LyX and you generally do not need to modify them unless you are a developer.

- **CREDITS** this file contains the list of LyX developers. The contents are displayed with the menu entry `Help ▶ About LyX`.
2.2 Your local configuration directory

chkconfig.ltx this is a \LaTeX{} script used during the configuration process. Do not run directly.

configure.py this is the script that is used to re-configure \LaTeX{}. It creates configuration files in the directory it was run from.

2.1.4 Other files needing a line or two...

encodings this contains tables describing how different character encodings can be mapped to Unicode

external_templates this file contains the templates available to the new External Template feature.

languages this file contains a list of all the languages currently supported by \LaTeX{}.

2.2 Your local configuration directory

Even if you are using \LaTeX{} as an unprivileged user, you might want to change \LaTeX{} configuration for your own use. The UserDir directory contains all your personal configuration files. This is the directory described as “user directory” in Help $\Rightarrow$ About \LaTeX{}. This directory is used as a mirror of LyXDir, which means that every file in UserDir is a replacement for the corresponding file in LyXDir. Any configuration file described in the above sections can be placed either in the system-wide directory, in which case it will affect all users, or in your local directory for your own use.

To make things clearer, let’s provide a few examples:

- The preferences set in the Tools $\Rightarrow$ Preferences dialog are saved to a file preferences in UserDir.

- When you reconfigure using Tools $\Rightarrow$ Reconfigure, \LaTeX{} runs the configure.py script, and the resulting files are written in your local configuration directory. This means that any additional text class file that you might have added in UserDir/layouts will be added to the list of classes in the Document $\Rightarrow$ Settings dialog.

- If you get some updated documentation from \LaTeX{} ftp site and cannot install it because you do not have sysadmin rights on your system, you can just copy the files in UserDir/doc/ and the items in the Help menu will open them!

2.3 Running LyX with multiple configurations

The configuration freedom of the local configuration directory may not suffice if you want to have more than one configuration at your disposal. For example, you may want to be use different key bindings or printer settings at different times. You can
achieve this by having several such directories. You then specify which directory to use at run-time.

Invoking \LaTeX{}X with the command line switch `-userdir <some directory>\' instructs the program to read the configuration from that directory, and not from the default directory. (You can determine the default directory by running \LaTeX{}X without the `-userdir\' switch.) If the specified directory does not exist, \LaTeX{}X offers to create it for you, just like it does for the default directory on the first time you run the program. You can modify the configuration options in this additional user directory exactly as you would for the default directory. These directories are completely independent (but read on). Note that setting the environment variable `LYX_USERDIR_VER` to some value has exactly the same effect.

Having several configurations also requires more maintenance: if you want to add a new layout to `NewUserDir/layouts` which you want available from all your configurations, you must add it to each directory separately. You can avoid this with the following trick: after \LaTeX{}X creates the additional directory, most of the subdirectories (see above) are empty. If you want the new configuration to mirror an existing one, replace the empty subdirectory with a symbolic link to the matching subdirectory in the existing configuration. Take care with the `doc/` subdirectory, however, since it contains a file written by the configuration script (also accessible through `Tools\rightarrow Reconfigure`) which is configuration-specific.
3 The Preferences dialog

All options of the preferences dialog are described in the Appendix *The Preferences Dialog* in the User’s Guide. For some options you might find here more details.

### 3.1 Formats

The first step is to define your file formats if they are not already defined. To do so, open the Tools $\rightarrow$ Preferences dialog. Under File Handling $\rightarrow$ File formats press the New... button to define your new format. The Format field contains the name used to identify the format in the GUI. The Short Name is used to identify the format internally. You will also need to enter a file extension. These are all required. The optional Shortcut field is used to provide a keyboard shortcut on the menus. (For example, pressing Alt-V D will View $\rightarrow$ DVI.)

A Format can have a Viewer and an Editor associated with it. For example, you might want to use Ghostview to view PostScript files. You can enter the command needed to start the program in the corresponding fields. In defining this command, you can use the four variables listed in the next section. The viewer is launched when you view an image in LyX or use the View menu. The editor is for example launched when you right-click on an image and choose Edit externally in the appearing context menu.

The Document format option tells LyX that a format is suitable for document export. If this is set and if a suitable conversion route exists (see sec. 3.3), the format will appear in the File $\rightarrow$ Export menu. The format will also appear in the View menu if a viewer is specified for the format. Pure image formats, such as png, should not use this option. Formats that can both represent vector graphics and documents like pdf should use it.

The option Vector graphics format tells LyX that a format can contain vector graphics. This information is used to determine the target format of included graphics for pdflatex export. Included graphics may need to be converted to either pdf, png, or jpg, since pdflatex cannot handle other image formats. If an included graphic is not already in pdf, png, or jpg format, it is converted to pdf if the vector format option is set, and otherwise to png.

### 3.2 Copiers

Since all conversions from one format to another take place in LyX’s temporary directory, it is sometimes necessary to modify a file before copying it to the temporary
directory in order that the conversion may be performed. This is done by a Copier: It copies a file to (or from) the temporary directory and may modify it in the process.

The definitions of the copiers may use four variables:

$$s$$ The LyX system directory (e.g. /usr/share/lyx).

$$i$$ The input file

$$o$$ The output file

$$l$$ The ‘\LaTeX name’

The latter should be the filename as it would be used in a \LaTeX’s \texttt{include} command. It is relevant only when exporting files suitable for such inclusion.

Copiers can be used to do almost anything with output files. For example, suppose you want generated pdf files to be copied to a special directory, /home/you/pdf/. Then you could write a shell script such as this one:

```bash
#!/bin/bash
FROMFILE=$1
TOFILE='basename $2 '
cp $FROMFILE /home/you/pdf/$TOFILE
```

Save it in your local LyX directory—say, /home/you/.lyx/scripts/pdftopper.sh—and make it executable, if you need to do so on your platform. Then, in the Tools▷ Preferences dialog, select under File Handling▷ File formats the PDF(pdflatex) format—or one of the other pdf formats—and enter pdftopper.sh $$i$$ $$o$$ into the Copier field.

Copiers are used by LyX in various of its own conversions. For example, if appropriate programs are found, LyX will automatically install copiers for the HTML and HTML (MS Word) formats. When these formats are exported, the copier sees that not just the main HTML file but various associated files (style files, images, etc.) are also copied. All these files are written to a subdirectory of the directory in which the original LyX file was found.

### 3.3 Converters

You can define your own Converters to convert files between different formats. This is done in the Tools▷ Preferences▷ File Handling▷ Converters dialog.

---

1. For example, the file may refer to other files—images, for example—using relative file names, and these may become invalid when the file is copied to the temporary directory.

2. This copier can be customized. The optional “-e” argument takes a comma-separated list of extensions to be copied; if it is omitted, all files will be copied. The “-t” argument determines the extension added to the generated directory. By default, it is “LyXconv”, so HTML generated from /path/to/filename.lyx will end up in /path/to/filename.html.LyXconv.
3.3 Converters

To define a new converter, select the **From format** and **To format** from the dropdown lists, enter the command needed for the conversion, and then press the **Add** button. Several variables can be used in the definition of converters:

- **$$s$$**: The \LaTeX{} system directory
- **$$i$$**: The input file
- **$$o$$**: The output file
- **$$b$$**: The base filename of the input file (i.e., without the extension)
- **$$p$$**: The path to the input file
- **$$r$$**: The path to the original input file (this is different from $$p$$ when a chain of converters is called).

In the **Extra Flag** field you can enter the following flags, separated by commas:

- **latex**: This converter runs some form of \LaTeX. This will make \LaTeX{}'s \LaTeX{} error logs available.
- **needaux**: Needs the \LaTeX{} .aux file for the conversion.
- **xml**: Output is XML.

The following three flags are not really flags at all because they take an argument in the **key = value** format:

- **parselog**: If set, the converter’s standard error will be redirected to a file `infolog`, and the script given as argument will be run as: `script < infilename`.
- **resultdir**: The name of the directory in which the converter will dump the generated files. \LaTeX{} will not create this directory, and it does not copy anything into it, though it will copy this directory to the destination. The argument may contain $$b$$, which will be replaced by the base name of the input and output files, respectively, when the directory is copied.
- **resultfile**: Determines the output file name and may contain $$b$$. Sensible only with resultdir and optional even then; if not given, it defaults to ‘index’.

None of these last three are presently used in any of the converters that are installed with \LaTeX{}.

You do not have to define converters for all formats between which you want to convert. For example, you will note that there is no ‘\LaTeX{} to PostScript’ converter, but \LaTeX{} will export PostScript. It does so by first creating a \LaTeX{} file (no converter
needs to be defined for this) which is then converted to DVI using the ‘\LaTeX to DVI’ converter, and finally converting the resulting DVI file to PostScript. \LaTeX{} finds such ‘chains’ of converters automatically, and it will always choose the shortest possible chain. You can, though, still define multiple conversion methods between file formats. For example, the standard \LaTeX{} configuration provides three ways to convert \LaTeX{} to PDF: Directly, using \texttt{pdflatex}; via (DVI and) PostScript, using \texttt{ps2pdf}; or via DVI, using \texttt{dvipdfm}. To define such alternate chains, you must define multiple target ‘file formats’, as described in section 3.1. For example, in the standard configuration, the formats named \texttt{pdf}, \texttt{pdf2}, and \texttt{pdf3} are defined, all of which share the extension \texttt{.pdf}, and which correspond to the conversion methods just mentioned.
4 Internationalizing LyX

LyX supports using a translated interface. Last time we checked, LyX provided text in thirty languages. The language of choice is called your *locale*. (For further reading on locale settings, see also the documentation for locale that comes with your operating system. For Linux, the manual page for `locale(5)` could be a good place to start).

Notice that these translations will work, but do contain a few flaws. In particular, all dialogs have been designed with the English text in mind, which means that some of the translated text will be too large to fit within the space allocated. This is only a display problem and will not cause any harm. Also, you will find that some of the translations do not define shortcut keys for everything. Sometimes, there are simply not enough free letters to do it. Other times, the translator just hasn’t got around to doing it yet. Our localization team, which you may wish to join, will of course try to fix these shortcomings in future versions of LyX.

4.1 Translating LyX

4.1.1 Translating the graphical user interface (text messages).

LyX uses the GNU `gettext` library to handle the internationalization of the interface. To have LyX speak your favorite language in all menus and dialogs, you need a `po`-file for that language. When this is available, you’ll have to generate a `mo`-file from it and install the `mo`-file. The process of doing all of this is explained in the documentation for GNU `gettext`. It is possible to do this just for yourself, but if you’re going to do it, you might as well share the results of your labors with the rest of the LyX community. Send a message to the LyX developers’ list for more information about how to proceed.

In short, this is what you should do (xx denotes the language code):

- Check out the LyX source code. (See the information on the web.)

- Copy the file `lyx.pot` to the folder of the `*.po` files. Then rename it to `xx.po`. (If `lyx.pot` doesn’t exist anywhere, it can be remade with the console command `make lyx.pot` in that directory, or you can use an existing `po`-file for some other language as a template).

---

1If you are a fluent speaker of a language other than English, joining these teams is a great way to give back to the LyX community!
• Edit \texttt{xx.po}.\footnote{This is just a text file, so it can be edited in any text editor. But there are also specialized programs that support such editing, such as \texttt{Poedit} (for all platforms) or \texttt{kBabel} (for KDE). \texttt{Emacs} contains a ‘mode’ for editing \texttt{po} files, as well.} For some menu- and widget-labels, there are also shortcut keys that should be translated. Those keys are marked after a ‘|’, and should be translated according to the words and phrases of the language. You should also fill also out the information at the beginning of the new \texttt{po}-file with your email-address, etc., so people know where to reach you with suggestions and entertaining flames.

If you are just doing this on your own, then:

• Generate \texttt{xx.mo}. This can be done with \texttt{msgfmt -o xx.mo < xx.po}.

• Copy the \texttt{mo}-file to your locale-tree, at the correct directory for application messages for the language \texttt{xx}, and under the name \texttt{lyx.mo} (e.g. /usr/local/share/locale/xx/LC_MESSAGES/lyx.mo).

As said, however, it would be best if the new \texttt{po}-file could be added to the \texttt{LyX} distribution, so others can use it. Adding it involves making additional changes to \texttt{LyX}. So send an email to the developers’ mailing list if you’re interested in doing that.

### 4.1.1.1 Ambiguous messages

Sometimes it turns out that one English message needs to be translated into different messages in the target language. One example is the message \texttt{To} which has the German translation Nach or Bis, depending upon exactly what the English “to” means. \texttt{GNU gettext} does not handle such ambiguous translations. Therefore you have to add some context information to the message: Instead of \texttt{To} it becomes \texttt{To[[as in ‘From format x to format y’]]} and \texttt{To[[as in ‘From page x to page y’]]}.

Now the two occurrences of \texttt{To} are different for \texttt{gettext} and can be translated correctly to Nach and Bis, respectively.

Of course the context information needs to be stripped off the original message when no translation is used. Therefore you have to put it in double square brackets at the end of the message (see the example above). The translation mechanism of \texttt{LyX} ensures that everything in double square brackets at the end of messages is removed before displaying the message.

### 4.1.2 Translating the documentation.

The online documentation (in the \texttt{Help}-menu) can (and should!) be translated. If there are translated versions of the documentation available\footnote{As of March 2008, at least some of the documents have been translated into fourteen languages, with the Tutorial available in a few more.} and the locale is set accordingly, these will be used automagically by \texttt{LyX}. \texttt{LyX} looks for translated versions as \texttt{LyXDir/doc/xx/DocName.lyx}, where \texttt{xx} is the code for the language currently in use.
use. If there are no translated documents, the default English versions will be displayed. Note that the translated versions must have the same filenames (DocName above) as the original. If you feel up to translating the documentation (an excellent way to proof-read the original documentation by the way!), there are a few things you should do right away:

- Check out the documentation translation web page at http://www.lyx.org/Translation. That way, you can find out which (if any) documents have already been translated into your language. You can also find out who (if anyone) is organizing the effort to translate the documentation into your language. If no one is organizing the effort, please let us know that you’re interested.

Once you get to actually translating, here’s a few hints for you that may save you trouble:

- Join the documentation team! There is information on how to do that in Intro.lyx (Help▷Introduction), which by the way is the first document you should translate.

- Learn the typographic conventions for the language you are translating to. Typography is an ancient art and over the centuries, a great variety of conventions have developed throughout different parts of the world. Also study the professional terminology amongst typographers in your country. Inventing your own terminology will only confuse the users. *(Warning! Typography is addictive!)*

- Make a copy of the document. This will be your working copy. You can use this as your personal translated help-file by placing it in your UserDir/doc/xx/directory.

- Sometimes the original document (from the LyX-team) will be updated. Use the source viewer at http://www.lyx.org/trac/timeline to see what has been changed. That way you can easily see which parts of the translated document need to be updated.

If you ever find an error in the original document, fix it and notify the rest of the documentation team of the changes! *(You didn’t forget to join the documentation team, did you?)*

### 4.2 International Keymap Stuff

The next two sections describe the .kmap and .cdef file syntax in detail. These sections should help you design your own key map if the ones provided do not meet your needs.
4 Internationalizing LyX

4.2.1 The .kmap File

A .kmap file maps keystrokes to characters or strings. As the name suggests, it sets a keyboard mapping. The .kmap file keywords \texttt{kmap}, \texttt{kmod}, \texttt{ksmod}, and \texttt{kcomb} are described in this section.

\texttt{kmap} \quad Map a character to a string

\begin{verbatim}
\textbackslash kmap \ char \ string
\end{verbatim}

This will map \texttt{char} to \texttt{string}. Note that in \texttt{string}, the double-quote (" ) and the backslash (\ ) must be escaped with a preceding backslash (\ ).

An example of a \texttt{kmap} statement to cause the symbol / to be output for the keystroke \& is:

\begin{verbatim}
\textbackslash kmap \ & \ / \\
\end{verbatim}

\texttt{kmod} \quad Specify an accent character

\texttt{\textbackslash kmod \ char \ accent \ allowed}

This will make the character \texttt{char} be an accent on the \texttt{allowed} character(s). This is the dead key\footnote{The term dead key refers to a key that does not produce a character by itself, but when followed with another key, produces the desired accent character. For example, a German character with an umlaut like ä can be produced in this manner.} mechanism.

If you hit \texttt{char} and then another key not in \texttt{allowed}, you will get a \texttt{char} followed by the other, not allowed key, as output. Note that a \texttt{Backspace} cancels a dead key, so if you hit \texttt{char Backspace}, the cursor will not go one position backwards but will instead cancel the effect that \texttt{char} might have had on the next keystroke.

The following example specifies that the character ‘ is to be an acute accent, allowed on the characters a, e, i, o, u, A, E, I, O, and U:

\begin{verbatim}
\textbackslash kmod \ ' \ acute \ aeiouAEIOU
\end{verbatim}

\texttt{ksmod} \quad Specify an exception to the accent character

\texttt{\textbackslash ksmod \ accent \ char \ result}

This defines an exception for accent on \texttt{char}. The \texttt{accent} must have been assigned a keystroke with a previous \texttt{\textbackslash kmod} declaration and \texttt{char} must not belong in the \texttt{allowed} set of \texttt{accent}. When you enter the \texttt{accent char} sequence, \texttt{result} is produced. If such a declaration does not exist in the .kmap file and you enter \texttt{accent char}, you get \texttt{accent_key char} where \texttt{accent_key} is the first argument of the \texttt{\textbackslash kmod} declaration.

The following command produces causes åi to be produced when you enter acute-i (‘i):
4.2 International Keymap Stuff

\kxmod acute i "\\'{i}"

kcomb Combine two accent characters

\kcomb accent1 accent2 allowed

This one is getting pretty esoteric. It allows you to combine the effect of \textit{accent1} and \textit{accent2} (in that order!) on \textit{allowed} chars. The keystrokes for \textit{accent1} and \textit{accent2} must have been set with a \kmod command at a previous point in the file.

Consider this example from the \texttt{greek.kmap} file:

\kmod ; acute aeioyvhAEIOYVH \kmod : umlaut iyIY \kcomb acute umlaut iyIY

This allows you to press ;:i and get the effect of "\'{i}". A backspace in this case cancels the last dead key, so if you press ;: Backspace i you get "\'{i}".

4.2.2 The .cdef File

After the .kmap mapping is performed, a .cdef file maps the strings that the symbols generate to characters in the current font. The \textsc{LyX} distribution currently includes at least the iso8859-1.cdef and iso8859-2.cdef files.

In general the .cdef file is a sequence of declarations of the form

\begin{verbatim}
char_index_in_set string
\end{verbatim}

For example, in order to map \"{e} to the corresponding character in the iso-8859-1 set (233), the following declaration is used

\begin{verbatim}
233 "\\'{e}"
\end{verbatim}

with \ and " being escaped in \textit{string}. Note that the same character can apply to more than one string. In the iso-8859-7.cdef file you have

\begin{verbatim}
192 "\\'{i}\"{i}"
192 "\\"{i}\"{i}"\end{verbatim}

If \textsc{LyX} cannot find a mapping for the string produced by the keystroke or a deadkey sequence, it will check if it looks like an accented char and try to draw an accent over the character on screen.
4 Internationalizing LyX

4.2.3 Dead Keys

There is a second way to add support for international characters through so-called dead-keys. A dead-key works in combination with a letter to produce an accented character. Here, we’ll explain how to create a really simple dead-key to illustrate how they work.

Suppose you happen to need the circumflex character, “ˆ”. You could bind the ^-key [a.k.a. Shift-6] to the LyX command accent-circumflex in your lyxrc file. Now, whenever you type the ^-key followed by a letter, that letter will have a circumflex accent on it. For example, the sequence “^e” produces the letter: “ê”. If you tried to type “^t”, however, LyX will complain with a beep, since a “t” never takes a circumflex accent. Hitting Space after a dead-key produces the bare-accent. Please note this last point! If you bind a key to a dead-key, you’ll need to rebind the character on that key to yet another key. Binding the , key to a cedilla is a bad idea, since you’ll only get cedillas instead of commas.

One common way to bind dead-keys is to use Meta-, Ctrl-, and Shift- in combination with an accent, like “~” or “’” or “’’”. Another way involves using xmodmap and xkeycaps to set up the special Mode_Switch key. The Mode_Switch acts in some ways just like Shift and permits you to bind keys to accented characters. You can also turn keys into dead-keys by binding them to something like usldead_cedilla and then binding this symbolic key to the corresponding LyX command. You can make just about anything into the Mode_Switch key: One of the Ctrl- keys, a spare function key, etc. As for the LyX commands that produce accents, check the entry for accent-acute in the Reference Manual. You’ll find the complete list there.

4.2.4 Saving your Language Configuration

You can edit your preferences so that your desired language environment is automatically configured when LyX starts up, via the Edit Preferences dialog.

