

# Documentation Project Style Sheet

by John Weiss

January 20, 2008

## Abstract

This article is a style sheet. It describes, with examples, how the documentation should look and sound. The first few sections explain the font conventions and typography conventions all documentation writers should follow. Those sections also contain examples. It also explains the purpose of each of the different manuals. Follow it not merely to the letter, but also in spirit.

The Style Sheet for LyX documentation (hereafter known as the Style Sheet) applies to *all* forms of LyX documentation, regardless of language. There is a section for translators in the Style Sheet, towards the end. *Read the entire Style Sheet!* Even if you are a translator, I assume you know enough English to comprehend this document.

## 1 Questions and Clarifications

After the second version of this Style Sheet grew uncomfortably large, the LyX DocTeam decided it needed to lose some excess weight. It seems the Style Sheet began to specify too many special cases, too many points of clarification. On the other hand, contributors to the documents were discovering many creative ways of misinterpreting the Style Sheet specifications. Therefore:

If you have any questions about anything in the Style Sheet, *ask first, write second!*

Field all questions to the LyX Developer's Mailing List. There are seasoned DocTeam members who can answer your questions. If you have any problems with the Style Sheet itself, also contact the mailing list.

## 2 Fonts

We'll start with the easiest section, yet also the most important.

This is how you should fontify text in the manuals:

*Emphasized*      general emphasis, generic arguments, titles of books, names the other manuals and of their sections, and notes from the authors

Do not overemphasize your text.

Typewriter      program names, file names, man-page names, L<sup>A</sup>T<sub>E</sub>X code, L<sup>A</sup>T<sub>E</sub>X commands, L<sup>A</sup>T<sub>E</sub>X package names, and L<sup>Y</sup>X code and functions

Sans Serif      menu, button, or popup names, the names/labels of all widgets in a popup, the names of keyboard keys, and certain “special terms”

NOUN STYLE      people's names

Underlined Sans Serif Rich Fields added this to mimick the underlining of letters in the menus and on buttons. It helps to highlight the accelerator keys, and human brains store information best when they see it frequently.

**WARNING!** — When you do this, make sure you *only* shut off the underlining. Too many people send in things that look like:

This

... i. e. they not only shut off underlining, they turned off Sans Serif, too! *Don't do that!* Make sure the entire word is still in Sans Serif after you shut off the underlining.

**Bold**      Unused.

If you want to emphasize any text, use *Emphasized*. L<sup>A</sup>T<sub>E</sub>X will put many things boldface on its own, such as Sections, certain parts of equations, et cetera.

Repeat: do not use boldface.

Here are some examples:

1. The function `math-mode` appears in configuration files and in the LyX source. Therefore, it (and all other LyX function names) count as code and is set in `Typewriter`.
2. However, Math mode is a menu item in the Math menu, so it appears in `Sans Serif`. Notice the use of Underlined Sans Serif for the accelerator keys.
3. Consider the following excerpt from the introduction of one of the manuals:

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.

Notice that the key name, Return, is in `Sans Serif`, but *only* when it references the key itself! When I described how the manufacturer chose to label its keyboard, I used Roman and put the word in quotes. There is a semantic difference.

4. Take the following command:

```
lpr -Pprintername
```

Notice that the argument to the `-P` option is in *Roman Italics* (i.e. emphasized). This is a case where you don’t use `Typewriter` for code, because you want the generic argument label to stand out. On the other hand, if you were specifying an argument, for example, “`lpr -Pduaneps`”, you’d leave it in `Typewriter`.

5. Any L<sup>A</sup>T<sub>E</sub>X commands and code, and any *unsupported* L<sup>A</sup>T<sub>E</sub>X package names get set in `Typewriter`. For example, “`multicol`” is the name of an unsupported L<sup>A</sup>T<sub>E</sub>X package, but “`book`” is a supported L<sup>A</sup>T<sub>E</sub>X class.

### 3 Keys

The canonical keyboard contains these keys:

- C- or Control-
- S- or Shift-
- M- or Meta-  
Self-explanatory. Be lazy and *use the abbreviations* whenever possible.
- F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12  
The function keys. Most modern keyboards have all 12.
- Esc  
The “Escape key.”
- Insert Delete Home End PageUp PageDown  
These are the 6 keys that appear above the cursor keys on many PC keyboards. Consider them as part of the standard motion keys.
- Left Right Up Down  
The four standard motion keys. There is no need to put the word “arrow” anywhere, since that’s obvious.<sup>1</sup>
- Return and Enter  
I won’t throw a hissy fit if you use one instead of the other. I’d prefer if you used **Return** over **Enter**, but it’s okay if you slip up and forget. Since these two keys are bound to the same function in LyX, it doesn’t really matter.

You do not need to explain everywhere what the *Meta* key is or where the **Left** key is, et cetera. The user isn’t stupid. Also, someone will document anything that isn’t clear (e. g. the *Meta* vs. **Alt** problem) someplace in the introduction. No need for you to repeat someone else’s work.

LyX does not support keyboards missing any of the keys described above, with one exception. LyX can support a keyboard missing F11 and F12. There is a keyboard accelerator for F10, but this is the only function key LyX assumes exists. Nevertheless, these details are of minor, if any, concern for the documentation. Assume the aforementioned keys exist.

---

<sup>1</sup>Same goes for “cursor key”. Even the word “key” after one of these may be redundant in certain situations.

