Structured writing with LyX

What you see is what you mean?

Terry Hancock

In the hubbub over the Open Document Format and competing "what you see is what you get" (WYSIWYG) word processors, a long-standing alternative model of word processing systems, with much deeper roots in the free software world, has been mostly overlooked. The author of LyX, Matthias Ettrich, calls this approach "what you see is what you mean" (WYSIWYM). However, it's a philosophy that you will find in many "native" free software text-processing systems everywhere, from online "content management systems" to book publishing. You write what you mean, then you use some type of formatter to create presentation layouts. Sometimes it's called "structured writing" or "structured authoring", but whatever name it goes by, you'll see this idea repeated in many places. LyX[1], with its integrated graphical environment, may be the friendliest place to learn it.

Starting an article in LyX

WYSIWYM does take some getting used to if you've come fresh from the world of WYSIWYG. The problem is that you get an urge to control details about presentation (rather than meaning or "content"), and you may try to control them too early in the authoring process.

With LyX, you don't need to do such things. First of all, the general purpose styles that are provided with the program are good enough for everyday writing tasks (so you won't often have to think about presentation at all). The main point, though, is that once you have the document created, you can apply transformations to the presentation after the fact, and they will be propagated throughout your document with a minimum of fuss.

> Let the program do the scutwork for you. Concentrate on what you are writing, not on how you are going to format it

So, my first bit of advice when starting out with LyX has to be "let go". Let the program do the scutwork for you. Concentrate on what you are writing, not on how you are going to format it. Just start typing. As the need arises, you can add styles to already written text.

Styles

If you take a look at the default "article" template for LyX, you can see what structured writing is all about. The document view in the window looks very much like it would in any WYSIWYG word processor, but you should notice that the style menu is prominent on the upper left. Most of your formatting work is done with this menu, and involves a single step of selecting the style for the current paragraph.

The main style you will use is the "Standard" style, which applies to ordinary paragraph text. After this, the next most frequent types you will deal with are the heading styles, called: "Part", "Section", "Subsection", "Subsubsection", "Paragraph", and "Subparagraph". For most documents,

Figure 1: The most commonly used widget in the LyX interface is the style menu

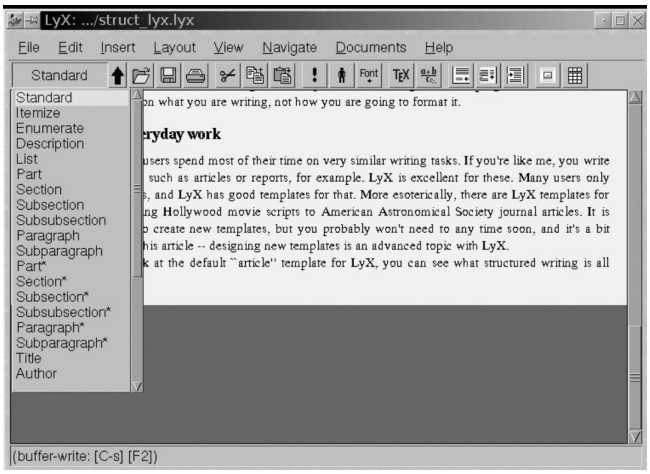
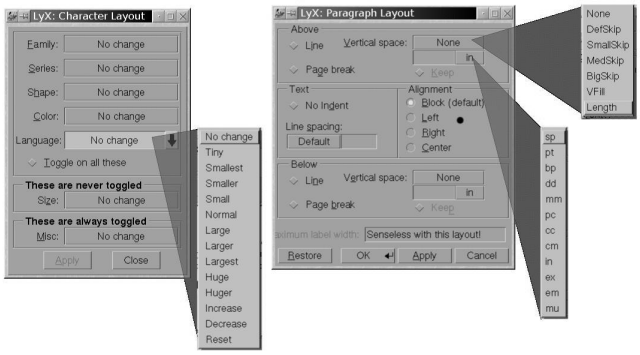


Figure 2: Dialogs for characters (left) and paragraphs (right), use relative terms rather than absolute ones, so that global document changes can be made later



you'll only need two or three levels of headings, but, as you can see, it is possible for a structure to be quite finely divided. If you use the numbered styles (the ones with no special markings), then section numbers will automatically be generated alongside of the headings, and autogeneration and navigation tools will treat them specially.