---

5Note from John Weiss: This is exactly what I do in my ~/.lyx/lyxrc and my ~/.xmodmap files. I have my Scroll Lock key set up as Mode_Shift and a bunch of these “usldead_*” symbolic keys bound such things as Scroll Lock¬ and Scroll Lock=. This is how I produce my accented characters.
5 Installing New Document Classes, Layouts, and Templates

In this chapter, we describe the procedures for creating and installing new LYX layout and template files, as well as offer a refresher on correctly installing new \LaTeX\ document classes. Some definitions: a document class is a \LaTeX\ file (usually ending in .cls or .sty) that describes the format of a document such as an article, report, journal preprint, etc, and all the commands needed to realize that format. A layout file is a LYX file that corresponds to a \LaTeX\ document class and that tells LYX how to “draw” things on the screen to make the display look something like the final printed page. More precisely, a layout file describes a “text class” which is the internal construct LYX uses to render the screen display. “Layout” and “text class” can be used somewhat interchangeably, but it is better to refer to the file as the layout, and the thing living in LYX’s memory as the text class. A template file is simply a LYX document that contains a set of predefined entries for a given document class—entries that are generally required for that class. Templates are especially useful for things like journal manuscripts that are to be submitted electronically.

5.1 Installing a new \LaTeX\ package

Some installations may not include a \LaTeX\ package that you would like to use within LYX. For example, you might need Foil\TeX, a package for preparing slides or viewgraphs for overhead projectors. Modern \LaTeX\ distributions like \TeX\Live (2008 or newer) or MiK\TeX\ provide a user interface for that. For example on MiK\TeX\ you start its program “Browse Packages” to get a list of available packages. To install one, right click on it or use the installing toolbar button. When the package you want to install is not in the list, but you have it in form of a .cls or .sty-file, then copy these files to a subfolder of your \LaTeX\ distribution, for example to the folder ~\tex\latex. Then update the file name database of your \LaTeX\-distribution. For example on MiK\TeX\ this is done by pressing the button Refresh FNDB that you find in MiK\TeX\’s “Settings” program. In both cases you need afterwards to reconfigure LYX using the menu Tools\rightarrow Reconfigure and then to restart LYX.

If your \LaTeX\ distribution doesn’t provide a user interface, then you can follow these steps by using a UNIX/Linux console.

1. Get the package from CTAN or wherever.

2. You can install this package in several different places. If you want it to be
available for all users on your system, then you should install it in your ‘local’ \TeX\ tree; if you want (or need) it to be available just for you, then you can install it in your own ‘user’ \TeX\ tree. Where these should be created, if they do not already exist, depends upon the details of your system. To find out, look in the file \texttt{texmf.cnf}.ootnote{This usually lives in the directory \texttt{$TEXMF/web2c}, though you can run \texttt{kpsewhich texmf.cnf} to locate it.} The location of the ‘local’ \TeX\ tree is defined by \texttt{TEXMFLOCAL}; this is usually somewhere like \texttt{/usr/local/share/texmf/}. The ‘user’ \TeX\ tree is defined by \texttt{TEXMFHOME} and is commonly at \texttt{$HOME/texmf/}. (If these variables are not predefined, you can define them.) You’ll probably need root permissions to create or modify the ‘local’ tree; but your ‘user’ tree shouldn’t have such limitations.

3. Make sure \texttt{TEXMF} includes the \texttt{TEXMFLOCAL} and \texttt{TEXMFHOME} variables; e.g.

\begin{verbatim}
TEXMF = {$TEXMFHOME,!!$TEXMFLOCAL,!!$TEXMFMAIN}
\end{verbatim}

But, again, most of this will “just work”.

4. Create your local\footnote{We’ll assume henceforth that you’re defining ‘local’ \TeX\ tree. If you’re defining a user tree, just adjust as necessary.} \TeX\ tree. You must follow the directory structure of your existing \texttt{texmf} directory, which will be found at \texttt{TEXMFMAIN}. For example, latex packages should go under \texttt{$TEXMFLOCAL/tex/latex/}.

5. Install the package. For example, you would unpack the Foil\TeX\ tarball and copy it to \texttt{$TEXMFLOCAL/tex/latex/foiltex}. The \texttt{foiltex} directory contains various files.

6. Run: \texttt{texhash}. This should create \texttt{$TEXMFLOCAL/ls-R} amongst others.

Your package is now installed and available to \LaTeX. To make it available to \LX, you need to create a Layout file, if one is not already available. (See the next section.) Once you have a layout file, you need only reconfigure (Tools > Reconfigure) and then restart \LX. You should then see your new package—for example \texttt{slides} (Foil\TeX)—under Document > Settings in the Document Class drop box.

### 5.2 Layouts

This section describes how to write and install your own \LX layout files and walks through the \texttt{article} text class format as an example. The \texttt{.layout} files describe what paragraph and character styles are available for a given document class and how \LX should display them. We try to provide a thorough description of the process here; however, there are so many different types of documents supported by \LaTeX\ classes that we can’t hope to cover every different possibility or problem you might encounter. (The \LX users’ list is frequented by people with lots of experience with layout design who are willing to share what they’ve learned.)
As you prepare to write a new layout, it is extremely helpful to look at the example layouts distributed with LyX. If you use a nice \LaTeX document class that might be of interest for others, too, and have a nice corresponding LyX layout, feel free to contribute the stuff to us, so we may put it into the distribution. There is also a section on the LyX wiki for this kind of material.

All the tags described in this chapter are case-insensitive; this means that Style, style and STYlE are really the same command. The possible values are printed in brackets after the feature’s name. The default value if a feature isn’t specified inside a text class-description is typeset \textit{emphasized}. If the argument has a data type like “string” or “float”, the default is shown like this: \texttt{float=default}.

### 5.2.1 Layout modules

Similar to layout files, and new with LyX 1.6, are layout \textit{modules}. Modules are to \LaTeX packages much as layouts are to \LaTeX classes, and some modules—such as the endnotes module—provide support for just such a package. In a sense, layout modules are similar to included files—files like \texttt{stdsections.inc}—in that modules are not specific to a given document layout but may be used with many different layouts. The difference is that using a layout module does not require editing the layout file. Rather, modules are selected in the Document\texttt{\&} Settings dialog.

Building modules is the easiest way to get started with layout editing, since it can be as simple as adding a single new paragraph or flex inset. But modules may, in principle, contain anything a layout file can contain.

A module must begin with a line like the following:

```
:\DeclareLYXModule[endnotes.sty]{Endnotes}
```

The argument in square brackets is optional: It declares any \LaTeX packages on which the module depends. The mandatory argument, in curly brackets, is the name of the module, as it should appear in Document\texttt{\&} Settings.

The module declaration should then be followed by lines like the following:

```
#\DescriptionBegin
# Adds an endnote command, in addition to footnotes.
# You will need to add \texttt{theendnotes} in \TeX code where you
# want the endnotes to appear.
#\DescriptionEnd
# Requires: somemodule | othermodule
# Excludes: badmodule
```

The description is used in Document\texttt{\&} Settings to provide the user with information about what the module does. The Requires line is used to identify other modules with which this one must be used; the Excludes line is used to identify modules with which this one may not be used. Both are optional, and, as shown, multiple modules should be separated with the pipe symbol: \texttt{|}. Note that the required modules are
treated disjunctively: \textit{at least one} of the required modules must be used. Similarly, \textit{no} excluded module may be used. Note that modules are identified here by their \textit{file-names} without the .module extension. So \texttt{somemodule} is really \texttt{somemodule.module}.

After creating a new module, you will need to reconfigure and then restart LyX for the module to appear in the menu. However, changes you make to the module will be seen immediately, if you open Document $\rightarrow$ Settings, highlight something, and then hit “OK”. It is strongly recommended that you save your work before doing so. In fact, \textit{it is strongly recommended that you not attempt to edit modules while simultaneously working on documents}. Though of course the developers strive to keep LyX stable in such situations, syntax errors and the like in your module file could cause strange behavior.

### 5.2.2 Supporting new document classes

There are two situations you are likely to encounter when wanting to support a new \LaTeX document class, involving \LaTeX2ε class (.cls) and style (.sty) files. Supporting a style file is usually fairly easy. Supporting a new document class is a bit harder.

### 5.2.3 A layout for a sty file

If your new document class is provided as a style file that is used in conjunction with an existing, supported document class—for the sake of the example, we’ll assume that the style file is called \texttt{myclass.sty} and it is meant to be used with \texttt{report.cls}, which is a standard class—start by copying the existing class’s layout file into your local directory:

```bash
cp report.layout ~/.lyx/layouts/myclass.layout
```

Then edit \texttt{myclass.layout} and change the line:

```latex
\DeclareLaTeXClass{report}
```

to read

```latex
\DeclareLaTeXClass[report, myclass.sty]{report (myclass)}
```

Then add:

```
Preamble
  \usepackage{myclass}
EndPreamble
```
5.3 Declaring a new text class

near the top of the file.

Start \LaTeXX and select Tools >> Reconfigure. Then restart \LaTeXX and try creating a new document. You should see 'report (myclass)' as a document class option in the Document >> Settings dialog. It is likely that some of the sectioning commands and such in your new class will work differently from how they worked in the base class—report in this example—so you can fiddle around with the settings for the different sections if you wish.

5.2.4 Layout for a cls file

There are two possibilities here. One is that the class file is itself based upon an existing document class. For example, many thesis classes are based upon book.cls. To see whether yours is, look for a line like

\LoadClass{book}

in the file. If so, then you may proceed largely as in the previous section, though the \Declare\LaTeXXClass line will be different. If your new class is thesis, and it is based upon book, then the line should read:\footnote{And it will be easiest if you save the file to thesis.layout: \LaTeXX assumes that the document class has the same name as the layout file.}

\Declare\LaTeXXClass[thesis,book]{thesis}

If, on the other hand, the new class is not based upon an existing class, you will probably have to “roll your own” layout. We strongly suggest copying an existing layout file which uses a similar \LaTeXX class and then modifying it, if you can do so. At least use an existing file as a starting point so you can find out what items you need to worry about. Again, the specifics are covered below.

5.3 Declaring a new text class

When it’s finally time to get your hands dirty and create or edit your own layout file, the following sections describe what you’re up against. Our advice is to go slowly, save and test often, listen to soothing music, and enjoy one or two of your favorite adult beverages; more if you are getting particularly stuck. It’s really not that hard, except that the multitude of options can become overwhelming if you try to do to much in one sitting. Go have another adult beverage, just for good measure.

Here we go!

Lines in a layout file which begin with a # are comments. There is one exception to this rule: all layouts should begin with lines like:

\% Do not delete the line below; configure depends on this
# \Declare\LaTeXXClass{article}
The second line is used when you configure $\LaTeX$. The layout file is read by the $\LaTeX$ script `chkconfig.ltx`, in a special mode where `#` is ignored. The first line is just a $\LaTeX$ comment, and the second one contains the declaration of the text class. If these lines appear in a file named `article.layout`, then they define a text class of name `article` (the name of the layout file) which uses the $\LaTeX$ document class `article_cls` (the default is to use the same name as the layout). The string “article” that appears above is used as a description of the text class in the Document ➤ Settings dialog.

Let’s assume that you wrote your own text class that uses the `article_cls` document class, but where you changed the appearance of the section headings. If you put it in a file `myarticle.layout`, the header of this file should be:

```
#% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[article]{article (with my own headings)}
```

This declares a text class `myarticle`, associated with the \LaTeX\ document class `article_cls` and described as “article (with my own headings)”. If your text class depends on several packages, you can declare it as:

```
#% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[article,foo.sty]{article (with my own headings)}
```

This indicates that your text class uses the foo.sty package. Finally, it is also possible to declare classes for DocBook code. Typical declarations will look like

```
#% Do not delete the line below; configure depends on this
# \DeclareDocBookClass[article]{SGML (DocBook article)}
```

Note that these declarations can also be given an optional parameter declaring the name of the document class (but not a list).

So, to be as explicit as possible, the form of the layout declaration is:

```
# \DeclareLaTeXClass[class,package.sty]{layout description}
```

The class need only be specified if the name of the $\LaTeX$ class file and the name of the layout file are different; if the name of the class file is not specified, then $\LaTeX$ will simply assume that it is the same as the name of the layout file.

When the text class has been modified to your taste, all you have to do is to copy it either to `LYXDir/layouts/` or to `UserDir/layouts`, run `Tools ➤ Reconfigure`, exit $\LaTeX$ and restart it. Then your new text class should be available along with the others.

In versions of $\LaTeX$ prior to 1.6, you had to restart $\LaTeX$ to see any changes you made to your layout files. As a result, editing layout files could be very time consuming. Beginning with 1.6, however, you can force a reload of the layout currently in use by using the $\LaTeX$ function `layout-reload`. There is no default binding for this function—though, of course, you can bind it to a key yourself. If you want to use this function, then, you should simply enter it in the mini-buffer. **Warning:** This
5.3 Declaring a new text class

is very much an ‘advanced feature’. It is strongly recommended that you save your work before using this function. In fact, it is strongly recommended that you not attempt to edit your layout while simultaneously working on a document that you care about. Use a test document. Syntax errors and the like in your layout file could cause peculiar behavior. In particular, such errors could cause LyX to regard the current layout as invalid and to attempt to switch to some other layout. The LyX team strives to keep LyX stable in such situations, but safe is better than sorry.

5.3.1 File format

The first non-comment line must contain the file format number:

Format [int] This tag was introduced with LyX 1.4.0 (layout files of LyX 1.3.x and earlier don’t have an explicit file format). The file format that is documented here is format 11.

5.3.2 General text class parameters

These are the general parameters which describe the form of the entire document:

AddToPreamble Adds information to the document preamble. Must end with “EndPreamble”.

ClassOptions Describes various global options supported by the document class. See Section 5.3.3 for a description. Must end with “End”.

Columns [1, 2] Whether the class should default to having one or two columns. Can be changed in the Document > Settings dialog.

Counter This sequence defines a new counter. See Section 5.3.7 for details. Must end with “End”.

DefaultFont Sets the default font used to display the document. See Section 5.3.8 for how to declare fonts. Must end with “EndFont”.

DefaultModule [string] Specifies a module to be included by default with this document class, which should be specified by filename without the .module extension. The user can still remove the module, but it will be active at the outset. (This applies only when new files are created, or when this class is chosen for an existing document.)

DefaultStyle [string] This is the style that will be assigned to new paragraphs, usually Standard. This will default to the first defined style if not given, but you are highly encouraged to use this directive.

ExcludesModule [string] Indicates that the module in question—which should be specified by filename without the .module extension—cannot be used with this document class. This might be used in a journal-specific layout file to prevent,
5 Installing New Document Classes

say, the use of the theorems-sec module that numbers theorems by section. This tag may not be used in a module. Modules have their own way of excluding other modules (see 5.2.1).

**Float** Defines a new float. See Section 5.3.5 for details. Must end with “End”.

**Input** As its name implies, this command allows you to include another layout definition file within yours to avoid duplicating commands. Common examples are the standard layout files, for example, stdclass.inc, which contains most of the basic layouts.

**InsetLayout** This section (re-)defines the layout of an inset. It can be applied to an existing inset of to a new, user-defined inset, e.g. a new character style. See Section 5.3.6 for more information. Must end with “End”.

**LeftMargin** A string that indicates the width of the left margin on the screen, for example, “MMMMM”.

**NoFloat** This command deletes an existing float. This is particularly useful when you want to suppress a float that has been defined in an input file.

**NoStyle** This command deletes an existing style. This is particularly useful when you want to suppress a style that has been defined in an input file.

**OutputType** A string indicating what sort of output documents using this class will produce. At present, the options are: ‘docbook’, ‘latex’, and ‘literate’.

**PageStyle** [plain, empty, headings] The class default pagestyle. Can be changed in the Document▷Settings dialog.

**Preamble** Sets the preamble for the \LaTeX document. Note that this will completely override any prior Preamble or AddToPreamble declarations. Must end with “EndPreamble”.

**Provides** [string] [0, 1] Whether the class already provides the feature string. A feature is in general the name of a package (amsmath, makeidx, ...) or a macro (url, boldsymbol, ...); the complete list of supported features is unfortunately not documented outside the LyX source code—but see \LaTeXFeatures.cpp if you’re interested. Help▷\LaTeX Configuration also gives an overview of the supported packages.

**ProvidesModule** [string] Indicates that this layout provides the functionality of the module mentioned, which should be specified by the filename without the .module extension. This will typically be used if the layout includes the module directly, rather than using the DefaultModule tag to indicate that it ought to be used. It could be used in a module that provided an alternate implementation of the same functionality.
5.3 Declaring a new text class

**Requires** [string] Whether the class requires the feature string. Multiple features must be separated by commas. Note that you can only request supported features.

**RightMargin** A string that indicates the width of the right margin on the screen, for example, “MMMMMM”.

**SecNumDepth** Sets which divisions get numbered. Corresponds to the secnumdepth counter in \LaTeX{}.

**Sides** [1, 2] Whether the class-default should be printing on one or both sides of the paper. Can be changed in the Document-&gt;Settings dialog.

**Style** This sequence defines a new paragraph style. If the style already exists, it will redefine some of its parameters instead. See Section 5.3.4 for details. Must end with “End”.

**TitleLatexName** [string="maketitle"] The name of the command or environment to be used with TitleLatexType.

**TitleLatexType** [CommandAfter, Environment] Indicates what kind of markup is used to define the title of a document. CommandAfter means that the macro with name TitleLatexName will be inserted after the last layout which has “InTitle 1”. Environment corresponds to the case where the block of paragraphs which have “InTitle 1” should be enclosed into the TitleLatexName environment.

**TocDepth** Sets which divisions are included in the table of contents. Corresponds to the tocdepth counter in \LaTeX{}.

### 5.3.3 ClassOptions section

The ClassOptions section can contain the following entries:

**FontSize** [string="10|11|12"] The list of available font sizes for the document’s main font, separated by “|”.

**Header** Used to set the DTD line with XML-based output classes. E.g.: PUBLIC “-//OASIS//DTD DocBook V4.2//EN”.

**PageStyle** [string="empty|plain|headings|fancy"] The list of available page styles, separated by “|”.

**Other** [string=""] Some document class options, separated by a comma, that will be added to the optional part of the \documentclass command.

The ClassOptions section must end with “End”.

469
5 Installing New Document Classes

5.3.4 Paragraph Styles

A paragraph style description looks like this:\(^4\)

```latex
Style name
...
End
```

where the following commands are allowed:

- **Align** \[block, left, right, center\] Paragraph alignment.

- **AlignPossible** \[block, left, right, center\] A comma separated list of permitted alignments. (Some \LaTeX\X styles prohibit certain alignments, since those wouldn’t make sense. For example a right-aligned or centered enumeration isn’t possible.)

- **BottomSep** \[float=0\]^5\ The vertical space with which the last of a chain of paragraphs with this layout is separated from the following paragraph. If the next paragraph has another layout, the separations are not simply added, but the maximum is taken.

- **Category** \[string\] The category for this style. This is used to group related styles in the Layout combobox on the toolbar. Any string can be used, but you may want to use existing categories with your own styles.

- **CommandDepth** depth of XML command. Used only with XML-type formats.

- **CopyStyle** \[string\] Copies all the features of an existing style into the current one.

- **DependsOn** The name of a style whose preamble should be output \textit{before} this one. This allows to ensure some ordering of the preamble snippets when macros definitions depend on one another.\(^6\)

- **EndLabelType** \[No_Label, Box, Filled_Box, Static\] The type of label that stands at the end of the paragraph (or sequence of paragraphs if \LaTeXType is \Environment, \ItemEnvironment or \ListEnvironment). \NoLabel means “nothing”, Box (resp. \Filled_Box) is a white (resp. black) square suitable for end of proof markers, Static is an explicit text string.

- **EndLabelString** \[string=""] The string used for a label with a \Static EndLabelType.

- **Fill_Bottom** \[0,1\] Similar to Fill_Top.

^4Note that this will either define a new layout or modify an existing one.

^5Note that a ‘float’ here is a real number, such as: 1.5.

^6Note that, besides that functionality, there is no way to ensure any ordering of preambles. The ordering that you see in a given version of \LaTeX\X may change without warning in later versions.
5.3 Declaring a new text class

**Fill_Top** \[0,1\] With this parameter the Fill value of the “Vertical space above” list of the Edit>Paragraph Settings dialog can be set when initializing a paragraph with this style.\(^7\)

**Font** The font used for both the text body and the label. See section 5.3.8. Note that defining this font automatically defines the LabelFont to the same value. So you should define this one first if you also want to define LabelFont.

**FreeSpacing** \[0, 1\] Usually LYX doesn’t allow you to insert more than one space between words, since a space is considered as the separation between two words, not a character or symbol of its own. This is a very fine thing but sometimes annoying, for example, when typing program code or plain \LaTeX\ code. For this reason, FreeSpacing can be enabled. Note that LYX will create protected blanks for the additional blanks when in another mode than \LaTeX\-mode.