## 4 Mice

The word “mouse” and any description of the 3 mouse buttons have no special handling. Don’t typeset them in any other font than the default, which is Roman.

Exception: you’re writing an Author’s Note (see section 8.2) and you need to mention something about the mouse. Since the rest of the note is in *Emphasized*, your description of “middle mouse button” should be emphasized, as well. There are a couple of other cases like this one; use your judgement.

There are only 3 mouse buttons. The use of them and of the mouse itself is obvious. There are few — if any — nonstandard things we do with the mouse. Therefore, there’s no need to make the word “mouse” or “mouse button” stand out.

## 5 Special Typography

Do the following:

**Multi-word names** Use a Protected Blank between the words for menu and widget names. E. g.: Save As, not Save As.

This holds for things in **Typewriter** as well as **Sans Serif**, with one caveat. If you have a long code example, one that can’t simply be inlined and put in **Typewriter**, put that in a **LyX Code** environment.

I want the Protected Blank so that the name doesn’t get split between two lines, which is visually disruptive. If something with a Protected Blank is near the end of a line and overflows the margin, use a `\sloppypar` in that paragraph (consult a **L<sup>A</sup>T<sub>E</sub>X** book for more about “`\sloppypar`”) or manually add a hyphenation point.

**Special Terms** These are things like the following:

- HFill
- VFill
- Table Float
- Figure Float

Use **Sans Serif** font and, in the case of HFill and VFill, capitalize the first two letters.

Why are these terms special? They are concepts which the seasoned L<sup>A</sup>T<sub>E</sub>X-pert is familiar with, but which the new LyX user is not. I want them to stand out from the rest of the text, hence the use of Sans Serif for them.

Seasoned LyX Team Members: Are there other terms that require this special status? On the other hand, should we eliminate this style completely?

**Terminology** Note the following:

- “dialog” not “popup” or “window”. “Confirmation alert” not “dialog”.
- PostScript<sup>®</sup> is a registered trademark of Adobe Corp. *You must put the <sup>®</sup> after it or we’ll get sued!* I also want it written as seen here, always capitalized. Not “postscript<sup>®</sup>,” or “Postscript<sup>®</sup>” but “PostScript<sup>®</sup>” - both of them capitalized, in the default font (i. .e. Roman). If you must, cut and paste it from here.

I am going to say this again:

*You must put the <sup>®</sup> after PostScript<sup>®</sup> or we’ll get sued!*

I mean it! American companies like to sue anything that moves. We could get in *major trouble* by forgetting that <sup>®</sup>. So, don’t. Got it?

- Similarly, if you use any other registered trademark in any documentation, put the <sup>®</sup> after it, too.

**Menu Items** When quick-referencing an item in a menu, use the menu separator: “▷”. Example: File▷Save. Notice that there are *no spaces* around the “▷” and that it’s in Sans Serif, just like the menu and item names.

1. The reason why I want no spaces around the “▷” is to prevent LyX from splitting terms across lines. The same goes for using Protected Blanks between multi-word terms. The split would be visually disruptive.

2. A “▷” goes between menu names and item names *only*. (Yes, submenus are okay, too).
3. *NEVER* put “▷” between menu items and dialog names. Example: “Layout▷Paper▷Paper Dialog” *IS STRICTLY FORBIDDEN!*  
*NEVER* put “▷” between popup names and any dialog. Example: “Paper Dialog▷Portrait” *IS STRICTLY FORBIDDEN!*  
*NEVER* put “▷” between menu items and any dialog. Example: “Layout▷Paper▷Portrait” *IS STRICTLY FORBIDDEN!*  
 Either write out the description, or use context to eliminate any need to repeat menu items, dialog names, etc.

**Note Boxes** LyX has a feature for adding comments that appear only within the LyX GUI. Here’s one: . You will see nothing in a printout of the Style Sheet. Therefore, they have no place in the manuals. Period.

If you have a parenthetical comment you want to make, the reader should see it too, even in the printed version. Use an Author’s Note (see section 8.2) in place of the Note-Boxes.

“(…)”, “[...]” and “{...}” I have recently been corrected about the use of parentheses. Standard English typesetting uses the normal parentheses, “(…)”, around any aside, remark, or parenthetical expression. The bracket, “[...]”, is used only within quotations (see section 8.3.3). The brace, “{...}”, is never used. Therefore, never use “[...]” or “{...}” unless otherwise specified by this Style Sheet.

**Dashes:** Be sure to use the correct one. A single “-” character is not a dash, it’s a hyphen. Use it only as a hyphen.

Instead, use an “en-dash” or an “em- dash.” Two back-to-back “-” characters form the en-dash. Three “-” characters create an em-dash, which is slightly longer than the en-dash. In the printed version of any document, LyX will combine the two or three “-” characters into a single, unbroken line.

## 6 Cross-References and Labels

Use the following labelling conventions:

- sec:xxx Use this for Sections as well as Chapters, Subsections, Subsubsections, etc.
- eqn:xxx Use this for Equations, should you need to create any.
- tbl:xxxx Use this for tables inside of a table float.
- fig:xxx Use this for figures inside of figure floats.

Additionally, you should put the label at one of two locations:

1. The *beginning of the paragraph*, after a section heading, or at the beginning of captions, etc. It should be the first thing on the first line. Don't put a space between it and the first word.
2. If there is no paragraph after a section heading, put it at the *end of the last line*.