If you just want the "look" of a heading, but no numbering or special handling, there are variants of these styles (pre-pended with an asterisk, as in "*Subsection") provided for that purpose. There are also less-commonly used styles, such as "Quotation" for block quotations.

Layout menus

Styling of individual words is more limited, though there are provisions for bold, emphasis, and noun text on the "Layout" menu. These names may sound a little odd, because they describe the meaning of the typography rather than its implementation. So these terms are preferred to the literal terms for what happens in the standard article template: bold, italic, and small caps. For the feature-hungry, this may seem like sparse pickings, but it reflects the reality that for most writing applications it's extremely good advice to stick to these simple choices.

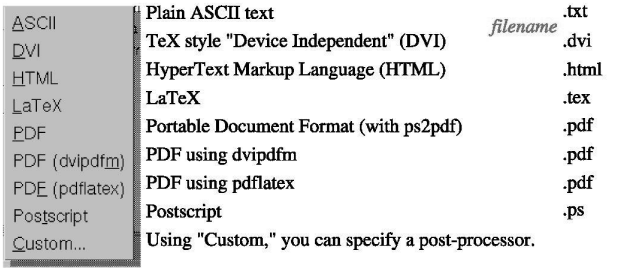
If you need more detail than this, however, it is there. Also in the Layout menu are options for the "Character…", "Paragraph…", and "Document…" layout characteristics. The character dialog will alter the text you currently have selected, while the paragraph dialog will affect the selected paragraphs or, if there is no selection, just the paragraph you are currently editing. The "Document…" options obviously affect the overall look of your document. Generally you won't need to worry about document layout until you are ready to print or export your work.

Most properties, such as text size, are indicated in generic or relative terms

One thing you may notice, is that most properties, such as text size, are indicated in generic or relative terms. For example the sizes of text you are presented with have names like "Smaller" and "Larger", rather than, say, exact point, metric, or pixel sizes. That's because the sizes are all defined relative to the normal size of text in the document. Later on, this will be convenient, because it will allow you to change all of the sizes up or down for the whole document. So, it shouldn't be surprising that the default behavior is not some fixed standard size, but simply "No Change".

Paragraphs have a similar variety of style factors affecting alignment, spacing, and indentation. The inter-paragraph spacing is controlled, for example, by defining a "Vertical space" in the "Above" or "Below" paragraph settings. Furthermore, you have a range of choices, from internally-defined relative spacings such as "BigSkip" or "SmallSkip" to specified lengths, which may be in physical (absolute) units such as inches or millimeters, or in typographical

Figure 3: Export options for LyX depend on what's installed on the computer. This set is fairly typical



units, such as "ex" or "sp" which are based on the size of font used (an "ex" is the height of the letter "x" in the font, while a "sp" or "space" is the full pitch between lines of typeset text). You will also note that you can choose to force a page break behind a certain paragraph or a horizontal line (these are good ways to layout chapter beginnings, for example).

One thing you will find missing, though, are the "natural" page breaks in the document. That's because the editing interface doesn't give them any special treatment—the handling of page breaks (aside from forced breaks) is handled as part of "document generation", not "document editing".

Generating output

Once you've finished your document, or even a first draft, you will probably want to look at the result. Most likely, you'll simply print it out on paper, which is easy enough to do. LyX actually accomplishes this, however, by converting the contents to LaTeX, then using the LaTeX interpreter to make a "Device Independent" DVI file, then a Postscript file, and possibly a Portable Document Format (PDF) file. The details of what kind of transformation is possible will vary according to what your system has installed on it. For example, figure 3 shows the LyX export menu on the system I am using to write this article.

Because of the way that LyX generates output, you can make many changes after writing your article, and expect to see them correctly propagated throughout your document. If you look in the "Layout" → "Document…" dialog, for example, you will find options to alter the margins, page numbering, fonts, and other overall properties. Since all of the individual elements of the document are defined in relative terms, the document will continue to look right after being altered.

When not to use LyX

No piece of software is the tool for every job. LyX is optimized for jobs where the writing is what's important and you want the layout to take care of itself as invisibly and automatically as possible. It's beyond the scope of this article, but it's also an appropriate tool if you want to define templates, so that you control the layout and allow other people to apply it effortlessly to their own work. But, it's probably not the best tool for laying out fancy advertising copy, comics, or other very creative layout and compositing jobs.