**InnerTag** [[FIXME]] (Used only with XML-type formats.)

**InTitle** \[1, 0\] If 1, marks the layout as being part of a title block (see also the TitleLatexType and TitleLatexName global entries).

**ItemSep** \[\text{float}=0\] This provides extra space between paragraphs that have the same layout. If you put other layouts into an environment, each is separated with the environment’s Parsep. But the whole items of the environment are additionally separated with this Itemsep. Note that this is a multiplier.

**ItemTag** [[FIXME]] (Used only with XML-type formats.)

**KeepEmpty** \[0, 1\] Usually LYX doesn’t allow you to leave a paragraph empty, since it would lead to empty \LaTeX\ output. There are some cases where this could be desirable however: in a letter template, the required fields can be provided as empty fields, so that people do not forget them; in some special classes, a layout can be used as some kind of break, which does not contain actual text.

**LabelBottomsep** \[\text{float}=0\] The vertical space between the label and the text body. Only used for labels that are above the text body (Top_Environment, Centered_Top_Environment).

**LabelCounter** \[\text{string}=\”\”\] The name of the counter for automatic numbering (see Section 5.3.7 for details). This must be given if Labeltype is Counter.

**LabelFont** The font used for the label. See section 5.3.8.

**LabelIndent** Text that indicates how far a label should be indented.

**Labelsep** \[\text{string}=\”\”\] The horizontal space between the label and the text body. Only used for labels that are not above the text body.

---

\(^7\) Note from Jean-Marc: I’m not sure that this setting has much use, and it should probably be removed in later versions.
5 Installing New Document Classes

**LabelString** `[string=""]` The string used for a label with a Static labeltype. When `LabelCounter` is set, this string can be contain the special formatting commands described in Section 5.3.7.\(^8\)

**LabelStringAppendix** `[string=""]` This is used inside the appendix instead of LabelString. Note that every LabelString statement resets LabelStringAppendix too.

**LabelTag** `[FIXME]` (Used only with XML-type formats.)

**Labeltype** `[No_Label, Manual, Static, Top_Environment, Centered_Top_Environment, Sensitive, Counter]`
Manual means the label is the very first word (up to the first real blank).\(^9\) Static means it is defined in the layout (see LabelString). Top_Environment and Centered_Top_Environment are special cases of Static. The label will be printed above the paragraph, but only at the top of an environment or the top of a chain of paragraphs with this layout. Usage is for example the Abstract layout or the Bibliography layout. This is also the case for Manual labels with latex type Environment, in order to make layouts for theorems work correctly. Sensitive is a special case for the caption-labels “Figure” and “Table”. Sensitive means the (hardcoded) label string depends on the kind of float. The Counter label type defines automatically numbered labels. See Section 5.3.7.

**LatexName** The name of the corresponding \LaTeX stuff. Either the environment or command name.

**LatexParam** An optional parameter for the corresponding LatexName stuff. This parameter cannot be changed from within LYX.

**LatexType** `[Paragraph, Command, Environment, Item_Environment, List_Environment]`
How the layout should be translated into \LaTeX. Paragraph means nothing special. Command means `\LatexName{...}` and Environment means `\begin{LatexName}...\end{LatexName}`. Item_Environment is the same as Environment, except that a `\item` is generated for each paragraph of this environment. List_Environment is the same as Item_Environment, except that LabelWidthString is passed as an argument to the environment. LabelWidthString can be defined in the Layout>Paragraph dialog. LatexType is perhaps a bit misleading, since these rules apply to SGML classes, too. Visit the SGML class files for specific examples.

Putting the last few things together, the \LaTeX output will be either:

\[
\textbackslash latexname[latexparm]{...}
\]

or:

\^8For the sake of backwards compatibility, the string `@style-name@` will be replaced by the expanded LabelString of style `style-name`. This feature is now obsolete and should be replaced by the mechanisms of Section 5.3.7.

\^9Use protected spaces if you want more than one word as the label.
5.3 Declaring a new text class

\begin{latexname}[latexparam] ... \end{latexname}.

depending upon the \LaTeX type.

**LeftMargin [string=""]** If you put layouts into environments, the leftmargins are not simply added, but added with a factor $\frac{4}{\text{depth}+4}$. Note that this parameter is also used when the margin is defined as Manual or Dynamic. Then it is added to the manual or dynamic margin.

The argument is passed as a string. For example “MM” means that the paragraph is indented with the width of “MM” in the normal font. You can get a negative width by prefixing the string with “-”. This way was chosen so that the look is the same with each used screen font.

**Margin [Static, Manual, Dynamic, First_Dynamic, Right_Address_Box]**

The kind of margin that the layout has on the left side. Static just means a fixed margin. Manual means that the left margin depends on the string entered in the Edit $\triangledown$ Paragraph Settings dialog. This is used to typeset nice lists without tabulators. Dynamic means that the margin depends on the size of the label. This is used for automatic enumerated headlines. It is obvious that the headline “5.4.3.2.1 Very long headline” must have a wider left margin (as wide as “5.4.3.2.1” plus the space) than “3.2 Very long headline”, even if standard “word processors” are not able to do this. First_Dynamic is similar, but only the very first row of the paragraph is dynamic, while the others are static; this is used, for example, for descriptions. Right_Address_Box means the margin is chosen in a way that the longest row of this paragraph fits to the right margin. This is used to typeset an address on the right edge of the page.

**NeedProtect [0,1]** Whether fragile commands in this layout should be \protect’ed. (Note: This is not whether this command should itself be protected.)

**Newline [0,1]** Whether newlines are translated into \LaTeX newlines (\\) or not. The translation can be switched off to allow more comfortable \LaTeX editing inside LYX.

**NextNoIndent [1,0]** Whether the following Paragraph is allowed to indent its very first row. 1 means that it is not allowed to do so; 0 means it could do so if it wants to.

**ObsoletedBy** Name of a layout that has replaced this layout. This is used to rename a layout, while keeping backward compatibility.

**OptionalArgs [int=0]** The number of optional arguments that can be used with this layout. This is useful for things like section headings, and only makes sense with \LaTeX.
5 Installing New Document Classes

**ParIndent** [\texttt{string=”“}] The indent of the very first line of a paragraph. The \texttt{ParIndent} will be fixed for a certain layout. The exception is Standard layout, since the indentation of a Standard layout paragraph can be prohibited with \texttt{NextNoIndent}. Also, Standard layout paragraphs inside environments use the \texttt{Parindent} of the environment, not their native one. For example, Standard paragraphs inside an enumeration are not indented.

**Parsep** [\texttt{float=0}] The vertical space between two paragraphs of this layout.

**Parskip** [\texttt{float=0}] \LaTeX allows the user to choose either “indent” or “skip” to typeset a document. When “indent” is chosen, this value is completely ignored. When “skip” is chosen, the parindent of a \LaTeX type “Paragraph” layout is ignored and all paragraphs are separated by this parskip argument. The vertical space is calculated with \texttt{value * DefaultHeight} where \texttt{DefaultHeight} is the height of a row with the normal font. This way, the look stays the same with different screen fonts.

**PassThru** [0, 1] Whether the contents of this paragraph should be output in raw form, meaning without special translations that \LaTeX would require.

**Preamble** Information to be included in the \LaTeX preamble when this style is used. Used to define macros, load packages, etc., required by this particular style. Must end with “EndPreamble”.

**Requires** [\texttt{string}] Whether the layout requires the feature \texttt{string}. See the description of \texttt{Provides} above (page 471) for information on ‘features’.

**RightMargin** [\texttt{string=”“}] Similar to \texttt{LeftMargin}.

**Spacing** [\texttt{single, onehalf, double, other \texttt{value}}] This defines what the default spacing should be in the layout. The arguments \texttt{single, onehalf} and \texttt{double} correspond respectively to a multiplier value of 1, 1.25 and 1.667. If you specify the argument \texttt{other}, then you should also provide a numerical argument which will be the actual multiplier value. Note that, contrary to other parameters, \texttt{Spacing} implies the generation of specific \LaTeX code, using the package \texttt{setspace.sty}.

**TextFont** The font used for the text body. See section 5.3.8.

**TocLevel** [\texttt{int}] The level of the style in the table of contents. This is used for automatic numbering of section headings.

**TopSep** [\texttt{float=0}] The vertical space with which the very first of a chain of paragraphs with this layout is separated from the previous paragraph. If the previous paragraph has another layout, the separations are not simply added, but the maximum is taken.
5.3 Declaring a new text class

5.3.5 Floats

Since version 1.3.0 of LyX, it is has been both possible and necessary to define the floats (figure, table, ...) in the text class itself. Standard floats are included in the file stdfloats.inc, so you may have to do no more than add

\input stdfloats.inc

to your layout file. If you want to implement a text class that proposes some other float types (like the AGU class bundled with LyX), the information below will hopefully help you:

**Extension** [string=””] The file name extension of an auxiliary file for the list of figures (or whatever). \LaTeX{} writes the captions to this file.

**GuiName** [string=””] The string that will be used in the menus and also for the caption.

**\LaTeX{}Builtin** [0, 1] Set to 1 if the float is already defined by the \LaTeX{} document class. If this is set to 0, the float will be defined using the \LaTeX{} package `float`.

**ListName** [string=””] The heading used for the list of floats.

**NumberWithin** [string=””] This (optional) argument determines whether floats of this class will be numbered within some sectional unit of the document. For example, if within is equal to `chapter`, the floats will be numbered within chapters.

**Placement** [string=””] The default placement for the given class of floats. The string should be as in standard \LaTeX: t, b, p and h for top, bottom, page, and here, respectively. On top of that there is a new type, H, which does not really correspond to a float, since it means: put it “here” and nowhere else. Note however that the H specifier is special and, because of implementation details, cannot be used in non-built in float types. If you do not understand what this means, just use “tbp”.

**Style** [string=””] The style used when defining the float using `\newfloat`.

**Type** [string=””] The “type” of the new class of floats, like program or algorithm. After the appropriate `\newfloat`, commands such as `\begin{program}` or `\end{algorithm*}` will be available.

Note that defining a float with type *type* automatically defines the corresponding counter with name *type*.

\footnote{Note that the order of these letters in the string is irrelevant, like in \LaTeX{}.}
5.3.6 Flex insets and InsetLayout

\LaTeX{} has supported character styles since version 1.4.0; as of version 1.6.0, these are called Flex insets.

Flex insets come in three different kinds:

- character style (\texttt{CharStyle}): These define semantic markup corresponding to such \LaTeX{} commands as \texttt{\textbackslash noun} and \texttt{\textbackslash code}.

- user custom (\texttt{Custom}): These can be used to define custom collapsible insets, similar to \TeX{} code, footnote, and the like. An obvious example is an endnote inset, which is defined in the \texttt{endnote} module.

- XML elements (\texttt{Element}): For use with DocBook classes.

Flex insets are defined using the \texttt{InsetLayout} tag, which shall be explained in a moment.

The \texttt{InsetLayout} tag also serves another function: It can be used to customize the general layout of many different types of insets. Currently, \texttt{InsetLayout} can be used to customize the layout parameters for footnotes, marginal notes, note insets, \TeX{} code (ERT) insets, branches, listings, indexes, boxes, tables, algorithms, URLs, and optional arguments, as well as to define Flex insets.

The \texttt{InsetLayout} definition must begin with a line of the form:

\begin{verbatim}
InsetLayout <Type>
\end{verbatim}

Here \texttt{<Type>} indicates the inset whose layout is being defined, and here there are two cases.

1. The layout for a pre-existing inset is being modified. In this case, \texttt{<Type>} can be any one of the following: \texttt{Algorithm}, \texttt{Branch}, \texttt{Box}, \texttt{Box:shaded}, \texttt{ERT}, \texttt{Figure}, \texttt{Foot}, \texttt{Index}, \texttt{Info}, \texttt{Info:menu}, \texttt{Info:shortcut}, \texttt{Info:shortcuts}, \texttt{Listings}, \texttt{Marginal}, \texttt{Note:Comment}, \texttt{Note:Note}, \texttt{Note:GreyedOut}, \texttt{OptArg}, \texttt{Table}, or \texttt{URL}.

2. The layout for a Flex inset is being defined. In this case, \texttt{<Type>} can be any valid identifier not used by a pre-existing inset. Note that the definition of a flex inset must also include a \texttt{LyXType} entry.

The \texttt{InsetLayout} definition can contain the following entries:

\texttt{BgColor} The color for the inset’s background. The valid colors are defined in \texttt{src/ColorCode.h}.

\texttt{CopyStyle} As with paragraph styles (see page 5.3.4).

\texttt{CustomPars} \([0,1]\) Indicates whether the user may employ the Paragraph Settings dialog to customize the paragraph.
5.3 Declaring a new text class

Decoration can be Classic, Minimalistic, or Conglomerate, describing the rendering style used for the inset’s frame and buttons. Footnotes generally use Classic, ERT insets generally Minimalistic, and character styles Conglomerate.

End Required at the end of the InsetLayout declarations.

Font The font used for both the text body and the label. See section 5.3.8. Note that defining this font automatically defines the LabelFont to the same value, so define this first and define LabelFont later if you want them to be different.

ForceLTR Force the “latex” language, leading to Left-to-Right (latin) output, e.g. in \TeX code or URL. A kludge.

ForcePlain [0,1] Indicates whether the PlainLayout should be used or, instead, the user can change the paragraph style used in the inset.

FreeSpacing As with paragraph styles (see page 471).

KeepEmpty As with paragraph styles (see page 471).

LabelFont The font used for the label. See section 5.3.8. Note that this definition can never appear before Font, lest it be ineffective.

LabelString What will be displayed on the button or elsewhere as the inset label. Some inset types (\TeX code and Branch) modify this label on the fly.

LatexName The name of the corresponding \LaTeX stuff. Either the environment or command name.

LatexParam The optional parameter for the corresponding LatexName stuff, including possible bracket pairs like \[]. This parameter cannot be changed from within LYX.

LatexType As with paragraph styles (see page 472).

LyxType Can be charstyle, custom, element, or end (indicating a dummy definition ending definitions of charstyles, etc). This entry is required in and is only meaningful for Flex insets. Among other things, it determines on which menu this inset will appear.

MultiPar [0,1] Whether multiple paragraphs are permitted in this inset. This will also set CustomPars to the same value and ForcePlain to the opposite value. These can be reset to other values, if they are used after MultiPar.

NeedProtect [0,1] Whether fragile commands in this layout should be \protect’ed. (Note: This is not whether the command should itself be protected.)

PassThru [0,1] As with paragraph styles (see page 5.3.4).
Preamble  As with paragraph styles (see page 474).

Requires  [string] As with paragraph styles (see page 474).

5.3.7 Counters

Since version 1.3.0 of LYX, it is both possible and necessary to define the counters (chapter, figure, ...) in the text class itself. The standard counters are defined in the file stdcounters.inc, so you may have to do no more than add

Input stdcounters.inc

to your layout file to get them to work. But if you want to define custom counters, then you can do so. The counter declaration must begin with:

Counter name

where of course ‘name’ is replaced by the name of the counter. And it must end with “End”. The following parameters can also be used:

LabelString [string=”"] when this is defined, this string defines how the counter is displayed. Setting this value sets LabelStringAppendix to the same value. The following special constructs can be used in the string:

- \thecounter will be replaced by the expansion of the LabelString (or LabelStringAppendix) of the counter counter.
- counter values can be expressed using \LaTeX-like macros \texttt{\textbackslash numbertype\{counter\}}, where \texttt{numbertype} can be:\texttt{arabic}: 1, 2, 3,...; \texttt{alph} for lower-case letters: a, b, c,...; \texttt{Alph} for upper-case letters: A, B, C,...; \texttt{roman} for lower-case roman numerals: i, ii, iii,...; \texttt{Roman} for upper-case roman numerals: I, II, III...; \texttt{hebrew} for hebrew numerals.

If LabelString is not defined, a default value is constructed as follows: if the counter has a master counter \texttt{master} (defined via \texttt{Within}), the string \texttt{\textbackslash themaster.\arabic\{counter\}} is used; otherwise the string \texttt{\arabic\{counter\}} is used.

LabelStringAppendix [string=”"] Same as LabelString, but for use in the Appendix.

Within [string=””] If this is set to the name of another counter, the present counter will be reset every time the other one is increased. For example, \texttt{subsection} is numbered inside \texttt{section}.

\footnote{Actually, the situation is a bit more complicated: any \texttt{numbertype} other than those described below will produce arabic numerals. It would not be surprising to see this change in the future.}
5.3 Declaring a new text class

5.3.8 Font description

A font description looks like this:

```latex
Font or LabelFont
...
EndFont
```

The following commands are available:

**Color** [none, black, white, red, green, blue, cyan, magenta, yellow]

**Family** [Roman, Sans, Typewriter]

**Misc** [string] Valid argument are: emph, noun, underbar, no_emph, no_noun and no_bar. Each of these turns on or off the corresponding attribute.

**Series** [Medium, Bold]

**Shape** [Up, Italic, SmallCaps, Slanted]

**Size** [tiny, small, normal, large, larger, largest, huge, giant]

5.3.9 Upgrading old layout files

The file format of layout files changes from time to time, so old layout files need to be converted. This process has been automated since \LaTeX{} 1.4.0: If \LaTeX{} reads an old format layout file it will call the conversion tool \texttt{LYXDir/scripts/layout2layout.py} and convert it to a temporary file in current format. The original file is left untouched. If you want to convert the layout file permanently, just call the converter by hand:

```
python $LYXDir/scripts/layout2layout.py myclass.layout myclassnew.layout
```

(You need to replace \texttt{$LYXDir$} with the name of your \LaTeX{} system directory, unless you happen to have defined such an environment variable.) Then copy \texttt{myclassnew.layout} to \texttt{UserDir/layouts/}.

The automatic conversion only handles syntax changes. It cannot handle the case where the contents of included files was changed, so these will have to be converted separately.
5.4 Creating Templates

Templates are created just like usual documents. The only difference is that usual documents contain all possible settings, including the font scheme and the paper size. Usually a user doesn’t want a template to overwrite his defaults in these cases. For that reason, the designer of a template should remove the corresponding commands like `\fontscheme` or `\papersize` from the template LyX file. This can be done with any simple text-editor, for example vi or xedit.

Put the edited template files you create in `UserDir/templates/`, copy the ones you use from the global template directory in `LyXDir/templates/` to the same place, and redefine the template path in the Tools > Preferences > Paths dialog.

Note that there is a template which has a particular meaning: `defaults.lyx`. This template is loaded every time you create a new document with File > New in order to provide useful defaults. To create this template from inside LyX, all you have to do is to open a document with the correct settings, and use the Save as Document Defaults button.
6 Including External Material

**WARNING:** This portion of the documentation has not been updated for some time. We certainly hope that it is still accurate, but there are no guarantees.

The use of material from sources external to LyX is covered in detail in the *Embedded Objects* manual. This part of the manual covers what needs to happen behind the scenes for new sorts of material to be included.

### 6.1 How does it work?

The external material feature is based on the concept of a template. A template is a specification of how LyX should interface with a certain kind of material. As bundled, LyX comes with predefined templates for Xfig figures, various raster format images, chess diagrams, and LilyPond music notation. You can check the actual list by using the menu **Insert** ➤ **File** ➤ **External Material**. Furthermore, it is possible to roll your own template to support a specific kind of material. Later we’ll describe in more detail what is involved, and hopefully you will submit all the templates you create so we can include them in a later LyX version.

Another basic idea of the external material feature is to distinguish between the original file that serves as a base for final material and the produced file that is included in your exported or printed document. For example, consider the case of a figure produced with Xfig. The Xfig application itself works on an original file with the `.fig` extension. Within Xfig, you create and change your figure, and when you are done, you save the `fig`-file. When you want to include the figure in your document, you invoke *transfig* in order to create a PostScript file that can readily be included in your LyX file. In this case, the `.fig` file is the original file, and the PostScript file is the produced file.

This distinction is important in order to allow updating of the material while you are in the process of writing the document. Furthermore, it provides us with the flexibility that is needed to support multiple export formats. For instance, in the case of a plain text file, it is not exactly an award-winning idea to include the figure as raw PostScript. Instead, you’d either prefer to just include a reference to the figure or try to invoke some graphics to ASCII converter to make the final result look similar to the real graphics. The external material management allows you to do this, because it is parametrized on the different export formats that LyX supports.

Besides supporting the production of different products according to the exported format, it supports tight integration with editing and viewing applications. In the
case of an Xfig figure, you are able to invoke Xfig on the original file with a single click from within the external material dialog in \LaTeX, and also preview the produced PostScript file with Ghostview with another click. No more fiddling around with the command line and/or file browsers to locate and manipulate the original or produced files. In this way, you are finally able to take full advantage of the many different applications that are relevant to use when you write your documents, and ultimately be more productive.