Example: You have a Section which is immediately followed by a Subsection heading. This is a case where you need to put the label at the end of the Section heading. I know it looks ugly; not much we can do about that, though.

## 7 Content — What Goes Where

This is *very* important to anyone documenting a new feature. If you need to add new documentation, pay attention.

In the dim and distant past, whenever someone wanted to document a new feature they added, they either wrote a mini-doc and stuck it into the documentation directory, or they added a new section to the lone manual. No one paid much attention to organization in those days, either. The result was a totally bloated, totally un navigable, and incomplete manual orbited by a swarm of tiny, incomplete mini-docs. I don't want the docs to fall back into that mess.

With that in mind, I have some instructions for how to keep things organized:

`Intro.lyx` Please, don't touch this file. It's essentially complete, anyhow. Only the editor(s) should make changes to this, as this file contains info about how to contribute to the doc project. That's

really the only stuff that should need to change, and even then, only when a new maintainer comes along.

**UserGuide.lyx** This file is complete. Any changes should be for updates *only*. DO NOT ADD new features to here willy-nilly. Let the editor decide if — and when — new sections go in here. Place any new features in...

**Extended.lyx** This is where all new features go from now on. It's in the style of a bound journal — each section is an “article” from the author of the feature. Also, anyone who writes a `.layout` file for a new document class should write a section describing that new class and how to use it. That also goes here.

Note, however, that you are *not* excused from following this Style Sheet just because the sections of **Extended.lyx** are in the form of a journal article.

**Tutorial.lyx** This file is complete. Do not change or add to without permission of the Documentation Project Editor.

**Customization.lyx** This document describes advanced features, most of which alter the look, feel, and behavior of LyX. This manual is still a bit incomplete, although that may change soon. Check for updates often.

If you are unsure whether or not something belongs in **Customization.lyx**, then, most-likely, it *really* belongs in **Extended.lyx**. Again, let the current editor of the LyX documentation decide if your new section should be migrated to **Customization.lyx**.

**Reference.lyx** I'd prefer to completely finish this one before doing anything else, but that's unrealistic. LyX keeps changing so much that I think the *Reference Manual* will be the last one completed. However, I'd like it if the developer's would add entries anytime they create a new function or popup. That would help things immensely!

Note that the *Reference Manual* follows this Style Sheet for the most part. There are, however, additional rules to follow when making new entries. At the top of this manual, there are examples of and a template for *Reference Manual* entries. Use them.

## 8 Writing Style: The Primary Manuals

While I want to make contributing to the Documentation Project as painless as possible for newcomers, I also want the newcomers to be painless on the existing Documentation Team! Ergo, I've written this section to give some flavor to guide everyone's individual style.

### 8.1 Language

All contributions to the *primary* L<sup>A</sup>T<sub>E</sub>X documentation must be in English. I don't care if it's British, Australian, or American English. The DocTeam editor will proofread for glaring mistakes and fix them.

Don't get hung up on semantics. English is a flexible language, and just because your Mothertongue-to-English dictionary gives only one translation for a word doesn't necessarily mean it's so. We've had some discussions and misunderstandings on the Developers' List because of this very problem. If something is unclear (or just plain weird) due to a translation problem, one of the American/British/Australian developers can fix it.

Nota Bene: by *primary* documentation, I exclude the translations. We usually don't have enough people to cover the manuals in one language, let alone more than one. Subsequently, the translations are just that — translations of the English version of the L<sup>A</sup>T<sub>E</sub>X manuals. The translation efforts require have their own set of guidelines. See section ?? for more info.

### 8.2 Wearing Many Hats: Commentary from the Author (i. e.: You)

I want to make it easy for everyone when it comes to documenting things. Some features are incomplete. Some, you might not know everything about. Sometimes, you may want to communicate something to me or the reader or other DocTeam members. Sometimes, you may want to “speak for yourself;” you want to say something that is your personal opinion and using “we” would be inappropriate.

In short, when you contribute to the L<sup>A</sup>T<sub>E</sub>X Docs, you wear many hats.

For occasions when you need to switch hats, I've designed some special mechanisms.

### 8.2.1 Personal Notes: The “Me” Hat

These are footnotes. They begin with the following:

Note from NAME OF PERSON:

...using the NOUN STYLE for the person’s name and without the quotes. The rest of the footnote is the actual comment.

Use these when you need to quote a comment by someone (usually yourself), and need to identify that person. This includes occasions when you need wear the “me” hat, i. e. to speak for yourself instead of for the LyX Team.

If the comment is too large to put in a footnote, don’t use a Personal Note. When quoting more than about 3 sentences or 5 lines of text, use a bona fide quote. Don’t use any leading “Note from NAME OF PERSON:” in that case. In a real quote, you’ll give credit to the original speaker in either the paragraph before or after the body of the Quotation.

### 8.2.2 Author’s Notes: The “Author” Hat

There will be times when you are not speaking for the LyX Team, yet you are not entirely speaking for yourself. Instead, you are speaking on behalf of the manual itself, as the author of the manual. Some examples of when you might need to do this are:

- You need to contradict something you just wrote because the feature isn’t quite ready yet, but you wanted to document what it will do.
- You need to leave a note for yourself.
- You need to leave a note for the editor or the other DocTeam members.
- You need to point out something about the manuals to the reader, something that doesn’t fit into the context of the current paragraph.

