Introduction to LyX

by the LyX Team*

January 7, 2014

Contents

---

*If you have comments on or corrections to this documentation, please send them to the LyX Documentation mailing list: lyx-docs@lists.lyx.org

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, indexes, etc. It is very good for working with 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. Here are 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 and 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 18 pt bold instead of 16 pt, so you type in the heading for Section 4 in a different font from the one 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 is 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

---

[1]To 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.

use a screwdriver to drive in nails.

1.2 Differences between L_YX and Other Word Processors

Here's a list of things you won't find in L_YX:

- The document ruler

- Tab stops

- Extra whitespace (e. g. hitting Enter or Space two or more times)

Tab stops, along with a ruler showing you the position of things on the page, are useless in L_YX. The program worries about where things go on the page, not you. Extra whitespace is similar; L_YX adds it 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 L_YX, 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 L_YX, you generally don't need them. L_YX 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 L_YX.[2]

Lastly, there are a few areas where we believe L_YX (and L^AT_EX) surpasses many word processors:

---

[2]There are ways to adjust all of these (only some of which require knowledge of L^AT_EX), 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 WYSIWYM concept. However, they've only recently been able to do so, whereas LyX is built upon the LaTeX document preparation system. LaTeX has been around since 1985, and *works*.

1.3   What is LaTeX?

LaTeX is a document preparation system designed by Leslie Lamport in 1985.[3] 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, hence 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 a 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 2_ε, the current version of LaTeX, during the late 1980's. This new version of LaTeX has commands which provide an

---

[3]The source for the info in this section is *A Guide to LaTeX 2_ε*, by Helmut Kopka and Patrick Daly, which has an entry in the bibliography of the *User's Guide*.

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 *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 *style* differs from a class in that it doesn't define a new type of document, but a different type of *behavior* that any document can use. For example, 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:

TeX: Typesetting language with macro capability.

LaTeX: Macro package built upon TeX.

classes: Descriptions of types of document used with LaTeX.

styles: Descriptions of the default behavior of particular elements of LaTeX.

LyX: Visual, WYSIWYM document processor that uses LaTeX to do its typesetting.

This section attempts to explain the difference between LyX and a word processor. Simply put, LaTeX is the difference. By using LaTeX as its backend, LyX helps you to think more about *what* (as in the *words*) you write. The computer then handles *how* they should look.

2 Navigating the Documentation

To make it easier to answer your questions and describe all of the features of LyX, the documentation has been split up into several different files. Each one has its own purpose, as described below. Before you go ploughing into any of those files, however, you should read this chapter thoroughly, since it contains a lot of useful information and commentary that can save you some time.

Hopefully, the development of LyX will never stop; so some of the documentation may be incomplete or a bit out of date, though we try to keep up-to-date. Like the rest of LyX, the manuals are the work of a group of volunteers who have "Real Jobs", families, dishes to clean, *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 emailing lyx-docs@lists.lyx.org. If you have questions that 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 of you may have printed out the manuals. Others may be reading them within LyX. There are some differences between the LyX-file and the printed version. In LyX, 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: foot 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. In the LyX-file, 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 referenced 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. Click on Go Back to go back to your earlier location.

Now that we've cleared up some of the differences between the printed and LyX-file 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.

- "Left" "Right" "Up" "Down": self-explanatory.

- "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. LyX 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 LyX.

Table 1: Units

unit	name/description
mm	millimeter
cm	centimeter
in	inch ($1\,\text{in} = 2.54\,\text{cm}$)
pt	point ($72.27\,\text{pt} = 1\,\text{in}$)
pc	pica ($1\,\text{pc} = 12\,\text{pt}$)
sp	scaled point ($65536\,\text{sp} = 1\,\text{pt}$)
bp	big point ($72\,\text{bp} = 1\,\text{in}$)
dd	didot ($72\,\text{dd} \approx 37.6\,\text{mm}$)
cc	cicero ($1\,\text{cc} = 12\,\text{dd}$)
Scale%	% of original image width
text%	% of text width
col%	% of column width
page%	% of paper width
line%	% of line width
theight%	% of text height
pheight%	% of paper height
ex	height of letter x in current font
em	width of letter M in current font
mu	math unit ($1\,\text{mu} = 1/18\,\text{em}$)

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 through 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 have 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 formula.

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. It also includes information about layout files, which are needed to get LyX to support LaTeX classes or packages.

Shortcuts Tables of the currently defined LyX shortcuts.

LaTeX Configuration A report produced by LyX about your system. This file contains information 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 on which you are running LyX. 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/lyxgit/lib/doc?rev=master

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* are 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 to 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.