6.2 The external template configuration file

It is relatively easy to add custom external template definitions to \LaTeX. However, be aware that doing this in an careless manner most probably will introduce an easily exploitable security hole. So before you do this, please read the discussion about security in section 6.4.

Having said that, we encourage you to submit any interesting templates that you create.

The external templates are defined in the \texttt{\ LyXDir/lib/external\_templates} file. You can place your own version in \texttt{UserDir/external\_templates}.

A typical template looks like this:

```plaintext
Template XFig
GuiName "XFig: $$\text{AbsOrRelPathParent}\$$\text{Basename}"
HelpText
An XFig figure.
HelpTextEnd
InputFormat fig
FileFilter "*.fig"
AutomaticProduction true
Transform Rotate
Transform Resize
Format \texttt{EPX}
TransformCommand Rotate RotationLatexCommand
TransformCommand Resize ResizeLatexCommand
Product "$$\text{RotateFront}\$$\text{ResizeFront}
\input{$$\text{AbsOrRelPathMaster}\$$\text{Basename}.pstex_t}$$\text{ResizeBack}$$\text{RotateBack}"
UpdateFormat pstex
UpdateResult "$$\text{AbsPath}\$$\text{Basename}.pstex_t"
Requirement "$\text{graphicx}$"
ReferencedFile latex "$$\text{AbsOrRelPathMaster}\$$\text{Basename}.pstex_t"
ReferencedFile latex "$$\text{AbsPath}\$$\text{Basename}.eps"
ReferencedFile dvi "$$\text{AbsPath}\$$\text{Basename}.eps"
FormatEnd
```
As you can see, the template is enclosed in Template ... TemplateEnd. It contains a header specifying some general settings and, for each supported primary document file format, a section Format ... FormatEnd.

### 6.2.1 The template header

**AutomaticProduction true|false** Whether the file represented by the template must be generated by LyX. This command must occur exactly once.

**FileFilter <pattern>** A glob pattern that is used in the file dialog to filter out the desired files. If there is more than one possible file extension (e.g. tgif has .obj and .tgo), use something like "*.{obj,tgo}". This command must occur exactly once.

**GuiName <guiname>** The text that is displayed on the button. This command must occur exactly once.
HelpText <text> HelpTextEnd The help text that is used in the External dialog. Provide enough information to explain to the user just what the template can provide him with. This command must occur exactly once.

InputFormat <format> The file format of the original file. This must be the name of a format that is known to \LaTeX{} (see section 3.1). Use "*" if the template can handle original files of more than one format. \LaTeX{} will attempt to interrogate the file itself in order to deduce its format in this case. This command must occur exactly once.

Template <id> A unique name for the template. It must not contain substitution macros (see below).

Transform Rotate|Resize|Clip|Extra This command specifies which transformations are supported by this template. It may occur zero or more times. This command enables the corresponding tabs in the external dialog. Each Transform command must have either a corresponding TransformCommand or a TransformOption command in the Format section. Otherwise the transformation will not be supported by that format.

6.2.2 The Format section

Format \documentclass|PDF|PlainText|DocBook The primary document file format that this format definition is for. Not every template has a sensible representation in all document file formats. Please define nevertheless a Format section for all formats. Use a dummy text when no representation is available. Then you can at least see a reference to the external material in the exported document.

Option <name> <value> This command defines an additional macro $$<name> for substitution in Product. <value> itself may contain substitution macros. The advantage over using <value> directly in Product is that the substituted value of $$<name> is sanitized so that it is a valid optional argument in the document format. This command may occur zero or more times.

Product <text> The text that is inserted in the exported document. This is actually the most important command and can be quite complex. This command must occur exactly once.

Preamble <name> This command specifies a preamble snippet that will be included in the \LaTeX{} preamble. It has to be defined using PreambleDef ... PreambleDefEnd. This command may occur zero or more times.

ReferencedFile <format> <filename> This command denotes files that are created by the conversion process and are needed for a particular export format. If the filename is relative, it is interpreted relative to the master document. This command may be given zero or more times.
6.3 The substitution mechanism

**Requirement <package>** The name of a required \LaTeX\ package. The package is included via `\usepackage{}` in the \LaTeX\ preamble. This command may occur zero or more times.

**TransformCommand Rotate RotationLatexCommand** This command specifies that the built in \LaTeX\ command should be used for rotation. This command may occur once or not at all.

**TransformCommand Resize ResizeLatexCommand** This command specifies that the built in \LaTeX\ command should be used for resizing. This command may occur once or not at all.

**TransformOption Rotate RotationLatexOption** This command specifies that rotation is done via an optional argument. This command may occur once or not at all.

**TransformOption Resize ResizeLatexOption** This command specifies that resizing is done via an optional argument. This command may occur once or not at all.

**TransformOption Clip ClipLatexOption** This command specifies that clipping is done via an optional argument. This command may occur once or not at all.

**TransformOption Extra ExtraLatexOption** This command specifies that an extra optional argument is used. This command may occur once or not at all.

**UpdateFormat <format>** The file format of the converted file. This must be the name of a format that is known to \LaTeX\ (see the Tools Preferences:Conversion dialog). This command must occur exactly once.

**UpdateResult <filename>** The file name of the converted file. The file name must be absolute. This command must occur exactly once.

### 6.2.3 Preamble definitions

The external template configuration file may contain additional preamble definitions enclosed by `PreambleDef ... PreambleDefEnd`. They can be used by the templates in the Format section.

### 6.3 The substitution mechanism

When the external material facility invokes an external program, it is done on the basis of a command defined in the template configuration file. These commands can contain various macros that are expanded before execution. Execution always take place in the directory of the containing document.
6 Including External Material

Also, whenever external material is to be displayed, the name will be produced by
the substitution mechanism, and most other commands in the template definition
support substitution as well.

The available macros are the following:

$$\text{AbsOrRelPathMaster}$$ The file path, absolute or relative to the master \texttt{LYX} doc-
ument.

$$\text{AbsOrRelPathParent}$$ The file path, absolute or relative to the \texttt{LYX} document.

$$\text{AbsPath}$$ The absolute file path.

$$\text{Basename}$$ The filename without path and without the extension.

$$\text{Contents("filename.ext")}$$ This macro will expand to the contents of the file
with the name \texttt{filename.ext}.

$$\text{Extension}$$ The file extension (including the dot).

$$\text{FName}$$ The filename of the file specified in the external material dialog. This is
either an absolute name, or it is relative to the \texttt{LYX} document.

$$\text{FPath}$$ The path part of $$\text{FName}$$ (absolute name or relative to the \texttt{LYX} document).

$$\text{RelPathMaster}$$ The file path, relative to the master \texttt{LYX} document.

$$\text{RelPathParent}$$ The file path, relative to the \texttt{LYX} document.

$$\text{Sysdir}$$ This macro will expand to the absolute path of the system directory. This
is typically used to point to the various helper scripts that are bundled with
\texttt{LYX}.

$$\text{Tempname}$$ A name and full path to a temporary file which will be automatically
deleted whenever the containing document is closed, or the external material
insertion deleted.

All path macros contain a trailing directory separator, so you can construct e.g. the
absolute filename with $$\text{AbsPath}\text{Basename}\text{Extension}$$.

The macros above are substituted in all commands unless otherwise noted. The
command \texttt{Product} supports additionally the following substitutions if they are en-
abled by the \texttt{Transform} and \texttt{TransformCommand} commands:

$$\text{ResizeFront}$$ The front part of the resize command.

$$\text{ResizeBack}$$ The back part of the resize command.

$$\text{RotateFront}$$ The front part of the rotation command.

$$\text{RotateBack}$$ The back part of the rotation command.

486
The value string of the `Option` command supports additionally the following substitutions if they are enabled by the `Transform` and `TransformOption` commands:

- `$Clip` The clip option.
- `$Extra` The extra option.
- `$Resize` The resize option.
- `$Rotate` The rotation option.

You may ask why there are so many path macros. There are mainly two reasons:

1. Relative and absolute file names should remain relative or absolute, respectively. Users may have reasons to prefer either form. Relative names are useful for portable documents that should work on different machines, for example. Absolute names may be required by some programs.

2. \LaTeX treats relative file names differently than LyX and other programs in nested included files. For LyX, a relative file name is always relative to the document that contains the file name. For \LaTeX, it is always relative to the master document. These two definitions are identical if you have only one document, but differ if you have a master document that includes part documents. That means that relative filenames must be transformed when presented to \LaTeX. Fortunately LyX does this automatically for you if you choose the right macros.

So which path macro should be used in new template definitions? The rule is not difficult:

- Use `$AbsPath` if an absolute path is required.
- Use `$AbsOrRelPathMaster` if the substituted string is some kind of \LaTeX input.
- Else use `$AbsOrRelPathParent` in order to preserve the user's choice.

There are special cases where this rule does not work and e.g. relative names are needed, but normally it will work just fine. One example for such a case is the command `ReferencedFile latex "$AbsOrRelPathMaster$Basename.pstex_t"` in the XFig template above: We can't use the absolute name because the copier for `.pstex_t` files needs the relative name in order to rewrite the file content.

### 6.4 Security discussion

The external material feature interfaces with a lot of external programs and does so automatically, so we have to consider the security implications of this. In particular, since you have the option of including your own filenames and/or parameter strings
Including External Material

and those are expanded into a command, it seems that it would be possible to create a malicious document which executes arbitrary commands when a user views or prints the document. This is something we definitely want to avoid.

However, since the external program commands are specified in the template configuration file only, there are no security issues if LyX is properly configured with safe templates only. This is so because the external programs are invoked with the execvp-system call rather than the system system-call, so it’s not possible to execute arbitrary commands from the filename or parameter section via the shell.

This also implies that you are restricted in what command strings you can use in the external material templates. In particular, pipes and redirection are not readily available. This has to be so if LyX should remain safe. If you want to use some of the shell features, you should write a safe script to do this in a controlled manner, and then invoke the script from the command string.

It is possible to design a template that interacts directly with the shell, but since this would allow a malicious user to execute arbitrary commands by writing clever filenames and/or parameters, we generally recommend that you only use safe scripts that work with the execvp system call in a controlled manner. Of course, for use in a controlled environment, it can be tempting to just fall back to use ordinary shell scripts. If you do so, be aware that you will provide an easily exploitable security hole in your system. Of course it stands to reason that such unsafe templates will never be included in the standard LyX distribution, although we do encourage people to submit new templates in the open source tradition. But LyX as shipped from the official distribution channels will never have unsafe templates.

Including external material provides a lot of power, and you have to be careful not to introduce security hazards with this power. A subtle error in a single line in an innocent looking script can open the door to huge security problems. So if you do not fully understand the issues, we recommend that you consult a knowledgeable security professional or the LyX development team if you have any questions about whether a given template is safe or not. And do this before you use it in an uncontrolled environment.
LyX Keyboard Shortcuts

By the LyX Team

August 17, 2009

This file lists the currently used shortcuts defined in bind file /home/paf/.lyx/bind/my.bind. You can change the shortcuts by selecting another bind file in the User Interface panel of the preference dialog (Tools > Preferences...).

### Buffer

<table>
<thead>
<tr>
<th>Action</th>
<th>Shortcut</th>
<th>Action</th>
<th>Shortcut</th>
<th>Action</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Ctrl+N</td>
<td>View DVI</td>
<td>Ctrl+D</td>
<td>Master View DVI</td>
<td>Ctrl+Alt+D</td>
</tr>
<tr>
<td>Open</td>
<td>Ctrl+O</td>
<td>View PS</td>
<td>Ctrl+T</td>
<td>Master View PS</td>
<td>Ctrl+Alt+T</td>
</tr>
<tr>
<td>Close</td>
<td>Ctrl+W</td>
<td>View PDF</td>
<td>undefined</td>
<td>Master View PDF</td>
<td>undefined</td>
</tr>
<tr>
<td>Save</td>
<td>Ctrl+S</td>
<td>Next</td>
<td>Ctrl+Tab</td>
<td>Previous</td>
<td>Ctrl+PgUp</td>
</tr>
</tbody>
</table>

### Edit

<table>
<thead>
<tr>
<th>Action</th>
<th>Shortcut</th>
<th>Action</th>
<th>Shortcut</th>
<th>Action</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>Ctrl+C</td>
<td>Bold</td>
<td>Ctrl+B</td>
<td>Search</td>
<td>F3</td>
</tr>
<tr>
<td>Paste</td>
<td>Ctrl+V</td>
<td>Emphasize</td>
<td>Ctrl+E</td>
<td>Spellcheck</td>
<td>F7</td>
</tr>
<tr>
<td>Cut</td>
<td>Ctrl+X</td>
<td>Typewriter</td>
<td>Ctrl+Shift+P</td>
<td>Save Bookmark 1</td>
<td>Ctrl+Alt+1</td>
</tr>
<tr>
<td>Undo</td>
<td>Ctrl+Z</td>
<td>Underline</td>
<td>Ctrl+U</td>
<td>Goto Bookmark 1</td>
<td>Ctrl+1</td>
</tr>
<tr>
<td>Redo</td>
<td>Ctrl+Shift+Z</td>
<td>Noun</td>
<td>Alt+C C</td>
<td>Goto last bookmark</td>
<td>Ctrl+&lt;</td>
</tr>
<tr>
<td>Normal Space</td>
<td>Ctrl+Alt+Space</td>
<td>newline</td>
<td>Ctrl+Return</td>
<td>Depth Increment</td>
<td>Alt+Shift+Right</td>
</tr>
<tr>
<td>Thin Space</td>
<td>Ctrl+Shift+Space</td>
<td>Linebreak</td>
<td>Ctrl+Shift+Return</td>
<td>Depth Decrement</td>
<td>Alt+Shift+Left</td>
</tr>
<tr>
<td>Append Column</td>
<td>Alt+M C I</td>
<td>Append Row</td>
<td>Alt+M W I</td>
<td>Copy Column</td>
<td>Alt+M C C</td>
</tr>
<tr>
<td>Delete Column</td>
<td>Alt+M C D</td>
<td>Delete Row</td>
<td>Alt+M W D</td>
<td>Copy Row</td>
<td>Alt+M W C</td>
</tr>
</tbody>
</table>

### Greek Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Shortcut</th>
<th>Symbol</th>
<th>Shortcut</th>
<th>Symbol</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>Alt+M G A</td>
<td>ν</td>
<td>Alt+M G N</td>
<td>Δ</td>
<td>Alt+M G Shift+D</td>
</tr>
<tr>
<td>β</td>
<td>Alt+M G B</td>
<td>ω</td>
<td>Alt+M G O</td>
<td>ε</td>
<td>Alt+M G Shift+E</td>
</tr>
<tr>
<td>χ</td>
<td>Alt+M G C</td>
<td>π</td>
<td>Alt+M G P</td>
<td>Φ</td>
<td>Alt+M G Shift+F</td>
</tr>
<tr>
<td>δ</td>
<td>Alt+M G D</td>
<td>ϑ</td>
<td>Alt+M G Q</td>
<td>Γ</td>
<td>Alt+M G Shift+G</td>
</tr>
<tr>
<td>ε</td>
<td>Alt+M G E</td>
<td>ρ</td>
<td>Alt+M G R</td>
<td>Λ</td>
<td>Alt+M G Shift+L</td>
</tr>
<tr>
<td>φ</td>
<td>Alt+M G F</td>
<td>σ</td>
<td>Alt+M G S</td>
<td>Π</td>
<td>Alt+M G Shift+P</td>
</tr>
<tr>
<td>γ</td>
<td>Alt+M G G</td>
<td>τ</td>
<td>Alt+M G T</td>
<td>Σ</td>
<td>Alt+M G Shift+S</td>
</tr>
<tr>
<td>η</td>
<td>Alt+M G H</td>
<td>υ</td>
<td>Alt+M G U</td>
<td>ζ</td>
<td>Alt+M G Shift+T</td>
</tr>
<tr>
<td>ι</td>
<td>Alt+M G I</td>
<td>θ</td>
<td>Alt+M G V</td>
<td>Γ</td>
<td>Alt+M G Shift+U</td>
</tr>
<tr>
<td>ϕ</td>
<td>Alt+M G J</td>
<td>ω</td>
<td>Alt+M G O</td>
<td>Θ</td>
<td>Alt+M G Shift+V</td>
</tr>
<tr>
<td>κ</td>
<td>Alt+M G K</td>
<td>ξ</td>
<td>Alt+M G X</td>
<td>Ω</td>
<td>Alt+M G Shift+O</td>
</tr>
<tr>
<td>λ</td>
<td>Alt+M G L</td>
<td>ψ</td>
<td>Alt+M G Y</td>
<td>Ξ</td>
<td>Alt+M G Shift+X</td>
</tr>
<tr>
<td>μ</td>
<td>Alt+M G M</td>
<td>ζ</td>
<td>Alt+M G Z</td>
<td>Ψ</td>
<td>Alt+M G Shift+Y</td>
</tr>
</tbody>
</table>
### Mathematical Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Math Mode</th>
<th>Symbol Shortcut</th>
<th>Displayed Math</th>
<th>Symbol Shortcut</th>
<th>Symbol</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math Mode</td>
<td>Ctrl+M</td>
<td>( \frac{a}{b} )</td>
<td>Ctrl+Shift+M</td>
<td>( \sqrt{a} )</td>
<td>Alt+M F</td>
<td>Alt+M -</td>
</tr>
<tr>
<td>Displayed Math</td>
<td>Ctrl+Shift+M</td>
<td>( \hat{a} )</td>
<td>Ctrl+Shift+M</td>
<td>( \bar{a} )</td>
<td>Alt+M S</td>
<td>Alt+M .</td>
</tr>
<tr>
<td>( \sum )</td>
<td>Alt+M U ( \bar{a} )</td>
<td>Alt+M &quot; ( \ddot{a} )</td>
<td>Alt+M Shift+V</td>
<td>Alt+M Shift+U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \int )</td>
<td>Alt+M I ( \dot{a} )</td>
<td>Alt+M H ( \breve{a} )</td>
<td>Alt+M V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \partial )</td>
<td>Alt+M P ( \ddot{a} )</td>
<td>Alt+M ( \bar{a} )</td>
<td>Alt+M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \infty )</td>
<td>Alt+M 8 ( \check{a} )</td>
<td>Alt+M &amp; ( \tilde{a} )</td>
<td>Alt+M B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \prime )</td>
<td>Alt+M ( \ddot{a} )</td>
<td>Alt+M ( \dot{a} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \Rightarrow )</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \Leftarrow )</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \rightarrow )</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \leftarrow )</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \uparrow )</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \downarrow )</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Delimiters

<table>
<thead>
<tr>
<th>Delimiter</th>
<th>Shortcut</th>
<th>Delimiter</th>
<th>Shortcut</th>
<th>Delimiter</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>( )</td>
<td>Alt+M (</td>
<td>{ }</td>
<td>Alt+M {</td>
<td>) }</td>
<td>Alt+M &gt;</td>
</tr>
<tr>
<td>[ ]</td>
<td>Alt+M [</td>
<td>( )</td>
<td>Alt+M &lt;</td>
<td>] ]</td>
<td>Alt+M ]</td>
</tr>
<tr>
<td>( undefined)</td>
<td>undefined</td>
<td>{ undefined)</td>
<td>undefined</td>
<td>{ undefined</td>
<td>undefined</td>
</tr>
</tbody>
</table>
**LFUN\_ACCENT\_ACUTE.**

*Action:* Adds an acute accent to the next character typed.

*Syntax:* accent-acute

---

**LFUN\_ACCENT\_BREVE.**

*Action:* Adds a breve accent to the next character typed.

*Syntax:* accent-breve

---

**LFUN\_ACCENT\_CARON.**

*Action:* Adds a caron to the next character typed.

*Syntax:* accent-caron

---

**LFUN\_ACCENT\_CEDILLA.**

*Action:* Adds a cedilla to the next character typed.

*Syntax:* accent-cedilla

---

**LFUN\_ACCENT\_CIRCLE.**

*Action:* Adds a circle accent to the next character typed.

*Syntax:* accent-circle

---

**LFUN\_ACCENT\_CIRCUMFLEX.**

*Action:* Adds a circumflex to the next character typed.

*Syntax:* accent-circumflex

---

**LFUN\_ACCENT\_DOT.**

*Action:* Adds a dot accent to the next character typed.

*Syntax:* accent-dot

---

**LFUN\_ACCENT\_GRAVE.**

*Action:* Adds a grave accent to the next character typed.

*Syntax:* accent-grave

---

**LFUN\_ACCENT\_HUNGARIAN\_UMLAUT.**

*Action:* Adds a Hungarian umlaut to the next character typed.

*Syntax:* accent-grave

---

**LFUN\_ACCENT\_MACRON.**

*Action:* Adds a macron to the next character typed.

*Syntax:* accent-macron

---

**LFUN\_ACCENT\_OGONEK.**

*Action:* Adds an ogonek accent to the next character typed.

*Syntax:* accent-ogonek

---

**LFUN\_ACCENT\_TIE.**

*Action:* Adds a tie over the next two characters typed.