At such times, you are wearing your “I am the Author” hat, if you will.

The typography for an “Author’s Note” is as follows:

- They go in the body of the text, in brackets, “[ ]”, not any other form of parentheses. The bracket are in the default character style.

- The text of the note itself, however, is emphasized.
- Begin with the words, “*Author’s Note:*” and end with an em-dash, “—”, followed by your initials.

Here’s an example: [*Author’s Note: This is an example note. —jw*].

The form of the Author’s Note, by the way, isn’t a suggestion or request. It is *mandatory*. All Author’s Notes must begin with this text, verbatim: “[*Author’s Note:*”. Abbreviations to “AN” are or any other variant are forbidden. The Author’s Note must end with an em-dash, which is 3 “-” characters: “—”. Do not use 2 or 1 “-”; you must use 3 (and 5 is right out).

“Author’s Notes” are inherently transient, and should disappear as a manual matures.

### 8.2.3 Footnotes:

You are also free to use footnotes on their own in addition to the Personal Notes and/or Author’s Notes. I’ve frequently used footnotes to . . . well, to comment on parts of a section without putting the commentary into the body of the text.

**Mixing Footnotes and Personal Notes** Personal Notes always go in footnotes, and should be 5 lines or fewer. Any larger quotation should be quoted properly, using the rules of standard English. Place quotes in a Quotation paragraph environment.

**Mixing Footnotes and Author’s Notes** Author’s Notes should *never* go in footnotes.

**Mixing Personal Notes and Author’s Notes** Forbidden; these two are mutually exclusive.

### 8.2.4 Summary of Use

- Personal Notes:  
A *short* opinion — yours or another LyX developer’s — about anything. Anywhere in the manuals you wish to speak for yourself instead the the LyX Team, use this. If you have a long rant, however, quote yourself (see section 8.3.3).

- **Author’s Note:**  
Use this to describe things in LyX (or the manuals) that may change in the future or are somehow incomplete. Author’s Notes are supposed to disappear as a manual matures.
- **Plain Footnotes:**  
Used for text fragments that almost fit into the flow of the text...but not quite.

When using these three mechanisms, in addition to rigorously following their descriptions, please use them properly. I listed some additional restrictions previously. Some of you may balk at these restrictions. Nevertheless, there is a reason for them: if you have an overwhelming desire to mix or modify footnotes, Personal Notes, and Author’s Notes, you shouldn’t be using any of them. More specifically, you’re trying to use a hammer to drive in a screw. What you want to say probably needs to go into the main body of the text.

## 8.3 General Stylistic Guidelines

Everything in this section is *mandatory to all documenters*, regardless the language you’re writing in.

### 8.3.1 Typography

1. Use the typography rules outlined in the beginning sections of this document.
2. Don’t, however, mimic the typography of this file. Yes, the Style Sheet doesn’t follow the Style Sheet (grin).
3. There is some typographic freedom in those rules in earlier sections. Use that freedom wisely. Most importantly, never sacrifice the online appearance for the printed appearance and vice versa.

An example is in the *User’s Guide*. There is a footnote using the `multcols` command to divide a long `litemize` environment into 3 columns. It adds some L<sup>A</sup>T<sub>E</sub>X commands to the online version, the so-called “Evil Red Text” that some so vehemently oppose. Without it, however, that footnote takes up over two pages, most of which is empty space. This is an example of permitting a little ugliness in the online version to get the printed version to look right.

4. When in doubt, compromise.  
When in doubt, use good judgement.

### 8.3.2 Semantics

1. You are “we”.  
When you speak, you speak for the entire LyX Team, so use “we” instead of “I”.
2. The reader is “you”.  
Whenever you want to say something to the reader, use “you,” not some contorted construction to avoid being too informal.
3. Use the term “mouse” for both the physical pointing object (mouse, joystick, touch pad, track ball, etc.) and the mouse cursor which the physical object moves about the screen.
4. Use the term “cursor” for the little blinking vertical bar that indicates where text can/will appear next.
5. When in doubt, compromise.  
When in doubt, use good judgement.

### 8.3.3 Quoting Yourself and Others

In some cases, you’ll have something to say, an opinion of yours. Since this is your opinion, you’re not speaking for the LyX Team. You have so much to say, in fact, that it won’t fit into a Personal Note or an Author’s Note. In these cases you want to quote yourself.

Any time you wish to quote someone, be it yourself or someone else, there are standard rules one follows. Every language has its own rules. You should follow the rules that apply to the language of the document to which you are contributing.

This creates a problem for the primary documentation. The primary documentation is written in English, yet the contributors come from many countries. The quoting rules for English (well, American English, at least) are outlined in the following Figure Float, for your convenience. Read them if you need to.