For those tasks, you might think of reaching for a conventional WYSIWYG word processor, but they aren't really the right tool for that job either (they're usually rather limited in what type of layout they can do, or at least in how easily it can be done). When I'm confronted with such situations, I usually pick either a vector graphics editor (Skencil[2] or Inkscape[3], for example), or a full WYSIWYG desktop publisher, like Scribus[4].

Bells and whistles

Naturally, LyX has some fancier features. Some of these will vary a lot in utility, depending on what kind of work you are using LyX for. One oddity you may wonder about is the way LyX will typeset "TeX", "LaTeX", and "LyX". This follows a tradition in the TeX community to use the name to show off the program's typesetting capacity. So it's really a logo as well as a name.

Figure 4: Some "stupid TeX tricks"—LyX automatically formats TeX-centric names according to their authors' wishes

TeX, LaTeX, and LyX

Figure 5: LyX has a very nice graphical equation editor

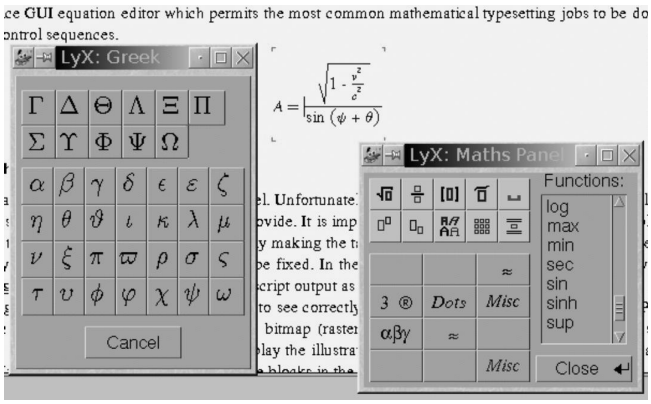


Figure 6: The same equation, as it appears in the finished document

$$A = \frac{\sqrt{1 - \frac{v^2}{c^2}}}{\sin(\psi + \theta)}$$

Templates, styles, and layout

Most people spend most of their time on very similar writing tasks. Some write lots of articles, reports, or technical documentation for which LyX's default styles are excellent. Others only write letters or memos, and LyX has good templates for that, as well. More esoterically, there are LyX templates for everything from writing Hollywood movie scripts to American Astronomical Society journal articles. Designing completely new templates is an advanced topic with LyX, but it's unlikely that you will need to anytime soon.

Mathematics!

Of course, deriving from TeX and LaTeX, it will be no surprise that LyX excels at typsetting mathematical equations. In fact, LyX includes a very nice GUI equation editor, which permits the most common mathematical typesetting jobs to be done without having to resort to TeX control sequences. What you see in the document you are editing is a representation of the equation, which may not look as professional as you had hoped. But don't worry, LyX will typeset the result using TeX with all the perfectionism that you'd expect from a system written by a mathematician.

Tables and graphics

LyX has a native table creation facility based on the TeX/LaTeX model. Unfortunately, that model is a bit outdated; so you will find that it comes nowhere near the simplicity of use that HTML tables provide. It's impossible, for example, to simply drop multiple lines of text into a table cell—the text will all be set on one line, possibly making the table far too wide (you have to create separate cells for each line, which can be made to look right, but it is time-consuming). This means that table typesetting natively in LyX can be pretty painful. Perhaps someday this will be fixed. In the meantime, however, I find that it's usually more efficient to produce tables in a vector graphic program, and import the Postscript output as a graphic.

LyX handles graphics quite naturally. If you want to see correctly scaled vector output, you'll want to use Postscript graphics. Otherwise, LyX can handle the most commonly used bitmap (raster) formats. One thing that is nice, is that since LyX doesn't attempt to use the literal WYSIWYG model, you can display the illustrations at a small size in your editor without affecting how they print (the "LyX View" and "Output" layout are separate blocks in the graphics dialog).

Outlines and nested bullets

The "Enumerate" and "Itemize" styles are used to create numbered or bulleted text, respectively. For a long time, I thought that LyX only supported a single level of bullets or numbers, but this is not true. It may not be the most intuitive naming scheme, but if you look in the "Layout" menu, you will find entries named "Increase Environment Depth" and "Decrease Environment Depth". These allow you to create a nested hierarchy of bullets or numbered sections to any desired depth.