*Notion:* The following character will finish the tie.

*Syntax:* accent-tie

---

**LFUN\_ACCENT\_TILDE.**

*Action:* Adds a tilde over the next character typed.

*Syntax:* accent-tilde

---

**LFUN\_ACCENT\_UMLAUT.**

*Action:* Adds an umlaut over the next character typed.

*Syntax:* accent-umlaut
LFUN_ACCENT_UNDERBAR.
  Action: Adds a bar under the next character typed.
  Syntax: accent-underbar

LFUN_ACCENT_UNDERDOT.
  Action: Adds a dot under the next character typed.
  Syntax: accent-underdot

LFUN.Caption_INSERT.
  Action: Inserts a caption inset.
  Syntax: caption-insert
  Origin: Lgb, 18 Jul 2000

LFUN_DATE_INSERT.
  Action: Inserts the current date.
  Syntax: date-insert [<ARG>]
  Params: <ARG>: Format of date. The default value (%x) can be set in Preferences->Date format.
  For possible formats see manual page of strftime function.
  Origin: jdblair, 31 Jan 2000

LFUN_FOOTNOTE_INSERT.
  Action: Inserts a footnote inset.
  Syntax: footnote-insert
  Origin: Jug, 7 Mar 2000

LFUN_ERT_INSERT.
  Action: Inserts an ERT inset.
  Syntax: ert-insert
  Origin: Jug, 18 Feb 2000

LFUN_FLOAT_INSERT.
  Action: Inserts a float inset.
  Syntax: float-insert <TYPE>
  Params: <TYPE>: type of float depends on the used textclass. Usually "algorithm", "table", "figure" parameters can be given.
  Origin: Lgb, 27 Jun 2000

LFUN_FLOAT_WIDE_INSERT.
  Action: Inserts float insets as in LFUN_FLOAT_INSERT but span multiple columns.
  Notion: Corresponds to the starred floats (figure*, table*, etc.) in \LaTeX.
  Syntax: float-wide-insert <TYPE>
  Params: <TYPE>: type of float depends on the used textclass. Usually "algorithm", "table", "figure" parameters can be given.
  Origin: Lgb, 31 Oct 2001

LFUN_FLOAT_LIST_INSERT.
  Action: Inserts the list of floats in the document.
  Syntax: float-list-insert <TYPE>
  Params: <TYPE>: type of float depends on the used textclass. Usually "algorithm", "table", "figure" parameters can be given.
  Origin: Lgb, 3 May 2001

LFUN_WRAP_INSERT.
  Action: Inserts floats wrapped by the text around.
  Syntax: wrap-insert <TYPE>
  Params: <TYPE>: table|figure
  Origin: Dekel, 7 Apr 2002

LFUN_OPTIONAL_INSERT.
  Action: Inserts an optional-argument (short title) inset.
  Syntax: optional-insert
  Origin: vermeer, 12 Aug 2002
**LFUN_LINE_INSERT.**

Action: Inserts a horizontal line.
Syntax: line-insert
Origin: Andre, Oct 27 2003

**LFUN_NEWPAGE_INSERT.**

Action: Inserts a new page.
Syntax: newpage-insert <ARG>
Params: <ARG>: <newpage|pagebreak|clearpage|cleardoublepage> default: newpage
Origin: uwestoehr, 24 Nov 2007

**LFUN_MARGINALNOTE_INSERT.**

Action: Inserts a marginal note.
Syntax: marginalnote-insert
Origin: Lgb, 26 Jun 2000

**LFUN_UNICODE_INSERT.**

Action: Inserts a single unicode character.
Syntax: unicode-insert <CHAR>
Params: <CHAR>: The character to insert, given as its code point, in hexadecimal.
Sample: unicode-insert 0x0100
Origin: Lgb, 22 Oct 2006

**LFUN_LISTING_INSERT.**

Action: Inserts a new listings inset.
Syntax: listing-insert
Origin: Herbert, 10 Nov 2001; bpeng, 2 May 2007

**LFUN_TAB_INSERT.**

Action: Insert a tab into a listings inset.
Notion: It also works on a selection.
Syntax: tab-insert
Origin: vfvanravesteijn, Sep 30 2008

**LFUN_TAB_DELETE.**

Action: Delete a tab or up to an equivalent amount of spaces from a listings inset.
Notion: It also works on a selection - it removes a tab or spaces from the beginning of each line spanned by the selection. This is useful if you want to indent/unindent multiple lines in one action.
Syntax: tab-delete
Origin: vfvanravesteijn, Sep 30 2008

**LFUN_QUOTE_INSERT.**

Action: Inserts quotes according to the type and quote-language preference.
Notion: Currently English, Swedish, German, Polish, French, Danish quotes are distinguished.
Syntax: quote-insert [<TYPE>]
Params: <TYPE>: 'single' for single quotes, otherwise double quotes will be used.

**LFUN_INFO_INSERT.**

Action: Displays shortcuts, lyxrc, package and textclass availability and menu information in a non-editable boxed InsetText.
Notion: Apart from lfun arguments you can use the following method:
1. input the type and argument of this inset, e.g. "menu paste", in the work area.
2. select the text and run info-insert lfun.
Syntax: info-insert <TYPE> <ARG>
Params: <TYPE>: shortcut|lyxrc|package|textclass|menu|buffer
<ARG>: argument for a given type. Look into InsetInfo.h for detailed description.
Origin: bpeng, 7 Oct 2007
LFUN_BRANCH_INSERT.

Action: Inserts branch inset.
Syntax: branch-insert <BRANCH-NAME>
Origin: vermeer, 17 Aug 2003

LFUN_BOX_INSERT.

Action: Inserts Box inset.
Syntax: box-insert [<TYPE>]
Params: <TYPE>: Boxed|Frameless|Framed|Ovalbox|Ovalbox|Shadowbox|Shaded|Doublebox
Framed is the default one.
Origin: vermeer, 7 Oct 2003

LFUN_FLEX_INSERT.

Action: Inserts CharStyle, Custom inset or XML short element.
Notion: Look into the Customization manual for more information about these elements.
    To make this command enabled the layout file for the document class you're using has to load the character styles. There are a few contained in the Logical Markup module. You can also of course create some yourself.
    For dissolving the element see LFUN_INSET_DISSOLVE.
Syntax: flex-insert <TYPE:Name>
Params: TYPE: CharStyle|Custom|Element|Standard
    Identifies whether this is a Character Style, a Custom Inset or an XML Element, and which dynamical sub-menu this flex inset is in on the LyX menu tree. If Standard (currently unused): none of these. Name: This name must be defined either in your layout file or imported by some module. The definition is
    InsetLayout <TYPE:Name>
Sample: flex-insert CharStyle:Code

LFUN_SELF_INSERT.

Action: Inserts the given string (accordingly to the correct layout).
Notion: Automatically replace the currently selected text. Depends on lyrc settings "auto_region_delete".
Syntax: self-insert <STRING>

LFUN_SPACE_INSERT.

Action: Inserts one of horizontal space insets.
Syntax: space-insert <NAME> [<LEN>]
Params: <NAME>: normal, protected, thin, quad, qquad, enspace, enskip, negthinspace, hfill, hfill*, dotfill, hrulefill, hspace, hspace*
<LEN>: length for custom spaces (hspace, hspace* for protected)
Origin: JSpitzm, 20 May 2003, Mar 17 2008

LFUN_HYPERLINK_INSERT.

Action: Inserts hyperlinks into the document (clickable in pdf output).
Notion: Hyperlink target can be set via selection + hyperlink-insert function.
Syntax: href-insert [<TARGET>]

LFUN_SPECIALCHAR_INSERT.

Action: Inserts various characters into the document.
Syntax: specialchar-insert <CHAR>
Origin: JSpitzm, 6 Dec 2007

LFUN_TOC_INSERT.

Action: Inserts table of contents.
Syntax: toc-insert
Origin: Lgb, 27 May 97
**LFUN_APPENDIX.**

**Action:** Start (or remove) Appendix on the given cursor position.
**Syntax:** `appendix`
**Origin:** ettrich, 5 May 1998

**LFUN_INDEX_INSERT.**

**Action:** Inserts Index entry.
**Notion:** It automatically takes the word on the cursor position.
**Syntax:** `index-insert`
**Origin:** leeming, 3 Aug 2000

**LFUN_INDEX_PRINT.**

**Action:** Inserts list of Index entries on a new page.
**Syntax:** `index-print`
**Origin:** Lgb, 27 Feb 1997

**LFUN_NOMENCL_INSERT.**

**Action:** Inserts Nomenclature entry.
**Notion:** It automatically takes the word on the cursor position if no symbol is given.
**Syntax:** `nomencl-insert [ <SYMBOL> ]`
**Origin:** Ugras, 4 Nov 2006

**LFUN_NOMENCLATURE_PRINT.**

**Action:** Inserts list of Nomenclature entries.
**Syntax:** `nomenclature-print`
**Origin:** Ugras, 4 Nov 2006

**LFUN_NOTE_INSERT.**

**Action:** Inserts Note on the current cursor position, move selection inside the inset.
**Syntax:** `note-insert [ <TYPE> ]`
**Params:** `<TYPE>`: `<Note|Greyedout|Comment>` default: Note

**LFUN_NOTE_NEXT.**

**Action:** Moves the cursor to the beginning of next Note inset.
**Syntax:** `note-next`

**LFUN_NOTES_MUTATE.**

**Action:** Changes all Note insets of a particular type (source) to a different type (target) for the current document.
**Syntax:** `notes-mutate <SOURCE> <TARGET>`
**Params:** `<SOURCE/TARGET>`: `<Note|Comment|Greyedout`
**Origin:** sanda, 18 Jun 2008

**LFUN_NEWLINE_INSERT.**

**Action:** Inserts a line break or new line.
**Syntax:** `newline-insert [ <ARG> ]`
**Params:** `<ARG>`: `<newline|linebreak>` default: newline
**Origin:** JSpitzm, 25 Mar 2008

**LFUN_ESCAPE.**

**Action:** Clears the selection. If no text is selected call LFUN_FINISHED_FORWARD.
**Syntax:** `escape`
**Origin:** Lgb, 17 May 2001

**LFUN_DOWN.**

**Action:** Moves the cursor one line in downward direction.
**Syntax:** `down`

**LFUN_UP.**

**Action:** Moves the cursor one line in upward direction.
**Syntax:** `up`
LFUN_DOWN_SELECT.
Action: Moves the cursor one line in downward direction adding the current position to the selection.
Syntax: down-select

LFUN_UP_SELECT.
Action: Moves the cursor one line in upward direction adding the current position to the selection.
Syntax: up-select

LFUN_SCREEN_DOWN.
Action: Moves the cursor one page in downward direction.
Syntax: screen-down

LFUN_SCREEN_UP.
Action: Moves the cursor one page in upward direction.
Syntax: screen-up

LFUN_SCREEN_DOWN_SELECT.
Action: Moves the cursor one screen in downward direction adding the current position to the selection.
Syntax: screen-down-select

LFUN_SCREEN_UP_SELECT.
Action: Moves the cursor one page in upward direction adding the current position to the selection.
Syntax: screen-up-select

LFUN_SCROLL.
Action: Scroll the buffer view.
Notion: Only scrolls the screen up or down; does not move the cursor.
Syntax: scroll <TYPE> <QUANTITY>
Params: <TYPE>: line|page
<QUANTITY>: up|down|<number>
Origin: Abdel, Dec 27 2007

LFUN_SCREEN_RECENTER.
Action: Recenters the screen on the current cursor position.
Syntax: screen-recenter

LFUN_SCREEN_SHOW_CURSOR.
Action: Repositions the screen such that the cursor is visible.
Syntax: screen-show-cursor

LFUN_CHAR_BACKWARD.
Action: Moves the cursor one position logically backwards.
Notion: This is not the action which should be bound to the arrow keys, because backwards may be left or right, depending on the language. The arrow keys should be bound to LFUN_CHAR_LEFT or LFUN_CHAR_RIGHT actions, which in turn may employ this one.
Syntax: char-backward

LFUN_CHAR_BACKWARD_SELECT.
Action: Moves the cursor one position logically backwards, adding traversed position to the selection.
Notion: See also LFUN_CHAR_BACKWARD.
Syntax: char-backward-select

LFUN_CHAR_DELETE_BACKWARD.
Action: Deletes one character in the backward direction (usually the "BackSpace" key).
Syntax: char-delete-backward

LFUN_CHAR_DELETE_FORWARD.
Action: Deletes one character in the backward direction (usually the "Delete" key).
Syntax: char-delete-forward
LFUN_CHAR_FORWARD.
Action: Moves the cursor one position logically forward.
Notion: This is not the action which should be bound to the arrow keys, because forward may be left or right, depending on the language. The arrow keys should be bound to LFUN_CHAR_LEFT or LFUN_CHAR_RIGHT actions, which in turn may employ this one.
Syntax: char-forward

LFUN_CHAR_FORWARD_SELECT.
Action: Moves the cursor one position logically forward, adding traversed position to the selection.
Notion: See also LFUN_CHAR_FORWARD.
Syntax: char-forward-select

LFUN_CHAR_LEFT.
Action: Moves the cursor one position "to the left".
Notion: This is the action which should be taken when the "left" key is pressed. Generally, it moves the cursor one position to the left. However, in Bidi text this become slightly more complicated, and there are different modes of cursor movement. In "visual mode", this moves left, plain and simple. In "logical mode", movement is logically forward in RTL paragraphs, and logically backwards in LTR paragraphs.
Syntax: char-left

LFUN_CHAR_LEFT_SELECT.
Action: Moves the cursor one position "to the left", adding traversed position to the selection.
Notion: See also LFUN_CHAR_LEFT for exact details of the movement.
Syntax: char-left-select

LFUN_CHAR_RIGHT.
Action: Moves the cursor one position "to the right".
Notion: This is the action which should be taken when the "right" key is pressed. Generally, it moves the cursor one position to the right. However, in Bidi text this become slightly more complicated, and there are different modes of cursor movement. In "visual mode", this moves right, plain and simple. In "logical mode", movement is logically forward in LTR paragraphs, and logically backwards in RTL paragraphs.
Syntax: char-right

LFUN_CHAR_RIGHT_SELECT.
Action: Moves the cursor one position "to the right", adding traversed position to the selection.
Notion: See also LFUN_CHAR_RIGHT for exact details of the movement.
Syntax: char-right-select

LFUN_WORD_BACKWARD.
Action: Moves the cursor to the logically previous beginning of a word.
Notion: This is not the action which should be bound to the arrow keys, because backwards may be left or right, depending on the language. The arrow keys should be bound to LFUN_WORD_LEFT or LFUN_WORD_RIGHT actions, which in turn may employ this one.
Syntax: word-backward

LFUN_WORD_BACKWARD_SELECT.
Action: Moves the cursor to the logically previous beginning of a word, adding the logically traversed text to the selection.
Notion: See also LFUN_WORD_BACKWARD.
Syntax: word-backward-select

LFUN_WORD_DELETE_BACKWARD.
Action: Deletes characters to the beginning of the word (usually the "C+BackSpace" key).
Syntax: word-delete-backward

LFUN_WORD_DELETE_FORWARD.
Action: Deletes characters to the end of the word (usually the "C+Delete" key).
Syntax: word-delete-forward
**LFUN_WORD_FIND_FORWARD.**

*Action:* Search for a given string in forward direction.

*Notion:* Case sensitive, match words. If no argument given, last search repeated.

*Syntax:* word-find-forward ["<STRING>"]

*Origin:* Etienne, 16 Feb 1998

**LFUN_WORD_FIND_BACKWARD.**

*Action:* Search for a given string in backward direction.

*Notion:* Case sensitive, match words. If no argument given, last search repeated.

*Syntax:* word-find-backward ["<STRING>"]

*Origin:* Etienne, 20 Feb 1998

**LFUN_WORD_FIND.**

*Action:* Search for next occurrence of a string.

*Syntax:* word-find [<DATA>]

*Params:* <DATA>: data encoded from Find dialog (see lyx::find2string()). If no parameter is given, search with last find-dialog data is used for search (i.e. find-next).

*Origin:* Andre, Jan 7 2004

**LFUN_WORD_REPLACE.**

*Action:* Replace a string in the document.

*Syntax:* word-replace [<DATA>]

*Params:* <DATA>: data is of the form "<search> <replace> <casesensitive> <matchword> <all> <forward>"

*Origin:* Andre, Jan 7 2004

**LFUN_WORD_FORWARD.**

*Action:* Moves the cursor to the logically next beginning of a word.

*Notion:* This is not the action which should be bound to the arrow keys, because forward may be left or right, depending on the language. The arrow keys should be bound to LFUN_WORD_LEFT or LFUN_WORD_RIGHT actions, which in turn may employ this one.

*Syntax:* word-forward

**LFUN_WORD_FORWARD_SELECT.**

*Action:* Moves the cursor to the logically next beginning of a word, adding the logically traversed text to the selection.

*Notion:* See also LFUN_WORD_FORWARD.

*Syntax:* word-forward-select

**LFUN_WORD_LEFT.**

*Action:* Moves the cursor to the next beginning of a word "on the left".

*Notion:* This is the action which should be taken when the (e.g., ctrl-) "left" key is pressed. Generally, it moves the cursor to the next beginning of a word on the left. However, in Bidi text this become slightly more complicated, and there are different modes of cursor movement. In "visual mode", this moves left, plain and simple. In "logical mode", movement is logically forward in RTL paragraphs, and logically backwards in LTR paragraphs.

*Syntax:* word-left

*Origin:* dov, 28 Oct 2007

**LFUN_WORD_LEFT_SELECT.**

*Action:* Moves the cursor to the next beginning of a word "on the left", adding *logically* traversed text to the selection.

*Notion:* See also LFUN_WORD_LEFT for exact details of the movement.

*Syntax:* word-left-select

*Origin:* dov, 28 Oct 2007
LFUN_WORD_RIGHT.

Action: Moves the cursor to the next beginning of a word "on the right".

Notion: This is the action which should be taken when the (e.g., ctrl-) "right" key is pressed. Generally, it moves the cursor to the next beginning of a word on the right. However, in Bidi text this become slightly more complicated, and there are different modes of cursor movement. In "visual mode", this moves right, plain and simple. In "logical mode", movement is logically forward in LTR paragraphs, and logically backwards in RTL paragraphs.

Syntax: word-right

Origin: dov, 28 Oct 2007

LFUN_WORD_RIGHT_SELECT.

Action: Moves the cursor to the next beginning of a word "on the right", adding *logically* traversed text to the selection.

Notion: See also LFUN_WORD_RIGHT for exact details of the movement.

Syntax: word-right-select

Origin: dov, 28 Oct 2007

LFUN_WORD_SELECT.

Action: Puts the word where the cursor stands into the selection.

Syntax: word-select

Origin: Andre, 11 Sep 2002

LFUN_WORD_CAPITALIZE.

Action: Capitalizes the words in the selection (i.e. the first letters) or the letter on the cursor position.

Syntax: word-capitalize

LFUN_WORD_UPCASE.

Action: Change the words in the selection or from the cursor position to the end of word to the upper case.

Syntax: word-upcase

LFUN_WORD_LOWCASE.

Action: Change the words in the selection or from the cursor position to the end of word to the lower case.

Syntax: word-lowcase

LFUN_THESAURUS_ENTRY.

Action: Look up thesaurus entries with respect to the word under the cursor.

Syntax: thesaurus-entry

Origin: Levon, 20 Jul 2001

LFUN_BUFFER_BEGIN.

Action: Move the cursor to the beginning of the document.

Syntax: buffer-begin

LFUN_BUFFER_BEGIN_SELECT.

Action: Move the cursor to the beginning of the document adding the traversed text to the selection.

Syntax: buffer-begin-select

LFUN_BUFFER_END.

Action: Move the cursor to the end of the document.

Syntax: buffer-end

LFUN_BUFFER_END_SELECT.

Action: Move the cursor to the end of the document adding the traversed text to the selection.