## Quoting rules for English:

- The body of the quote belongs in a **Quotation** environment.
- The sentences prior to the quote should flow logically and smoothly into the quote.
- The sentences immediately following the quote should continue the flow of the text.
- You must, *must* credit the original author of the quote in the sentences immediately before or after the quote.
- Crediting the original author of the quote should not, however, disrupt the flow of the text.
- If you omit text from the beginning of the first sentence in the quote, the quote must start with the text “[...]”. This is an ellipsis in square brackets.
- If you omit text from the end of the last sentence in the quote, the quote must end with the text “[...]” followed by the sentence’s punctuation mark.
- If you omit any text from the middle of the quote, be it whole sentences or parts of sentences, replace it with the text “[...]”.
- The quote must be grammatically correct.
  - If the original is wrong, you must correct it.
  - If omitting part of the quote “breaks” it, you must correct the problem.
  - For missing words (e. g. “the” goes missing), place the word in square brackets, “[...]” and insert in the quote where needed.
  - For mangled word order, correct the mangled text, following it with the text “[sic]”.
- Spelling in the quote must be correct. Correct any misspelled words and place the text “[sic]” after the corrected word.
- Back-to-back bracket blocks merge together. Example: “[...] [the]” is wrong. It should be “[... the]”. 15
- If you correct the spelling in 2 or more consecutive words, you can get away with one “[sic]” after the last mistake.

### 8.3.4 Coverage

When describing a new feature or `*.layout` file, be sure to:

1. Be *clear, concise, and to the point*. KISS = “Keep It Short and Sweet” (or, “Keep It Simple, Stupid!”)
  - Do *not* write paragraph after paragraph of verbage.
  - Get to the point.
  - Take a look at the manual for a commercial word processor — it’s a fine example of how **NOT** to write documentation. It’s all pithy, substanceless verbage, and its *utterly useless!*
2. Avoid being pedantic like The Plague!
3. In the same vein, don’t write more than you have to. You’re not working in a vacuum — refer freely to other parts of the manual (and other parts of other manuals). Even if that “other part of the manual” is incomplete or empty, refer to it. Someone will fill it in eventually.
4. On the other hand, BE THOROUGH!
  - (a) You are documenting *features*, not widgets, not how the source code is organized.
  - (b) Group by feature, not by widget.
  - (c) Stay on topic — one Section should cover *one* feature. Use Subsections and further subdivisions to group things if you’re documenting several related features in a single Section.
  - (d) Describe EVERYTHING related to that feature, no matter where it is.
    - i. Example: Paragraph Indenting. Several popups control its behavior. You would document *all* of this: which popups control it, when you use which setting on which popup to do which operation, et cetera.
    - ii. Note from JOHN WEISS:  
I’ve had people only document one popup — literally. This added off-topic information and only described half of the feature, since other menus, popups, and even unbound functions

contained additional stuff.

I got *really* cranky when that happens, because it means *I* ended up fixing it. Bad help is worse than no help at all.

These remarks still hold true: you'll piss of the DocTeam editor if you do things wrong, because he'll have to fix your mistakes.

- (e) Remember, there are people who will reference *your* section, just as you're referencing someone else's. You do want what you write to be useful, don't you?
5. When in doubt, compromise.  
When in doubt, use good judgement.

### 8.3.5 NEVER NEVER *NEVER EVER* Treat the Reader as if She is Stupid

1. No dumbing-down.
2. No talking down to the reader.
3. The reader is smart enough to know what a mouse is.
4. The reader is smart enough to know how to use a keyboard, including the Shift-, Control-, and Meta- keys. (The introduction of most of the manuals takes care of the "Meta- is the Alt- key" issue, so you don't need to.)
5. The reader is smart enough to know that "at the cursor" means "where the text cursor is sitting right now, in the buffer currently visible." (Anything more than the word "cursor" is, IMO, superfluous and will be deleted. So, save yourself the typing, save the editor the cutting, and save the reader the strain of sifting through extra verbage that adds no content.)
6. Rule of thumb: the reader is not an imbecile. The reader is merely lost; point them in the right direction, and they can take it from there.

## 8.4 Tips for the English Version

When contributing to the primary — i. e. the English language version — of the LyX manuals, keep the following in mind.

#### 8.4.1 Write as if You're Talking with a Friend.

1. Think that way when you write. Play the dialogue in your mind.
2. Be as informal as you please (without being rude).

#### 8.4.2 AVOID the Passive Voice

1. No: "It is felt that this name best explains the command's purpose."  
We know full well who wrote the command: "The LyX Team felt ..."  
Or, better yet, "We felt that ..."
2. Things don't happen by magic - somebody or something did it. Only politicians use the passive voice to cover up who did something. If LyX reformats a paragraph, write, "LyX reformatted the paragraph." If `ispell` makes changes, write, "`ispell` changes the document."  
Rule of thumb: any sentence you can express as, "It was done by foo," you can express as, "Foo did it." Much nicer.
3. I know it's tough. We all hear way, way too much garbage English on the TV every day in the passive voice. Some people think it makes speech better. It doesn't. It makes speech baroque, if not outright byzantine. With a little effort, you can wean yourself off of it.
4. I *will make you rewrite* anything in the passive voice. It's awkward and hard to read.
5. Note to non-Americans:  
Using passive voice is generally considered bad style in the U. S. as it is too easy to obfuscate your words with it. It also bloats sentences, often unnecessarily.

#### 8.4.3 Short Sentences. Short Paragraphs.

In English, there is a grammatical error known as the "run-on sentence." The classic example of a run-on sentence is 7 clauses strung together with the word "and." There are, however, less obvious run-on sentences, ones using too many subordinate clauses. Such sentences may look elegant because they are complex. However, they are also extremely difficult to read because they are so complex.

In general, stick to short sentences in written English. Getting rid of passive voice (“... was done by...”) shortens and simplifies them. Hacking apart sentences with many dependent clauses is another way to shorten sentences. There are ways to do this yet still have a smooth-flowing paragraph.