If you do this a lot, it's also worth noting that there are hot keys for both "S-M-left" and "S-M-right" which for mere

mortals who didn't cut their teeth on Emacs, means: "hold down the 'shift' and 'alt' keys and press the left or right arrow keys". Emacs retains the term "meta" instead of "alt" from some archaic computer system predating the PC, if not recorded history. It's a useful fact to file if you use anything touched by the GNU project (and you will).

Evil red text

You will note there is a button with a red TEX logo in the LYX interface. (Do not press the red button!) This is how you (or rather a LATEX expert) can put "evil red text" into the document, which will be passed unaltered to LATEX. So, if all else fails, you may be able to do what you need to do in LATEX.

If you're the sort of person who reads "Do not the press the red button!", and can't stand not to, then you're probably qualified to play with evil red text, but *you have been warned*.

Under the hood

The typesetting engine for LYX is Donald Knuth's TEX[5]. It was never designed to be particularly user-friendly, focusing instead on producing beautifully-formatted text for academic publications. TEX excels most notably at typesetting mathematical equations, which was a major breakthrough. However, it's also an excellent system for setting conventional text. TEX is also legendary for its stability—being one of the few pieces of software that is almost certainly "bug-free"—in that it performs exactly as specified. There's been a bounty available for anyone who can document an exception to this for over a decade. Which, has gone unclaimed. Knuth who is also the author of a popular set of computer science textbooks (maybe "classic" would be appropriate by now), is therefore accorded a great deal of respect in the programming community.

On top of TEX, lies Leslie Lamport's LATEX[6], which is where the original idea of separating document structure from document presentation comes from. This idea was so successful, that it has become a standard design philosophy throughout the free software world.

By separating these elements, the job of the writer (producing content) is cleanly separated from the job of publisher (presenting that content in an aesthetically pleasing way). This simplifies the writers' jobs, because they no longer have to worry about petty details such as "what font size should sub-headings have?". Instead, they can specify "sub-heading" and get on with their thesis.

This idea is particularly useful in a collaborative context, where many different authors are submitting content to be published in a consistent way. This was important to the original users of LATEX, who were mostly academic writers. Today it is common in the free software and free culture community, and on the world wide web in general, partly because collaboration is such an important part of those communities.

> Because of the way that LYX generates output, you can make many changes after writing your article, and expect to see them correctly propagated throughout your document

However, LATEX is still not "user friendly" by any stretch of the imagination. In order to write directly in LATEX, you use a text editor, and you have to know a lot of arcane command sequences and codes to lay your text out properly. It is conceptually simpler than TEX, but it's still not a "word processor".

That's where LYX comes in—it is an interactive word processor, which keeps track of the arcane codes for you, and provides controls in graphical menus. This aspect of LYX probably looks very familiar; but, unlike WYSIWYG systems, LYX retains the "structured document" mindset of LATEX. Although LYX does have its own native file format (*.lyx files), it easily exports to LATEX format because it shares the same conceptual model.

Structured document authoring

The structured document model is the one most native to the free software world. LATEX has been in use for many years on Unix and Unix-like operating systems, and many tools have evolved around it. LYX provides a very comfortable "word processor" environment. However, it strictly adheres to the design ideals of the LATEX model, maintaining a clean separation between content and presentation.

You will find the same ideas repeated throughout the free software world, with HTML, XML, and SGML authoring systems. The system on which Free Software Magazine itself is written is based on the same design principles. So, it's well worth your while to learn how to use this tried and true approach to writing, even if it does feel a little strange at first. LyX is a free software native!

Bibliography

[1] LyX website (http://lyx.org)

[2] Skencil website (http://skencil.org)

[3] Inkscape website (http://inkscape.org)

[4] Scribus website (http://scribus.net)

[5] TeX has no central website, but you can find good sources from Wikipedia (http://en.wikipedia.org/wiki/TeX).

[6] LaTeX is also well-documented on Wikipedia (http://en.wikipedia.org/wiki/LaTeX)

Copyright information

About the author

Terry Hancock is co-owner and technical officer of Anansi Spaceworks (http://www.anansispaceworks.com), dedicated to the application of free software methods to the development of space.