Syntax: buffer-end-select
LFUN_INSET_BEGIN.
Action: Move the cursor to the beginning of the current inset if it is not already there, or at the
beginning of the enclosing inset otherwise
Syntax: inset-begin
Origin: JMarc, 2009/03/16

LFUN_INSET_BEGIN_SELECT.
Action: Move the cursor to the beginning of the current inset if it is not already there, or at the
beginning of the enclosing inset otherwise (adding the traversed text to the selection).
Syntax: inset-begin-select
Origin: JMarc, 2009/03/16

LFUN_INSET_END.
Action: Move the cursor to the end of the current inset if it is not already there, or at the end of
the enclosing inset otherwise
Syntax: inset-end
Origin: JMarc, 2009/03/16

LFUN_INSET_END_SELECT.
Action: Move the cursor to the end of the current inset if it is not already there, or at the end of
the enclosing inset otherwise (adding the traversed text to the selection).
Syntax: inset-end-select
Origin: JMarc, 2009/03/16

LFUN_LINE_BEGIN.
Action: Move the cursor to the begining of the (screen) line.
Syntax: line-begin

LFUN_LINE_BEGIN_SELECT.
Action: Move the cursor to the beginning of the (screen) line adding the traversed text to the
selection.
Syntax: line-begin-select

LFUN_LINE_END.
Action: Move the cursor to the end of the (screen) line.
Syntax: line-end

LFUN_LINE_END_SELECT.
Action: Move the cursor to the end of the (screen) line adding the traversed text to the selection.
Syntax: line-end-select

LFUN_LINE_DELETE.
Action: Deletes the letters to the end of the (screen) line or deletes the selection.
Syntax: line-delete-forward

LFUN_COPY.
Action: Copies to the clipboard the last edit.
Syntax: copy

LFUN_CUT.
Action: Cuts to the clipboard.
Syntax: cut

LFUN_PASTE.
Action: Pastes material from the active clipboard.
Syntax: paste [<TYPE>]
Params: <TYPE>: emf|pdf|png|jpeg|linkback|wmf
LFUN_CLIPBOARD_PASTE.
  Action: Pastes text from the active clipboard.
  Syntax: clipboard-paste [<ARG>]
  Params: <ARG>: "paragraph" will cause pasting as one paragraph, i.e. "Join lines".
  Origin: baum, 10 Jul 2006

LFUN_PRIMARY_SELECTION_PASTE.
  Action: Pastes the currently text selected text.
  Notion: Primary selection mechanism is linux-only thing.
  Syntax: primary-selection-paste [<ARG>]
  Params: <ARG>: "paragraph" will cause pasting as one paragraph, i.e. "Join lines".

LFUN_SELECTION_PASTE.
  Action: Pastes the text in permanent selection.
  Syntax: selection-paste

LFUN_UNDO.
  Action: Undoes the last edit.
  Syntax: undo

LFUN_REDO.
  Action: Redoes the last thing undone.
  Syntax: redo

LFUN_REPEAT.
  Action: Repeat the given command.
  Syntax: repeat <COUNT> <LFUN-COMMAND>

LFUN_CHARS_TRANSPOSE.
  Action: Transposes the character at the cursor with the one before it.
  Syntax: chars-transpose
  Origin: Lgb, 25 Apr 2001

LFUN_DEPTH_DECREMENT.
  Action: Decrease the nesting depth of the (selected) paragraph(s) inside lists.
  Syntax: depth-decrement

LFUN_DEPTH_INCREMENT.
  Action: Increase the nesting depth of the (selected) paragraph(s) inside lists.
  Syntax: depth-increment

LFUN_FONT_BOLD.
  Action: Toggles the bold font (selection-wise) using mathbf in math.
  Syntax: font-bold

LFUN_FONT_BOLDSYMBOL.
  Action: Toggles the bold font (selection-wise) using boldsymbol in math.
  Syntax: font-boldsymbol

LFUN_FONT_TYPETRITER.
  Action: Toggles the typewriter family font (selection-wise).
  Syntax: font-typewriter

LFUN_FONT_UNDERLINE.
  Action: Toggles underline in the font (selection-wise).
  Syntax: font-underline

LFUN_FONT_EMPH.
  Action: Toggles the emphasis font style (selection-wise).
  Syntax: font-emph
**LFUN_FONT_NOUN.**

**Action:** Toggles Noun text style font (selection-wise).
**Syntax:** font-noun

**LFUN_FONT_ROMAN.**

**Action:** Toggles Roman family font (selection-wise).
**Syntax:** font-roman

**LFUN_FONT_SANS.**

**Action:** Toggles Sans Serif family font (selection-wise).
**Syntax:** font-sans

**LFUN_FONT_FRAK.**

**Action:** Toggles Fraktur family font (math-mode, selection-wise).
**Syntax:** font-frak
**Origin:** vermeer, 10 Jan 2002

**LFUN_FONT_ITAL.**

**Action:** Toggles Italics font shape (math-mode, selection-wise).
**Syntax:** font-ital
**Origin:** vermeer, 10 Jan 2002

**LFUN_FONT_DEFAULT.**

**Action:** Reverts the settings of the font to the default values (selection-wise).
**Syntax:** font-default

**LFUN_FONT_SIZE.**

**Action:** Sets font size according to lyx format string.
**Syntax:** font-size <SIZE>
**Params:**
- `<SIZE>`: tiny|scriptsize|footnotesize|small|normal|large|larger|largest|huge|giant|increase|decrease|default

**LFUN_TEXTSTYLE_APPLY.**

**Action:** Toggle user-defined (=last-time used) text style.
**Notion:** This style is set via LFUN_TEXTSTYLE_UPDATE, which is automatically triggered when using Text Style dialog.
**Syntax:** textstyle-apply
**Origin:** leeming, 12 Mar 2003

**LFUN_TEXTSTYLE_UPDATE.**

**Action:** Apply text style and update the settings to be used by LFUN_TEXTSTYLE_APPLY.
**Syntax:** textstyle-update <FONT_INFO>
**Params:**
- `<FONT_INFO>`: specifies font attributes, e.g. family, series, shape, size, emph, noun, underbar, number, color, language, toggleall.
  - Use `lyx -dbg action` for exact syntax of text-style dialog parameters.
**Origin:** leeming, 12 Mar 2003

**LFUN_SCREEN_FONT_UPDATE.**

**Action:** Update fonts and its metrics.
**Notion:** Automatically called after zoom, dpi, font names, or norm change.
**Syntax:** screen-font-update
**Origin:** ARRae, 13 Aug 2000

**LFUN_FONT_STATE.**

**Action:** Returns the info about the current font.
**Syntax:** font-state
LFUN_CITATION_INSERT.

Action: Inserts citation from loaded citation database.
Syntax: citation-insert |
Params: |
<KEY>: Citation (shortcut listed in available citations).
<TEXT_BEFORE>: text which should appear before citation.
Origin: AAS, 97-02-23

LFUN_BIBTEX_DATABASE_ADD.

Action: Adds database, which will be used for bibtex citations.
Notion: Databases are added to the first Bib\TeX\ inset (Inset->List/TOC->Bib\TeX\ bibliography) found from the cursor position.
Syntax: bibtex-database-add <DATABASE-NAME>
Origin: Ale, 30 May 1997

LFUN_BIBTEX_DATABASE_DEL.

Action: Adds database, which will be used for bibtex citations.
Notion: Databases are deleted from the first Bib\TeX\ inset (Inset->List/TOC->Bib\TeX\ bibliography) found from the cursor position.
Syntax: bibtex-database-del <DATABASE-NAME>
Origin: Ale, 30 May 1997

LFUN_LAYOUT.

Action: Sets the layout (that is, environment) for the current paragraph.
Syntax: layout <LAYOUT>
Params: <LAYOUT>: the layout to use

LFUN_LAYOUT_PARAGRAPH.

Action: Launches the paragraph settings dialog.
Syntax: layout-paragraph

LFUN_LAYOUT_TABULAR.

Action: Launches the tabular settings dialog.
Syntax: layout-tabular
Origin: Jug, 31 Jul 2000

LFUN_DROP_LAYOUTS_CHOICE.

Action: Displays list of layout choices.
Notion: In the current (as of 2007) Qt4 frontend, this LFUN opens the dropbox allowing for choice of layout.
Syntax: drop-layouts-choice

LFUN_LAYOUT_MODULES_CLEAR.

Action: Clears the module list.
Notion: Clears the list of included modules for the current buffer.
Syntax: layout-modules-clear
Origin: rgh, 25 August 2007

LFUN_LAYOUT_MODULE_ADD.

Action: Adds a module.
Notion: Adds a module to the list of included modules for the current buffer.
Syntax: layout-module-add <MODULE>
Params: <MODULE>: the module to be added
Origin: rgh, 25 August 2007

LFUN_LAYOUT_RELOAD.

Action: Reloading layout information.
Notion: Reloading all layout information for the current buffer from disk, thus recognizing any changes that have been made to layout files on the fly. This is intended to be used only by layout developers and should not be used when one is trying to do actual work.
Syntax: layout-reload
Origin: rgh, 3 September 2007
LFUN_TEXTCLASS_APPLY.

**Action:** Sets the text class for the current buffer.

**Syntax:**

```
ertextclass-apply <TEXTCLASS>
```

**Params:** `<TEXTCLASS>`: the textclass to set. Note that this must be the filename, minus the ".layout" extension.

LFUN_TEXTCLASS_LOAD.

**Action:** Loads information for a textclass from disk.

**Syntax:**

```
textclass-load <TEXTCLASS>
```

**Params:** `<TEXTCLASS>`: the textclass to load. Note that this must be the filename, minus the ".layout" extension.

LFUN_MARK_OFF.

**Action:** Disable selecting of text-region.

**Syntax:** `mark-off`

LFUN_MARK_ON.

**Action:** Enable selecting of text-region.

**Notion:** After enabling you can simply move arrow keys to get selected region.

**Syntax:** `mark-on`

LFUN_MARK_TOGGLE.

**Action:** Toggle between LFUN_MARK_ON and LFUN_MARK_OFF.

**Syntax:** `mark-toggle`

**Origin:** Andre, May 5 2006

LFUN_MATH_DELIM.

**Action:** Inserts math delimiters (e.g. parentheses, brackets) enclosing expression.

**Syntax:**

```
math-delim [\LEFT\] [\RIGHT\]
```

**Params:** `<LEFT/RIGHT>`: Delimiters to be used. Each delimiter can be specified by either a \LaTeX name or a valid character. ( is the default letter.

**Sample:** `math-delim \{ rangle`

**Origin:** Alejandro, 18 Jun 1996

LFUN_MATH_BIGDELIM.

**Action:** Inserts math fixed size delimiters (e.g. parentheses, brackets) enclosing expression.

**Syntax:**

```
math-bigdelim <LSIZE> <LDELIM> <RSIZE> <RDELIM>
```

**Params:** `<L/RSIZE>`: bigl/r|Bigl/r|biggl/r|Biggl/r

`<L/RDELIM>`: \LaTeX code for delimiter. See Delimiter dialog for delimiters to be used.

**Sample:** `math-bigdelim "Bigl" "Downarrow" "Bigg" n}`

**Origin:** Enrico & Georg, 7 May 2006

LFUN_MATH_DISPLAY.

**Action:** Creates a new displayed equation in text mode. Toggles inlined/display formula in math mode.

**Syntax:**

```
math-display [<ARG>]
```

**Params:** `<ARG>`: this argument will be passed to LFUN_MATH_INSERT when creating new equation from the text mode.

**Origin:** Alejandro, 18 Jun 1996

LFUN_MATH_INSERT.

**Action:** Inserts math objects and symbols.

**Syntax:**

```
math-insert <ARG>
```

**Params:** `<ARG>`: Symbol or \LaTeX code to be inserted.

**Notion:** When `<ARG>` is a _single_ _math inset with more than one cell (such as "x\_y\_z" or "\frac{x}{y}"), the content of cell(0) is replaced by the current selection (only works if the selection is in mathed). As an example, if "abc" is selected in mathed, "math-insert \frac{x}{y}" replaces "abc" with \frac{abc}{y}, and "math-insert _x\_y\_z" replaces "abc" with \abc_. If nothing is selected (or the selection is not in mathed), math-insert works as expected.
LFUN_MATH_SUBSCRIPT.
Action: Enters subscript expression in math expression.
Syntax: math-subscript
Origin: vermeer, 12 Dec 2001

LFUN_MATH_SUPERSCRIPT.
Action: Enters subscript expression in math expression.
Syntax: math-superscript
Origin: vermeer, 12 Dec 2001

LFUN_MATH_LIMITS.
Action: Toggles the position of the limits from above/below to the right side—an vice versa in integral symbol, a limit, a summation, etc.
Notion: Put the cursor before the symbol with the limits and then invoke math-limits.
Syntax: math-limits [<STATE>]
Params: <STATE>: limits|nolimits

LFUN_MATH_MACRO.
Action: Inserts a math macro definition at the cursor position in the text.
Syntax: math-macro <NAME> [<NARGS>] [def]
Params: <NAME>: The name of the macro, e.g. "mymacro". <NARGS>: The number of parameters of the macro. Default is 0. "def": Has no effect anymore, just for compatibility with former \LaTeX versions.
Origin: ale, 10 May 1997; sts, 21 Dec 2007

LFUN_MATH_MUTATE.
Action: Mutates the type of math inset to the newly selected one.
Syntax: math-mutate <TYPE>
Params: <TYPE>: none|simple|equation|eqnarray|align|alignat|xxalignat| multiline|gather|flalign
Origin: Andre, 23 May 2001

LFUN_MATH_SPACE.
Action: Inserts space into math expression.
Notion: Use spacebar after entering this space to change type of space.
Syntax: math-space [<TYPE>] [<LEN>]
Params: <TYPE>: negative spaces: \text{-}negthinspace|negmedspace|negthickspace
positive spaces: \text{,}thinspace|\text{:}medspace|\text{;}thickspace|\text{\enskip}enskip|\text{\quad}quad|\text{\qquad}qquad
custom space: \hspace
"," used by default. Note that ! is equivalent to negthinspace, , = thinspace, : = medspace, and ; = thickspace. <LEN>: length for custom spaces (\hspace)

LFUN_MATH_MATRIX.
Action: Inserts a matrix.
Syntax: math-matrix <COLS> <ROWS> [<ALIGN>]
Params: <ALIGN>: Alignment is a word composed of the vertical alignment (b, c or t) (i.e. 1 char) and the horizontal alignments (l, c or r) (i.e. <COL> chars).
Sample: math-matrix 3 3 bccc

LFUN_MATH_MODE.
Action: In text mode enters math mode (i.e. puts math insets on the current cursor position), in math mode enters text mode inside math expression.
Notion: If there is some selected text, it puts the text inside created math box.
Syntax: math-mode [<ARG>]
Params: <ARG>: eventual argument (\LaTeX code) is passed to LFUN_MATH_INSERT.
Origin: Alejandro, 4 Jun 1996
**LFUN_MATH_NUMBER_LINE_TOGGLE.**

**Action:** Toggles numbering of the current formula line.

**Notion:** Must be in display formula mode.

**Syntax:** math-number-line-toggle

**Origin:** Alejandro, 18 Jun 1996

**LFUN_MATH_NUMBER_TOGGLE.**

**Action:** Toggles numbering/labelling of the current formula.

**Notion:** Must be in display formula mode.

**Syntax:** math-number-toggle

**Origin:** Alejandro, 4 Jun 1996

**LFUN_MATH_EXTERN.**

**Action:** Calls external program and passes the current expression/equation as an argument for the calculation in the format appropriate to the given language.

**Notion:** Selection can be used to determine the input for the external program.

**Syntax:** math-extern <LANG> [COMMAND]

**Params:**
- `<LANG>`: octave|maxima|maple|mathematica|script
  - where "script" stands for the external script (normalized expression will be passed)

**Origin:** Andre, 24 Apr 2001

**Sample:** math-extern maple simplify

**LFUN_MATH_SIZE.**

**Action:** Changes arbitrarily the size used by math fonts inside a context.

**Notion:** Provides an interface to the \LaTeX\ math mode font size commands.

**Syntax:** math-size <STYLE>

**Params:**
- `<STYLE>`: displaystyle|textstyle|scriptstyle|scriptscriptstyle

**Origin:** Alejandro, 15 Aug 1996; ps, 14 Jun 2008

**LFUN_MATH_FONT_STYLE.**

**Action:** Changes the text style used in math.

**Syntax:** math-font-style <STYLE>

**Params:**
- `<STYLE>`: mathnormal|mathcal|mathfrak|mathrm|mathsf|mathbf |textnormal|textit|textbf |textsl|textrm|textsf|texttt|textup

**Origin:** vfr, 9 jan 2009

**LFUN_MATH_MACRO_UNFOLD.**

**Action:** Unfold a Math Macro.

**Notion:** Unfold the Math Macro the cursor is in, i.e. display it as foo.

**Syntax:** math-macro-unfold

**Origin:** sts, 06 January 2008

**LFUN_MATH_MACRO_FOLD.**

**Action:** Fold a Math Macro.

**Notion:** Fold the Math Macro the cursor is in if it was unfolded, i.e. displayed as foo before.

**Syntax:** math-macro-fold

**Origin:** sts, 06 January 2008

**LFUN_MATH_MACRO_ADD_PARAM.**

**Action:** Add a parameter.

**Notion:** Add a parameter to a Math Macro.

**Params:** `<NUM>`: The number of the parameter behind which the new one will be added (1 for the first, i.e. use 0 for add a parameter at the left), defaults to the last one.

**Syntax:** math-macro-add-param <NUM>

**Origin:** sts, 06 January 2008
**LFUN_MATH_MACRO_REMOVE_PARAM.**

**Action:** Remove the last parameter.

**Notion:** Remove the last parameter of a Math Macro and remove its value in all instances of the macro in the buffer.

**Params:** <NUM>: The number of the parameter to be deleted (1 for the first), defaults to the last one.

**Syntax:** math-macro-remove-param <NUM>

**Origin:** sts, 06 January 2008

---

**LFUN_MATH_MACRO_APPEND_GREEDY_PARAM.**

**Action:** Append a greedy parameter.

**Notion:** Append a greedy parameter to a Math Macro which eats the following mathed cell in every instance of the macro in the buffer.

**Syntax:** math-macro-append-greedy-param

**Origin:** sts, 06 January 2008

---

**LFUN_MATH_MACRO_REMOVE_GREEDY_PARAM.**

**Action:** Remove a greedy parameter.

**Notion:** Remove a greedy parameter of a Math Macro and spit out the values of it in every instance of the macro in the buffer. If it is an optional parameter the [value] format is used.

**Syntax:** math-macro-remove-greedy-param

**Origin:** sts, 06 January 2008

---

**LFUN_MATH_MACRO_MAKE_OPTIONAL.**

**Action:** Make a parameter optional.

**Notion:** Turn the first non-optional parameter of a Math Macro into an optional parameter with a default value.

**Syntax:** math-macro-make-optional

**Origin:** sts, 06 January 2008

---

**LFUN_MATH_MACRO_MAKE_NONOPTIONAL.**

**Action:** Make a parameter non-optional.

**Notion:** Turn the last optional parameter of a Math Macro into a non-optional parameter. The default value is remembered to be reused later if the user changes his mind.

**Syntax:** math-macro-make-nonoptional

**Origin:** sts, 06 January 2008

---

**LFUN_MATH_MACRO_ADD_OPTIONAL_PARAM.**

**Action:** Add an optional parameter.

**Notion:** Insert an optional parameter just behind the already existing optional parameters.

**Syntax:** math-macro-add-optional-param

**Origin:** sts, 06 January 2008

---

**LFUN_MATH_MACRO_REMOVE_OPTIONAL_PARAM.**

**Action:** Remove the last optional parameter.

**Notion:** Remove the last optional parameter of a Math Macro and remove it in all the instances of the macro in the buffer.

**Syntax:** math-macro-remove-optional-param

**Origin:** sts, 06 January 2008

---

**LFUN_MATH_MACRO_ADD_GREEDY_OPTIONAL_PARAM.**

**Action:** Add a greedy optional parameter.

**Notion:** Add a greedy optional parameter which eats the value from the following cells in mathed which are in the [value] format.

**Syntax:** math-macro-add-greedy-optional-param

**Origin:** sts, 06 January 2008
**LFUN_IN_MATHMACROTEMPLATE.**

**Action:** Only active in Math Macro definition.

**Notion:** Dummy function which is only active in a Math Macro definition. It’s used to toggle the Math Macro toolbar if the cursor moves into a Math Macro definition.

**Syntax:** `in-mathmacrotemplate`

**Origin:** sts, 06 January 2008

**LFUN_PARAGRAPH_MOVE_DOWN.**

**Action:** Moves the current paragraph downwards in the document.

**Syntax:** `paragraph-move-down`

**Origin:** Edwin, 8 Apr 2006

**LFUN_PARAGRAPH_MOVE_UP.**

**Action:** Moves the current paragraph upwards in the document.

**Syntax:** `paragraph-move-up`

**Origin:** Edwin, 8 Apr 2006

**LFUN_PARAGRAPH_UP.**

**Action:** Move the cursor to the next paragraph (or begining of the current one) in upward direction.

**Syntax:** `paragraph-up`

**Origin:** Asger, 1 Oct 1996

**LFUN_PARAGRAPH_UP_SELECT.**

**Action:** Move the cursor and select the text to the next paragraph (or begining of the current one) in upward direction.

**Syntax:** `paragraph-up-select`

**Origin:** Asger, 1 Oct 1996

**LFUN_PARAGRAPH_DOWN.**

**Action:** Move the cursor to the next paragraph (or begining of the current one) in downward direction.

**Syntax:** `paragraph-down`

**Origin:** Asger, 1 Oct 1996

**LFUN_PARAGRAPH_DOWN_SELECT.**

**Action:** Move the cursor and select the text to the next paragraph (or begining of the current one) in downward direction.