While I’m talking about paragraphs, I’ll apply the “shorter is better” motto to them, as well. At the time I started with the manuals (and this Style Sheet), I didn’t pay too much attention to paragraph size. I’ve since become a big proponent of short paragraphs, with one idea per paragraph. While long, flowing, multi-concept paragraphs can be nice in novels, we’re writing manuals. Our goal is rapid information location and comprehension, not a literary prize.

There is a single exception to the short sentence, short paragraph rule. Particularly complex ideas may need more “breathing room.” However, you shouldn’t encounter such complex ideas often when documenting LyX. Try to keep things short, and use your judgement as needed.

To reiterate, yet again, something I said before:

When in doubt, compromise.

When in doubt, use good judgement.

Hopefully, you’ve got the idea (grin).

## 9 Translations

### 9.1 Rules of the Translating Trade

While translating anything, there are certain “tools of the trade” you should use. They will help you greatly.

#### 9.1.1 Translate one paragraph at a time.

Most people translate word by word. Clearly, you lose all context if you do that. A word may have multiple meanings. You can’t tell which unless you look at the rest of the sentence.

There is another level to the context issue, however. Your dictionary may translate multiple English words the same way. All those words mean *roughly* the same thing. Each one, however, covers a different shade of meaning, a different mood or intent. It is often difficult to resolve those shades of

meaning in the context of even one sentence. A paragraph, however, will provide that context.

### **9.1.2 You will not translate it correctly on the first try.**

Alright, I admit that you may be able to translate some of the sentences at first glance. If you know a language well, you may even understand over half of the text. Nevertheless, overconfidence can lead you astray. There will be some sentences, no matter how few, that will simply confound you.

It is generally a good idea to make multiple passes over a paragraph you're translating. Even if you translate the entire paragraph on the first pass, make a second one. You'll often improve upon your first attempt.

### **9.1.3 When in doubt, write down all of the meanings for a word.**

You can often translate tricky parts of a text using the context of the surrounding sentences. So, if you hit a word or phrase you don't know, translate it more than one way. Picking the most likely translations, summarize them in one to three words in place of an "exact" translation.

### **9.1.4 Using context, fix the meanings on the next pass.**

This is where your multiple translations of a single word become useful. Using the other sentences you translated, you can now translate that mystery-sentence without reconsulting your dictionary.

### **9.1.5 Fix the grammar only after you've finished translating the sentence.**

If there's a mystery phrase in the middle of a sentence, you can't translate the entire sentence. Why grammatically rearrange the words you translated already? You may need to restructure the sentence a second time once you figure out how to translate that mystery phrase. Better to wait until you've completely translated the sentence to clean up its grammar. That way, you do so only once.

### **9.1.6 If you can't translate it, skip it and come back to it on the next pass.**

Remember the earlier discussion of context and its immense usefulness? There is no sin in making multiple passes over a tricky passage.

### **9.1.7 Translate the meaning first. The rest can wait.**

The information content of the text under translation is the most important part. This is especially important for a manual, where the information *is the only* important part of the original document. Lose that, and you lose the very point of performing the translation.

## **9.2 Tips for the Translators**

Those of you contributing to a translation of the LyX manuals must follow a modified set of rules. The first few rules are analogous to those in section 8.4. There are additional rules and regulations that follow those first few.

### **9.2.1 Write as if you are explaining LyX to a colleague you know well.**

1. Think that way when you write. Play the dialogue in your mind.
2. Use a conversational style in your writing. Pretend you are teaching LyX to a colleague you know well.
3. Use a style that is polite without being too formal. If, in your culture, informal language is appropriate to use with a colleague, use informal speech in the translation of the manual.

### **9.2.2 AVOID Snobby, Academic, Specialized, or “Dead” Writing.**

In English, the passive voice appears formal, dry, barren. It also often adds unnecessary complexity. In other languages, however, this is not the case. There is nothing wrong with passive voice, and people use it frequently in everyday conversation. Nevertheless, your translation of the LyX manuals must avoid “dead” writing.

In Germany, there is a magazine known as „Der Spiegel.“ The writing in it is so complex, it is extremely difficult for non-native German speakers

to understand. While sophisticated, the writing style of „Der Spiegel“ is not what a German uses in everyday conversation. Nor is the writing style for a Russian mathematics journal. Such specialized or overly-sophisticated styles are “dead” in the sense that they are seldom used by normal people in everyday speech.

We who write the LyX manuals, original or translated, seek to *inform*. If we write in a style only a few people use, and use seldomly, we will fail to inform. Use a writing style that mirrors everyday speech (without being vulgar, of course).

### 9.2.3 Keep the Writing Simple.

For the English version, I wrote, “Use short sentences and short paragraphs.” What if, however, short sentences and paragraphs are something only children use in your language? What if, in yet another language, short sentences imply rudeness? Naturally, you would not want to use them in your translation.

Nevertheless, the translations of the LyX manuals should be as clear as the originals. So, for our international colleagues, we apply this rule: Keep your sentences and paragraphs as short as makes sense.

Remember: we’re translating manuals here, folks. Our goal is rapid information location and comprehension, not a literary prize. Try to keep your writing concise yet smooth-flowing. And use your judgement as needed:

When in doubt, compromise.

When in doubt, use good judgement.

### 9.2.4 Translators must follow the Style Sheet, too!

Everything in this manual — *except section 8.4* — applies to every LyX documenter, no matter what the language.

### 9.2.5 Translators must read the Style Sheet Supplement for their language.

For every translation project, there is a Supplement to the Style Sheet. It will be named:

`DocStyle_Supplement_<cn>.lyx`

... where “<cn>” is your language’s two-letter locale code. The Translation Project Chief for your language wrote this. If he hasn’t, pester him to do so! <*wink!*>

### 9.2.6 The English versions of the manuals are not Sacred Text.

You do not need to translate everything word for word. In fact, you shouldn’t. Keep to the spirit of the originals, not the letter. Be as creative as you want, as long as you *accurately* and *completely* convey all of the information contained in the English versions.

### 9.2.7 Any information in the L<sup>A</sup>T<sub>E</sub>X manuals must also be in the translations.

This falls under translating the originals accurately and completely.

- Omitting any feature description is *strictly forbidden*.
- Misrepresenting or misdescribing any L<sup>A</sup>T<sub>E</sub>X feature or operation *must be avoided*.
- The translation *cannot* outpace the original.  
If no one has documented new feature in the primary L<sup>A</sup>T<sub>E</sub>X manuals (i. e. the English versions), do not do so in the translations. If you’re really looking for something to do, either:

- ... focus on translating something you haven’t yet,  
OR
- ... update or repair the primary manual.

If you cannot or do not want to do one of the above, then take a break. Relax. Wait for the main manuals to catch up before translating anything else.

### 9.2.8 What you cannot translate, you may omit (usually).

Prepositions, idioms, metaphors, slang, Oh My! There’s a jungle of potentially untranslatable text you may face. Happily, none of these untranslatables are essential to the original text... usually. If you can’t translate a

phrase or two, try omitting them. If the rest of the paragraph still makes sense, then don't worry about the omission.

There may be special cases where omitting part of a sentence or paragraph violates rule 9.2.7. In those cases, *do not omit!* You must try and translate those tricky spots.

### **9.2.9 Translators may add their own fluff to the information content.**

After you do strip away all of the idioms, metaphors, slang, humor, and other “expendable text,” you may find that your translated manual is dull and dry. Why not add your own fluff? Add text that makes the manual a pleasure to read, that engages the reader. It may take the form of humor, or metaphors, or sayings. Whatever you add, it should be “in context.” It should not clash with the explanation of LyX features and functions.

## **9.3 For Translation Project Chiefs**

### **9.3.1 The First Is In Charge**

If you were the first person to start translating the manuals, you're the LyXDoc Translation Project Chief for your language. If you are the *only* person translating the LyXDocs, that automatically makes you the Translation Project Chief.

Amongst other things, that means that you must read this section and perform the tasks described here.

If you are a member of a LyX Documentation Translation Team, but *are not* its Chief, you may stop reading. The remainder of this section will be of no interest to you. If you came to the Style Sheet from the Supplement for your language, you may return to it.

If you have not read the Style Sheet Supplement for your language, you should read it now.

### **9.3.2 Read the Style Sheet**

No documenter is excused from following the Style Sheet, not even a Translation Project Chief.

Actually, it is *especially* important that the Translation Project Chiefs read the Style Sheet.

### 9.3.3 Make your translators read the Style Sheet

No documenter is excused from following the Style Sheet.

Since your translation team is translating, they know *some* English, at least. Therefore, they should be able to read the Style Sheet.

### 9.3.4 Provide a “Supplement” to this Style Sheet

There are parts of this Style Sheet that are English-specific. I have tried to provide a general, language-independent description of certain details in this section. Unfortunately, that general description doesn’t cover the specifics of each language.

That’s where you, as head of a LyXDoc Translation Team, come in.

Every Translation Team Chief is *required* to write a Supplement to the official Documentation Style Sheet, with specifics issues affecting your language. (You are, after all, the LyX Team expert on your native tongue.) Follow these guidelines when writing the Supplement:

1. Name the file:  
`DocStyle_Supplement_<cn>.lyx`  
...where “<cn>” is the two-letter code for your language. This is the same two-letter code that is part of the filenames for the translated manuals. Example: “\_de” for German, “\_sv” for Swedish, and “\_ru” for Russian.
2. Do not worry about where the file goes. The CVS maintainers will locate all documentation and Style Sheet Supplements in an appropriate place.
3. Document Properties:
  - For consistency, use the same document class and other document properties as the Style Sheet.<sup>2</sup>
  - Exceptions: Use margins, indentation/paragraph separation, language, and encoding appropriate for your language.
4. The title of the Supplement:

---

<sup>2</sup>Specifically, check the settings in the Document Layout popup. Use those in your Supplement.

- The title will use the “Title” paragraph environment. In your native tongue, the title will read:

Documentation Project Style Sheet:  
Supplement for the <foo> Translation Project  
(Replace “<foo>” with the name of your language.)

- If, in your language, “supplement” translates into “appendix,” we have a problem. In English, “Supplement” and “Appendix” have somewhat different meanings. An appendix is an extra part of a document. A supplement is an extra document.

Choose a replacement word accordingly. Whatever you choose to replace “Supplement,” it must not have the same translation as the word “appendix.”

5. Below the title, in the “Author” paragraph environment, place your name.

There will be no abstract.