**Syntax:** `paragraph-down-select`

**Origin:** Asger, 1 Oct 1996

**LFUN_PARAGRAPH_GOTO.**

**Action:** Jump to a paragraph given by its id number and optionally the desired position within the paragraph.

**Notion:** Note that id number of paragraph is not the sequential number of paragraph seen on the screen. Moreover the id is unique for all opened buffers (documents).

**Syntax:** `paragraph-goto <PAR_ID_NUMBER> <POSITION_IN_PAR>`

**Params:**
- `<PAR_ID_NUMBER>`: paragraph id
- `<POSITION_IN_PAR>`: desired position within the paragraph

**Origin:** Dekel, 26 Aug 2000

**LFUN_BREAK_PARAGRAPH.**

**Action:** Breaks the current paragraph at the current location.

**Syntax:** `break-paragraph`

**LFUN_BREAK_PARAGRAPH.**

**Action:** Breaks the current paragraph at the current location.

**Notion:** Removes the selection.

**Syntax:** `break-paragraph [<LAYOUT>]`

**Params:** `<LAYOUT>`: "inverse" - decreases depth by one (or change layout to default layout) when the cursor is at the end of the line.
**LFUN_PARAGRAPH_PARAMS.**

**Action:** Change paragraph settings.

**Notion:** Modifies the current paragraph, or currently selected paragraphs. This function only modifies, and does not override, existing settings. Note that the "leftindent" indent setting is deprecated.

**Syntax:** paragraph-params [ <INDENT> ] [ <SPACING> ] [ <ALIGN> ] [ <OTHERS> ]

**Params:**
- `<INDENT>`: noindent | indent | indent-toggle | leftindent LENGTH
- `<SPACING>`: paragraph_spacing default | single | onehalf | double | other
- `<ALIGN>`: align block | left | right | center | default
- `<OTHERS>`: labelwidth | string WIDTH | start_of_appendix

**Origin:** rgh, Aug 15 2007

**LFUN_PARAGRAPH_PARAMS_APPLY.**

**Action:** Change paragraph settings.

**Notion:** Overwrite all unspecified settings to the default ones. Use paragraph-params Ifun if you don’t want to overwrite others settings.

**Syntax:** paragraph-params-apply <INDENT> <SPACING> <ALIGN> <OTHERS>

**Params:** For parameters see LFUN_PARAGRAPH_PARAMS

**Origin:** leeming, 30 Mar 2004

**LFUN_PARAGRAPH_UPDATE.**

**Action:** Updates the values inside the paragraph dialog from the paragraph.

**Notion:** This is internal LFUN, not to be used by users. Called internally by LFUN_DIALOG_UPDATE.

**Origin:** leeming, 13 Mar 2003

**LFUN_OUTLINE_UP.**

**Action:** Move the current group in the upward direction in the structure of the document.

**Notion:** The "group" can be Part/Chapter/Section/etc. It moves the whole substructure of the group.

**Syntax:** outline-up

**Origin:** Vermeer, 23 Mar 2006

**LFUN_OUTLINE_DOWN.**

**Action:** Move the current group in the downward direction in the structure of the document.

**Notion:** The "group" can be Part/Chapter/Section/etc. It moves the whole substructure of the group.

**Syntax:** outline-down

**Origin:** Vermeer, 23 Mar 2006

**LFUN_OUTLINE_IN.**

**Action:** Moves the current group in the downward direction in the hierarchy of the document structure.

**Notion:** Part -> Chapter -> Section -> etc.

**Syntax:** outline-in

**Origin:** Vermeer, 23 Mar 2006

**LFUN_OUTLINE_OUT.**

**Action:** Moves the current group in the upward direction in the hierarchy of the document structure.

**Notion:** Part <- Chapter <- Section <- etc.

**Syntax:** outline-out

**Origin:** Vermeer, 23 Mar 2006

**LFUN_INSET_EDIT.**

**Action:** Edit the inset at cursor with an external application, if one is attributed.

**Syntax:** inset-edit [ <INSET_PARAMS> ]

**Params:** <INSET_PARAMS>: Parameters for the inset.

Currently only the filename will be considered.

**Origin:** JSpitzm, 27 Apr 2006
LFUN_TABULAR_INSERT.

Action: Inserts table into the document.
Syntax: tabular-insert \[<ROWS> \(<COLUMN>\]s\]
Params: In case no arguments are given show insert dialog.
Origin: Jug, 12 Apr 2000

LFUN_TABULAR_FEATURE.

Action: Sets various features to the table/cell on the current cursor position.
Notion: Various math-environment features are handled here as well, e.g. add-vline-left/right for the Grid/Array environment
Syntax: tabular-feature <FEATURE> \[<ARG>\]
Params: <FEATURE>: append-row|append-column|delete-row|delete-column|copy-row|copy-column|
toggle-line-top|toggle-line-bottom|toggle-line-left|toggle-line-right|align-left|align-right|align-center|align-block|
valign-top|valign-bottom|valign-middle|m-align-left|m-align-right|m-align-center|m-align-top|
m-valign-bottom|m-valign-middle|multicolumn|set-all-lines|unset-all-lines|set-longtabular|unset-longtabular|
set-mpwidth|set-mpwidth|set-rotate-tabular|unset-rotate-tabular|set-rotate-cell|unset-rotate-cell|
toggle-rotate-cell|set-usebox|set-lthead|unset-lthead|set-ltfirsthead|unset-ltfirsthead|set-ltlastfoot|unset-ltlastfoot|
special-column|set-special-column|set-booktabs|unset-booktabs|set-top-space|set-bottom-space|set-interline-space|set-border-lines
<ARG>: additional argument for some commands, use debug mode to explore its values.
Origin: Jug, 28 Jul 2000

LFUN_CELL_BACKWARD.

Action: Moves the cursor to the previous cell inside the table.
Syntax: cell-backward
Origin: Jug, 22 May 2000

LFUN_CELL_FORWARD.

Action: Moves the cursor to the next cell inside the table.
Syntax: cell-forward

LFUN_CELL_SPLIT.

Action: Splits cell and shifts right part to the next cell (inside the math grid).
Syntax: cell-split
Origin: Ale, 15 May 1997

LFUN_VC_REGISTER.

Action: Register the document as an file inside version control system (RCS, CVS).
Notion: File is registered inside cvs, svn or rcs repository according to the existence of cvs/svn/rcs entries in the document’s directory.
See LyX Additional Features Manual (Version Control Chapter) for additional information.
Syntax: vc-register
Origin: Lgb, 1 Jul 1997

LFUN_VC_CHECK_IN.

Action: Checks-in/commits the changes of the registered file to the repository.
Notion: In RCS case this also unlocks the file.
Syntax: vc-check-in
Origin: Lgb, 1 Jul 1997

LFUN_VC_CHECK_OUT.

Action: Checks-out the document for edit (and locks it for RCS).
Notion: This is implemented only for RCS and SVN, not CVS.
Syntax: vc-check-out
Origin: Lgb, 1 Jul 1997

LFUN_VC_REVERT.

Action: Reverts the document to the last check-in/commit in VCS.
Syntax: vc-revert
Origin: Lgb, 1 Jul 1997
LFUN_VC_UNDO_LAST.
Action: Undo last check-in.
Notion: This is currently implemented only for RCS.
Syntax: vc-check-out
Origin: Lgb, 1 Jul 1997

LFUN_VC_COMMAND.
Action: Executes external command. This command is intended to support additional VCS commands.
Syntax: vc-command <FLAG> <PATH> <COMMAND>
Params: <FLAG>: Flags for the command can be combined together.
   U - dummy - no flags
   D - Doc - need document loaded to proceed
   I - dirty - mark document dirty
   R - Reload - reload the document after command execution
   M - Message - ask for input string (commit message)
   <PATH>: path where to start. $$p will be replaced by the current document path.
   <COMMAND>: command to execute. $$i/$$/m will be replaced by the current document/path/message.
Sample: vc-command DR $$p "svn up"
Origin: sanda, 13 Jan 2009

LFUN_VC_LOCKING_TOGGLE.
Action: Toggles the locking property of the edited file.
Notion: This is currently implemented only for SVN.
Syntax: vc-locking-toggle
Origin: sanda, 25 Jun 2009

LFUN_CHANGES_TRACK.
Action: Toggles change tracking to on/off.
Syntax: changes-track
Origin: levon, 1 Oct 2002

LFUN_CHANGES_OUTPUT.
Action: Toggles showing of change tracking in typesetted output.
Syntax: changes-output
Origin: jspitzm, 21 Jan 2005

LFUN_CHANGE_NEXT.
Action: Moves the cursor to the position of the next change of the change tracking records.
Syntax: change-next
Origin: schnitt, 4 Oct 2006

LFUN_CHANGE_PREVIOUS.
Action: Moves the cursor to the position of the previous change of the change tracking records.
Syntax: change-previous
Origin: vfr, 4 Apr 2009

LFUN_CHANGES_MERGE.
Action: Open change tracking dialog for merging and moves the cursor to the position of the next change.
Syntax: changes-merge
Origin: Levon, 16 Oct 2002

LFUN_CHANGE_ACCEPT.
Action: Accepts tracked change inside the selection.
Syntax: change-accept
Origin: Levon, 16 Oct 2002
**LFUN_CHANGE_REJECT.**

Action: Rejects tracked change inside the selection.
Syntax: change-accept
Origin: Levon, 16 Oct 2002

**LFUN_ALL_CHANGES_ACCEPT.**

Action: Accepts all tracked changes in the document.
Syntax: all-changes-accept
Origin: Levon, 16 Oct 2002

**LFUN_ALL_CHANGES_REJECT.**

Action: Rejects all tracked changes in the document.
Notion: Reject does not work recursively; the user may have to repeat the operation.
Syntax: all-changes-reject
Origin: Levon, 16 Oct 2002

**LFUN_INSET_APPLY.**

Action: Apply data for an inset.
Notion: LFUN_INSET_APPLY is sent from the dialogs when the data should be applied. This is either changed to LFUN_INSET_MODIFY or LFUN_INSET_INSERT depending on the context where it is called.
Syntax: inset-apply <ARGS>
Params: See LFUN_INSET_INSERT.

**LFUN_INSET_DISSOLVE.**

Action: Dissolve the current inset into text.
Syntax: inset-dissolve [<INSET>]
Params: <INSET>: this can be used to make sure the right kind of inset is dissolved. For example "dissolve" entry in the charstyles sub-menu should only dissolve the charstyle inset, even if the cursor is inside several nested insets of different type.
For values see lyx::InsetLayout::lyxtype_.
Origin: JSpitz, 7 Aug 2006

**LFUN_INSET_INSERT.**

Action: Insert new inset (type given by the parameters).
Syntax: inset-insert <INSET> <ARGS>
Params: <INSET>: <bibitem|bibtex|cite|ert|listings|external|graphics|hyperlink|include|index|label|nomencl|vspace|ref|to}
<ARGS>: depends on the given inset. Use "lyx - dbg action" to explore.
Sample: inset-insert ref LatexCommand <Format> reference "<lab el name>" end_inset
where <lab el name> is the name of the referenced label and <Format> is one of the following:
  ref - <reference>
  eqref - (<reference>)
  pageref - <page>
  vpageref - on <page>
  vref - <reference> on <page>
  prettyref - Formatted reference

**LFUN_INSET_MODIFY.**

Action: Modify existing inset.
Notion: Used for label, floats, listings, box, branch, external, wrap bibtex, ert, command, graphics, note, space, vspace, tabular, bibitem, include, ref insets.
Syntax: inset-modify <INSET> <ARGS>
Params: See LFUN_INSET_INSERT for further details.

**LFUN_NEXT_INSET_MODIFY.**

Action: Modify the inset at cursor position, if there is one.
Notion: Used for label, floats, listings, box, branch, external, wrap bibtex, ert, command, graphics, note, space, vspace, tabular, bibitem, include, ref insets.
Syntax: next-inset-modify <INSET> <ARGS>
Syntax: next-inset-modify chancetype <TYPE>
Params: See LFUN_INSET_INSERT for further details.
Origin: JSpitzm, 23 Mar 2008

**LFUN_INSET_DIALOG_UPDATE.**

**Action:** Updates the values inside the dialog from the inset.

**Notion:** This is internal LFUN, not to be used by users. Called internally by LFUN_DIALOG_UPDATE

**Params:** <DIALOG-NAME>

**Origin:** leeming, 25 Feb 2003

**LFUN_INSET_SETTINGS.**

**Action:** Open the inset’s properties dialog.

**Notion:** Used for box, branch, ert, float, listings, note, tabular, wrap insets.

**Syntax:** inset-settings <INSET>

**Params:** <INSET>: <box|branch|ert|float|listings|note|tabular|wrap>

**Origin:** leeming, 30 Mar 2004

**LFUN_NEXT_INSET_TOGGLE.**

**Action:** Toggles the inset at cursor position. For collapsables, this means it will be (un-)collapsed, in case of other insets, the editing widget (dialog) will be entered. Also cf. LFUN_INSET_SETTINGS

**Notion:** Used for label, floats, listings, box, branch, external, wrap bibtex, ert, command, graphics, note, space, vspace, tabular, bibitem, include, ref insets.

**Syntax:** next-inset-toggle <ARG>

**Params:** <ARG>: these are passed as arguments to LFUN_INSET_TOGGLE.

**Origin:** lasgouttes, 19 Jul 2001

**LFUN_INSET_TOGGLE.**

**Action:** Toggles the collapsable inset we are currently in.

**Syntax:** inset-toggle [<ARG>]

**Params:** <ARG>: <open|close|toggle|assign>.
  - open/close/toggle are for collapsable insets. close can be currently used by LFUN_NEXT_INSET_TOGGLE.
  - toggle is used when no argument is given.
  - assign synchronize the branch-inset with activation status of the branch. Used for global toggling when changed activation.

**Origin:** lasgouttes, 19 Jul 2001

** LFUN_ALL_INSETS_TOGGLE.**

**Action:** Toggles (open/closes) all collapsable insets (of a given type) in the document.

**Notion:** Used for box, branch, ert, float, listings, note, tabular, wrap insets.

**Syntax:** all-insets-toggle <STATE> <INSET>

**Params:** <STATE>: <toggle|op en|close> default: toggle
  <INSET>: <box|branch|ert|float|listings|note|tabular|wrap> default: all insets

**Origin:** leeming, 30 Mar 2004

**LFUN_SET_GRAPHICS_GROUP.**

**Action:** Set the group for the graphics inset on the cursor position.

**Syntax:** set-graphics-group [<GROUP>]

**Params:** <GROUP>: Id for an existing group. In case the Id is an empty string, the graphics inset is removed from the current group.

**Origin:** sanda, 6 May 2008

**LFUN_FINISHED_FORWARD.**

**Action:** Moves the cursor out of the current slice, going forward.

**Notion:** Cursor movement within an inset may be different than cursor movement in the surrounding text. This action should be called automatically by the cursor movement within the inset, when movement within the inset has ceased (reached the end of the last paragraph, for example), in order to move correctly back into the surrounding text.

**LFUN_FINISHED_BACKWARD.**

**Action:** Moves the cursor out of the current slice, going backwards.

**Notion:** See also LFUN_FINISHED_FORWARD.

**LFUN_FINISHED_RIGHT.**

**Action:** Moves the cursor out of the current slice, going right.

**Notion:** See also LFUN_FINISHED_FORWARD
LFUN_FINISHED_LEFT.

Action: Moves the cursor out of the current slice, going left.
Notion: See also LFUN_FINISHED_FORWARD.

LFUN_LANGUAGE.

Action: Set language from the current cursor position.
Syntax: language <LANG>
Params: <LANG>: Requested language. Look in lib/languages for the list.
Origin: Dekel, 2 Mar 2000

LFUN_LABEL_GOTO.

Action: Goto a label.
Syntax: label-goto [<LABEL>]
Params: <LABEL>: Requested label. If no label is given and reference is on cursor position,
Bookmark 0 is saved and cursor moves to the position of referenced label.
Origin: Ale, 6 Aug 1997

LFUN_LABEL_INSERT.

Action: Inserts label to text or displayed formula.
Syntax: label-insert [<LABEL>]
Params: <LABEL>: Requested label. If no label is given dialog requesting name will be opened.

LFUN_REFERENCE_NEXT.

Action: Go to the next label or cross-reference.
Syntax: reference-next
Origin: Dekel, 14 Jan 2001

LFUN_BOOKMARK_GOTO.

Action: Moves the cursor to the numbered bookmark, opening the file if necessary. Note that
bookmarks are saved per-session, not per file.
Notion: Bookmark 0 has a special purpose. It is automatically set
1. to the paragraph you are currently editing
2. to the paragraph from where you are jumping to the last-edited position (jump-back feature)
3. when jumping from crossreference to the requested label by LFUN_LABEL_GOTO.
Syntax: bookmark-goto <NUMBER>
Params: <NUMBER>: the number of the bookmark to restore.
Origin: Dekel, 27 January 2001

LFUN_BOOKMARK_SAVE.

Action: Save a bookmark.
Notion: Saves a numbered bookmark to the sessions file. The number must be between 1 and 9,
inclusive. Note that bookmarks are saved per-session, not per file.
Syntax: bookmark-save <NUMBER>
Params: <NUMBER>: the number of the bookmark to save.
Origin: Dekel, 27 January 2001

LFUN_BOOKMARK_CLEAR.

Action: Clears the list of saved bookmarks.
Syntax: bookmark-clear
Origin: bpeng, 31 October 2006

LFUN_HELP_OPEN.

Action: Open the given help file according to the language setting.
Syntax: help-open <FILE>[/lyx]
Origin: Jug, 27 Jun 1999

LFUN_LYX_QUIT.

Action: Terminates the current LyX instance.
Notion: Terminates the current LyX instance, asking whether to save modified documents, etc.
Syntax: lyx-quit
LFUN_TOOLBAR_TOGGLE.

**Action:** Toggles visibility of a given toolbar between on/off/auto.

**Notion:** Skipping "auto" when allowauto is false.

**Syntax:** toolbar-toggle <NAME> [allowauto]

**Params:** <NAME>: standard|extra|table|math|mathmacrotemplate|minibuffer|review|view|update|math_panels|vcs

**Origin:** Edwin, 21 May 2007

LFUN_MENU_OPEN.

**Action:** Opens the menu given by its name.

**Syntax:** menu-open <NAME>

**Params:** <NAME>: menu name. See various .inc files in lib/ui for candidates.

LFUN_UI_TOGGLE.

**Action:** Various UI visibility-toggling actions.

**Syntax:** ui-toggle <statusbar|menubar|frame|fullscreen>

**Params:**
- statusbar: Toggle visibility of the statusbar.
- menubar: Toggle visibility of the menubar.
- scrollbar: Toggle visibility of the scrollbar.
- frame: Toggle visibility of the frames around editing window.
- fullscreen: Toggle fullscreen mode. This also covers calling the previous functions. However LFUN_TOOLBAR_TOGGLE for the custom tweaks of the toolbars should be used.

**Origin:** sanda, 9 Feb 2007

WINDOW_NEW.

**Action:** Creates new empty LyX window.

**Notion:** Already opened documents from the previous window can be found under View menu.

**Syntax:** window-new [<GEOMETRY>]

**Params:** <GEOMETRY>: pass the geometry of the window. This parameter is currently accepted only on Windows platform.

**Origin:** Abdel, 21 Oct 2006

LFUN_WINDOW_CLOSE.

**Action:** Closes the current LyX window.

**Syntax:** window-close

**Origin:** Abdel, 23 Oct 2006

LFUN_SPLIT_VIEW.

**Action:** Creates another split view of current buffer.

**Notion:** All split views act in the same way independently.

**Syntax:** split-view <vertical|horizontal>

**Params:**
- horizontal: The work areas are laid out side by side.
- vertical: The work areas laid out vertically.

**Origin:** Abdel, 20 Feb 2008

LFUN_CLOSE_TAB_GROUP.

**Action:** Close the current tab group.

**Notion:** This only closes the work areas, not the buffer themselves. The still opened buffers can be visualized in another tab group.

**Syntax:** close-tab-group

**Origin:** Abdel, 21 Feb 2008

LFUN_DIALOG_SHOW.

**Action:** Shows hidden dialog or create new one for a given function/inset settings etc.

**Syntax:** dialog-show <NAME> [<DATA>]

**Params:**
- <NAME>: aboutlyx|bibitem|bibtext|box|branch|changes|character|citation|document|errorlist|ert|external|file|findreplace|float|graphics|
- include|index|info|homenomenclature|label|log|mathdelimtext|mathmatrix
- note|paragraph|prefs|print|refsendto|space|spellchecker|symbols|
- tabular|tabularcreate|thesaurus|texinfo|toc|href|view-source|vspace
- wrap|listings|<SPECIAL>
LFUN_DIALOG_SHOW_NEW_INSET.
Action: Shows hidden dialog or create new one for a given inset settings etc.
Notion: Internally uses LFUN_DIALOG_SHOW with processed data for a given inset.
Syntax: dialog-show-new-inset <NAME> [<DATA>]
Params: See LFUN_DIALOG_SHOW.
Origin: leeming, 17 Jun 2003

LFUN_DIALOG_UPDATE.
Action: Updates the dialog values from the inset/paragraph/document.
Syntax: dialog-update <NAME>
Params: <NAME>: paragraph|prefs|<INSET>
    <INSET>: inset name
Origin: leeming, 25 Feb 2003

LFUN_DIALOG_HIDE.
Action: Hides showed dialog. Counterpart to LFUN_DIALOG_SHOW.
Syntax: dialog-hide <NAME>
Params: See LFUN_DIALOG_SHOW.
Origin: leeming, 25 Feb 2003

LFUN_DIALOG_TOGGLE.
Action: Toggles dialog between showed/hidden state.
Notion: Internally uses LFUN_DIALOG_SHOW, LFUN_DIALOG_HIDE.
Syntax: dialog-toggle <NAME> [<DATA>]
Params: See LFUN_DIALOG_SHOW.
Origin: JSpitzm, 30 Apr 2007

LFUN_DIALOG_DISCONNECT_INSET.
Action: Closes opened connection to opened inset.
Notion: Connection is used for apply functions.
Syntax: dialog-disconnect-inset <INSET-NAME>
Origin: leeming, 25 Feb 2003

LFUN_MOUSE_PRESS.
Action: This function is called when mouse button is pressed (inside workarea). Action depends on the context.
Notion: This is internal LFUN, not to be used by users.
Origin: Andre, 9 Aug 2002

LFUN_MOUSE_DOUBLE.
Action: This function is called when double click on mouse button is pressed (inside workarea). Action depends on the context.
Notion: This is internal LFUN, not to be used by users.
Origin: Andre, 9 Aug 2002

LFUN_MOUSE_TRIPLE.
Action: This function is called when triple click on mouse button is pressed (inside workarea). Action depends on the context.
Notion: This is internal LFUN, not to be used by users.
Origin: Andre, 9 Aug 2002

LFUN_MOUSE_MOTION.
Action: This function is called when mouse cursor is moving over the text. Action depends on the context.
Notion: This is internal LFUN, not to be used by users.
Origin: Andre, 9 Aug 2002

Origin: leeming, 17 Jun 2003

<DATA>: data, usually settings for the given dialog. Use debug mode for the details.

Origin: leeming, 25 Feb 2003

Origin: JSpitzm, 30 Apr 2007

Origin: Andre, 9 Aug 2002
LFUN_MOUSE_RELEASE.

**Action:** This function is called when mouse button is released (inside workarea). Action depends on the context.

**Notion:** This is internal LFUN, not to be used by users.

**Origin:** Andre, 9 Aug 2002

LFUN_KEYMAP_OFF.

**Action:** Turn off the loaded keyboard map.

**Syntax:** keymap-off

LFUN_KEYMAP_PRIMARY.

**Action:** Turn on the primary keyboard map.

**Notion:** Maps were widely used in past, when X-windows didn’t have nowadays keyboard support. They can be still used to maintain uniform keyboard layout across the various platforms.

The language is to be set in the Preferences dialog.

**Syntax:** keymap-primary

LFUN_KEYMAP_SECONDARY.

**Action:** Turn on the secondary keyboard map.

**Syntax:** keymap-secondary

LFUN_KEYMAP_TOGGLE.

**Action:** Toggles keyboard maps (first/second/off).

**Syntax:** keymap-toggle

**Origin:** leeming, 30 Mar 2004

LFUN_SERVER_GET_LAYOUT.

**Action:** Returns the current layout (that is environment) name on the cursor position.

**Syntax:** server-get-layout

LFUN_SERVER_GET_FILENAME.

**Action:** Returns path and file name of the currently edited document.

**Syntax:** server-get-filename

LFUN_SERVER_GOTO_FILE_ROW.

**Action:** Sets the cursor position based on the row number of generated \texttt{TeX} file.

**Notion:** This can be useful for DVI inverse-search or detection of the problematic line from \texttt{LaTeX} compilation. Note that before this function can be used export to \texttt{LaTeX} output must occur (in order to map the row numbers).

**Syntax:** server-goto-file-row <FILE>[.ext] <ROW_NUMBER>

**Params:** 
- `<FILE>`: the filename. Environment variables are expanded in the path. In case this LFUN does not work make sure you are giving correct path to the file.
  - If the file is located inside LyX temporary directory it will be mapped back into the appropriate opened buffer (e.g. for the case of generated .tex file).
- `.ext`: extensions will be automatically replaced by .lyx.

**Origin:** Edmar, 23 Dec 1998

LFUN_SERVER_NOTIFY.

**Action:** Sends notify message about the last key-sequence to client.

**Notion:** This can be used to grab last key-sequence used inside the LyX window. See also Debug extensions section in Additional features manual.

**Syntax:** server-notify

LFUN_SERVER_SET_XY.

**Action:** Sets the cursor position based on the editing area coordinates (similar as clicking on that point with left mouse button).

**Syntax:** server-set-xy <X> <Y>

LFUN_SERVER_GET_XY.

**Action:** Returns the coordinates of cursor position in the editing area.

**Syntax:** server-get-xy
LFUN_BUILD_PROGRAM.

**Action:** Generates the code (literate programming).

**Notion:** LaTeX file with extension `literates_extension` is generated. Then LYX invokes `build_command` (with a default of "make") to generate the code and `build_error_filter` to process the compilation error messages.

In case you want to process your literate file with a script, or some other program, just insert in your lyxrc file an entry with:

```
build_command "my_script my_arguments"
```

The `build_error_filter` differs from the `literates_error_filter` only in that the former will identify error messages from your compiler.

**Syntax:** `build-program`

LFUN_BUFFER_AUTO_SAVE.

**Action:** Saves the current buffer to a temporary file.

**Notion:** Saves the current buffer to a file named "#filename#". This LFUN is called automatically by LYX, to "autosave" the current buffer.

**Syntax:** `buffer-auto-save`

LFUN_BUFFER_CHILD_OPEN.

**Action:** Loads the given child document.

**Notion:** The current document is treated as a parent.

**Syntax:** `buffer-child-open <FILE>`

**Params:** `<FILE>`: Filename of the child. The directory of the parent is assumed by default.

**Origin:** Ale, 28 May 1997

LFUN_BUFFER_CHKTEX.

**Action:** Runs chktex for the current document.

**Syntax:** `buffer-chktex`

**Origin:** Asger, 30 Oct 1997

LFUN_BUFFER_TOGGLE_COMPRESSION.

**Action:** Toggles compression of the current document on/off.

**Syntax:** `buffer-toggle-compression`

**Origin:** bpeng, 27 Apr 2006

LFUN_BUFFER_CLOSE.

**Action:** Closes the current buffer.

**Notion:** Closes the current buffer, asking whether to save it, etc, if the buffer has been modified.

**Syntax:** `buffer-close`

LFUN_BUFFER_EXPORT.

**Action:** Exports the current buffer (document) to the given format.

**Syntax:** `buffer-export <FORMAT>`

**Params:** `<FORMAT>` is either "custom" or one of the formats which you can find in Tools->Preferences->File formats->Format. Usual format you will enter is "pdf2" (pdflatex), "pdflatex" (plain tex for pdflatex) or "ps" for postscript.

In case of "custom" you will be asked for a format you want to start from and for the command that you want to apply to this format. Internally the control is then passed to `LFUN_BUFFER_EXPORT_CUSTOM`.

**Origin:** Lgb, 29 Jul 1997

LFUN_BUFFER_EXPORT_CUSTOM.

**Action:** Exports the current buffer (document) from the given format using the given command on it.

**Syntax:** `buffer-export-custom <FORMAT> <COMMAND>`

**Params:** `<FORMAT>` format to start from (LYX will care to produce such intermediate file).

 `<COMMAND>` this command will be launched on the file. Note that you can use "$\$FName$" string to qualify the intermediate file.

**Sample:** `buffer-export-custom dvi dvips -f $$FName$$ -o myfile.ps`

**Origin:** leeming, 27 Mar 2004
LFUN_BUFFER_PRINT.

**Action:** Prints the current document.

**Notion:** Many settings can be given via the preferences dialog.

**Syntax:** buffer-print <TARGET> <TARGET-NAME> <COMMAND>

**Params:**
- `<TARGET>` is either "printer" or "file".
- `<TARGET-NAME>` is either "default" or file name or printer name.
- `<COMMAND>` command ensuring the printing job.

**Sample:** buffer-print file "/trash/newfile1.ps" "dvips"

**Origin:** leeming, 28 Mar 2004

LFUN_BUFFER_IMPORT.

**Action:** Import a given file as a lyx document.

**Notion:** File can be imported if lyx file format is (transitively) reachable via defined convertors in preferences. Look into File->Import menu to get an idea of the currently active import formats.

**Syntax:** buffer-import <FORMAT> [<FILE>]

**Origin:** Asger, 24 Jul 1998

LFUN_BUFFER_NEW.

**Action:** Creates a new buffer (that is, document).

**Notion:** Implicit path can be set in Preferences dialog.

**Syntax:** buffer-new [<FILE>]

**Params:** `<FILE>`: filename of created file with absolute path.

LFUN_BUFFER_NEW_TEMPLATE.

**Action:** Creates a new buffer (that is, document) from a template.

**Notion:** Path for new files and templates can be set in Preferences dialog. Template will be asked for via Open-dialog.

**Syntax:** buffer-new-template [<FILE>]

**Params:** `<FILE>`: filename of created file with absolute path.

LFUN_BUFFER_RELOAD.

**Action:** Reverts opened document.

**Syntax:** buffer-reload

**Origin:** Asger, 2 Feb 1997

LFUN_BUFFER_SWITCH.

**Action:** Switch to the given buffer.

**Notion:** This is useful also in case you need simultaneously more views of the edited document in different LyX windows.

**Syntax:** buffer-new-template <BUFFER>

**Params:** `<BUFFER>`: already opened document which is to be shown.

LFUN_BUFFER_TOGGLE_READ_ONLY.

**Action:** Toggle editing mode of the current document between read/write and read-only.

**Notion:** In the ->Readonly mode checks-in/commits the data if the file is under version control. In the Readonly-> mode checkouts the data from repository. If these operations fail, buffer won't be toggled.

**Syntax:** buffer-toggle-read-only

**Origin:** Lgb, 27 May 1997

LFUN_BUFFER_VIEW.

**Action:** Displays current buffer in chosen format.

**Notion:** Displays the contents of the current buffer in the chosen format, for example, PDF or DVI. This runs the necessary converter, calls the defined viewer, and so forth.

**Syntax:** buffer-view <FORMAT>

**Params:** `<FORMAT>`: The format to display, where this is one of the formats defined (in the current GUI) in the Tools>Preferences>File Formats dialog.
**LFUN_BUFFER_UPDATE.**

**Action:** Exports the current document and put the result into the temporary directory.

**Notion:** In case you are already viewing the exported document (see LFUN_BUFFER_VIEW) the output will be rewritten - updated. This is useful in case your viewer is able to detect such changes (e.g. ghostview for postscript).

**Syntax:** buffer-update <FORMAT>

**Params:** <FORMAT>: The format to display, where this is one of the formats defined (in the current GUI) in the Tools>Preferences>File Formats dialog.

**Origin:** Dekel, 5 Aug 2000

**LFUN_BUFFER_WRITE.**

**Action:** Saves the current buffer.

**Notion:** Saves the current buffer to disk, using the filename that is already associated with the buffer, asking for one if none is yet assigned.

**Syntax:** buffer-write

**LFUN_BUFFER_WRITE_AS.**

**Action:** Rename and save current buffer.

**Syntax:** buffer-write-as <FILENAME>

**Params:** <FILENAME>: New name of the buffer/file. A relative path is with respect to the original location of the buffer/file.

**LFUN_BUFFER_WRITE_ALL.**

**Action:** Save all changed documents.

**Syntax:** buffer-write-all

**Origin:** rgh, gpothier 6 Aug 2007

**LFUN_BUFFER_NEXT.**

**Action:** Switch to the next opened document.

**Notion:** Note that this does not necessarily mean next in tabbar (for full list see View menu).

**Syntax:** buffer-next

**LFUN_BUFFER_PREVIOUS.**

**Action:** Switch to the previous opened document.

**Syntax:** buffer-previous

**LFUN_MASTER_BUFFER_UPDATE.**

**Action:** When run from a child document, this updates (exports) document built from the master buffer. If a master is not found, it updates the current buffer.

**Syntax:** master-buffer-update

**Origin:** Tommaso, 20 Sep 2007

**LFUN_MASTER_BUFFER_VIEW.**

**Action:** When run from a child document, this command shows a preview built from the master buffer. If a master is not found, it previews the current buffer.

**Syntax:** master-buffer-view

**Origin:** Tommaso, 20 Sep 2007

**LFUN_BUFFER_LANGUAGE.**

**Action:** Set language of the current document.

**Syntax:** buffer-language <LANG>

**Params:** <LANG>: language name. See lib/languages for list.

**Origin:** leeming, 30 Mar 2004

**LFUN_BUFFER_SAVE_AS_DEFAULT.**

**Action:** Save the current document settings as default.

**Notion:** The file will will be sa v ed into ~/.lyx/templates/defaults.lyx .

**Syntax:** buffer-save-as-default [<ARGS>]

**Params:** <ARGS>: contains the particular settings to be saved. They obey the syntax you can find in document header of usual .lyx file.

**Origin:** leeming, 30 Mar 2004
LFUN_BUFFER_PARAMS_APPLY.
Action: Apply the given settings to the current document.
Syntax: buffer-params-apply [<ARGS>]
Params: <ARGS>: contains the particular settings to be saved. They obey the syntax you can find in document header of usual .lyx file.
Origin: leeming, 30 Mar 2004

LFUN_FILE_INSERT.
Action: Inserts another LyX file.
Syntax: file-insert [<FILE>]
Params: <FILE>: Filename to be inserted.

LFUN_FILE_INSERT_PLAINTEXT.
Action: Inserts plain text file.
Syntax: file-insert-plaintext [<FILE>]
Params: <FILE>: Filename to be inserted.
Origin: CFO-G, 19 Nov 1997

LFUN_FILE_INSERT_PLAINTEXT_PARA.
Action: Inserts plain text file as paragraph (i.e. join lines).
Syntax: file-insert-plaintext-para [<FILE>]
Params: <FILE>: Filename to be inserted.
Origin: Levon, 14 Feb 2001

LFUN_FILE_OPEN.
Action: Open LyX document.
Syntax: file-open [<FILE>]
Params: <FILE>: Filename to be opened.

LFUN_CALL.
Action: Executes a command defined in a .def file.
Notion: The definitions are by default read from lib/commands/default.def.
A .def file allows to define a command with \define "<NAME>" "<LFUN>" where <NAME> is the name of the new command and <LFUN> is the lfun code to be executed (see e.g. LFUN_COMMAND_SEQUENCE). \def_file "FileName" allows to include another .def file.
This is particularly useful in connection with toolbar buttons: Since the name of the button image for this lfun is lib/images/commands/<NAME>.png this is the way to assign an image to a complex command-sequence.
Syntax: call <NAME>
Params: <NAME>: Name of the command that must be called.
Origin: broderer, 2 Oct 2007

LFUN_META_PREFIX.
Action: Simulate halting Meta key (Alt key on PCs).
Notion: Used for buffer editation not for GUI control.
Syntax: meta-prefix

LFUN_CANCEL.
Action: Cancels sequence prepared by LFUN_META_PREFIX.
Syntax: cancel

LFUN_COMMAND_EXECUTE.
Action: Opens the minibuffer toolbar so that the user can type in there.
Notion: Usually bound to M-x shortcut.
Syntax: command-execute

LFUN_COMMAND_PREFIX.
Action: Return the current key sequence and available options as a string.
Notion: No options are added if no current map exists.
This is probably usable only with connection to lyxserver.
Syntax: command-prefix
**LFUN_COMMAND_SEQUENCE.**

**Action:** Run more commands (LFUN and its parameters) in a sequence.

**Syntax:** command-sequence <CMDS>

**Params:** <CMDS>: Sequence of commands separated by semicolons.

**Sample:** command-sequence cut; ert-insert; self-insert ; paste; self-insert {} ; inset-toggle;

**Origin:** Andre, 11 Nov 1999

**LFUN_COMMAND_ALTERNATIVES.**

**Action:** Runs the first listed command that is enabled.

**Notion:** This can be used to bind multiple functions to a single key, and then which one is used will depend upon the context.

**Syntax:** command-alternatives <CMDS>

**Params:** <CMDS>: Sequence of commands separated by semicolons.

**Sample:** command-alternatives completion-accept; cell-forward

**Origin:** rgh, 24 September 2008

**LFUN_MESSAGE.**

**Action:** Shows message in statusbar (for script purposes).

**Syntax:** message <STRING>

**Origin:** Lgb, 8 Apr 2001

**LFUN_PREFERENCES_SAVE.**

**Action:** Save user preferences.

**Syntax:** preferences-save

**Origin:** Lgb, 27 Nov 1999

**LFUN_RECONFIGURE.**

**Action:** Reconfigure the automatic settings.

**Syntax:** reconfigure

**Origin:** Asger, 14 Feb 1997

**LFUN_LYXRC_APPLY.**

**Action:** Apply the given settings to user preferences.

**Syntax:** lyxrc-apply <SETTINGS>

**Params:** <SETTINGS>: settings which are to be set. Take a look into ~/.lyx/preferences to get an idea which commands to use and their syntax. lyx::LYXRC::LYXRCTags has the list of possible commands.

**LFUN_CURSOR_FOLLOWS_SCROLLBAR_TOGGLE.**

**Action:** Determine whether keep cursor inside the editing window regardless the scrollbar movement.

**Syntax:** toggle-cursor-follows-scrollbar

**Origin:** ARRae, 2 Dec 1997

**LFUN_SET_COLOR.**

**Action:** Set the given LyX color to the color defined by the X11 name given.

**Notion:** A new color entry is created if the color is unknown. Color names can be stored as a part of user settings.

**Syntax:** set-color <LYX_NAME> <X11_NAME>

**Origin:** SLior, 11 Jun 2000

**LFUN_STATISTICS.**

**Action:** Count the statistics (number of words and characters) in the document or in the given selection.

**Notion:** Note that this function gives the number of words/chars written, not the number of characters which will be typeset.

**Syntax:** statistics

**Origin:** lasgouttes, Jan 27 2004; sanda, Jan 8 2008
LFUN_COMPLETION_INLINE.
   Action: Show the inline completion at the cursor position.
   Syntax: completion-inline
   Origin: sts, Feb 19 2008

LFUN_COMPLETION_POPUP.
   Action: Show the completion popup at the cursor position.
   Syntax: completion-popup
   Origin: sts, Feb 19 2008

LFUN_COMPLETION_COMPLETE.
   Action: Try to complete the word or command at the cursor position.
   Syntax: complete
   Origin: sts, Feb 19 2008

LFUN_COMPLETION_CANCEL.
   Action: Try to cancel completion, either the popup or the inline completion
   Syntax: completion-cancel
   Origin: sts, Sep 07 2008

LFUN_COMPLETION_ACCEPT.
   Action: Accept suggested completion.
   Syntax: completion-accept
   Origin: sanda, Sep 08 2008

LFUN_BRANCH_ACTIVATE.
   Action: Activate the branch
   Syntax: branch-activate <BRANCH>
   Params: <BRANCH>: The branch to activate
   Sample: lyx -e pdf2 -x "branch-activate answers" finalexam.lyx
      could be used to export a pdf with the answers branch includedwithout one’s having to open L\LaTeX
      and activate the branch manually.
   Origin: rgh, 27 May 2008

LFUN_BRANCH_DEACTIVATE.
   Action: De-activate the branch
   Syntax: branch-deactivate <BRANCH>
   Params: <BRANCH>: The branch to deactivate
   Origin: rgh, 27 May 2008

LFUN_COPY_LABEL_AS_REF.
   Action: Copies the label at the cursor as a cross-reference to be paster elsewhere.
   Syntax: copy-label-as-reference
   Origin: sts, 16 Nov 2008

LFUN_BUFFER_ZOOM_IN.
   Action: Increases the zoom of the screen fonts.
   Syntax: buffer-zoom-in [<ZOOM>]
   Params: <ZOOM>: The zoom in %, the default is 20.
   Origin: vfr, 30 Mar 2009

LFUN_BUFFER_ZOOM_OUT.
   Action: Decreases the zoom of the screen fonts.
   Syntax: buffer-zoom-out [<ZOOM>]
   Params: <ZOOM>: The zoom in %, the default is 20.
   Origin: vfr, 30 Mar 2009