6. The first Section of the Supplement:

The first thing you will do is strongly yet politely encourage the reader to stop reading the “Supplement” and go read the Style Sheet. The reader should not return to the “Supplement” until he has read *and* understood the Style Sheet proper.

### 9.3.5 Keep the Supplement Succinct

This Style Sheet is already very detailed. DocTeam members all have a lot to read. We don’t want to place an extra burden on translators. Therefore, keep the Supplement as short as you can without losing information.

### 9.3.6 Font Issues

The second Section will be about font issues... if you have any. Not all Translation Project Chiefs will need to deal with this issue. The fonts:

- Typewriter
- Sans Serif

- Roman  
*Emphasized (actually Italics)*
- Underlined
- **Bold**
- NOUN (ACTUALLY SMALL CAPS)

...certainly exist for all languages that use the Roman alphabet. Do they exist, however, for Greek? How about Cyrillic? These different fonts almost certainly do not exist for Devanagari, Chinese, Korean, Japanese, Hebrew, Arabic, and other scripts.

There will be some languages for which following the font-scheme specified in this Style Sheet may not be possible. If you are the Translation Project Chief for such a language, you have extra work.

In the font section of the Supplement, you will provide a new typographic style, designed specifically for your writing system. For consistency, the title of this section in every Supplement should translate into English as “fonts.” Before adding anything to this section, however, determine what this new typographic style will look like. Stick to the font specifications in this Style Sheet as best you can, whenever you can. When you cannot, use the following suggestions:

- Guidelines for “translating” fonts,  
or  
What to do when a font doesn’t exist:

Roman	Use the font that typesetters in your language use for printing books, manuals, etc. This will typically be the default font L <sup>A</sup> T <sub>E</sub> X (and L <sup>A</sup> T <sub>E</sub> X) uses in your language.
NOUN STYLE	This is for people’s names. If there is special font for names in your alphabet/writing system, use it in place of this. Otherwise, write names in the default font, typeset according to the rules of your language.
<i>Emphasized</i>	Use the font with which your language normally emphasizes text.

Use a font that is different from your language's equivalent of **Boldface**. In other words, your **Section**, **Subsection** and similar headers will be in one typeface, perhaps **Boldface**, perhaps not. Whatever that font is, avoid using it for *Emphasized* if at all possible!

**Typewriter** Pick up a computer program manual written in your language. It will use a special typeface for filenames, for command names, program names, and such. Use that same font in place of **Typewriter**.

**Sans Serif** Pick any other font that is different from the ones you're using in place of *Emphasized*, **Boldface**, and **Typewriter**. If you're unlucky, and your language's writing system doesn't have enough fonts, use the same font you picked to replace **Typewriter**. Only do this, however, if your alphabet/writing system has very few fonts to pick from.

**Sans Serif** Don't worry about this one.

If you use some special font on-screen to highlight the accelerator keys for menus, buttons, and other widgets, you might want to mimic that in the translations. It is not required, however. Therefore, if you can't mimic this typographic convention in your native writing system, don't.

Note that you may also want to describe fonts that your Translation Team should *never* use. For example, no contributor to the English/European versions may ever use **Boldface**, as this is used for section-headings. Since there are enough other fonts, we who use the Roman alphabet and its variants can afford to omit **Boldface**.

Once you have determined which fonts in your native writing system will replace one or more of the above, propose it to the LyX Developer's Mailing List. You may receive valuable feedback this way. If not, that's okay. If no one can read your writing system, and therefore cannot comment, that's also okay. Go ahead and describe the typographic standard you created in the Supplement.

Remember: stick to the font specifications in this Style Sheet as best you can, whenever you can.

### 9.3.7 Quoting Style and the Quote vs. Quotation Issue

The next section of the Supplement will cover the issue of quoting. Give it an appropriate title.

One of the first things you should do in that section is resolve the following issue:

- Decide whether `Quote` or `Quotation` is the correct paragraph environment for your language.
- In the Supplement, specify which one to use.

English has its own typography and style for quoting others. The Style Sheet describes that typography in section 8.3.3. Your language also has a specific typography and style for quotations. Describe that style in this section of the Supplement, too. Naturally, you do not need to go overboard. Section 8.3.3 of this Style Sheet is overly detailed for a good reason. Authors of the primary LyX manuals are not necessarily native English speakers. The members of your Translation Team, however, will all likely be native speakers of your language. Therefore, discuss proper quoting style of your native tongue to an appropriate level of detail.

### 9.3.8 Translations of Style Sheet Terminology

In the Supplement, you must provide a standard translation of certain key phrases for the members of your Translation Team. Place this in a section following the one about quotations.

In particular, standardize the translations of the phrases:

- Note from `<FOO>`:
- *Author's Note*:

Do *not* change the typography of the “Personal Note” and “Author’s Note,” however. Only provide a translation for the opening phrases. Insist that the members of your Translation Team use these two tools correctly.

While we are discussing proper use of the “Personal Note” and “Author’s Note” in translations, let’s talk about a related problem. The “Author’s Note” is meant to be a note from the author of a manual to the reader. In the case

of a translation, however, the translator is *not* the author! How then should a translator translate an “Author’s Note?”

You, as Translation Project Chief, must decide. You can forbid translation of pre-existing “Author’s Notes,” omitting them entirely instead. You could tell your translators to read any “Author’s Note” they may encounter, understand it, then write their own “Author’s Note” about the situation, not as a translation but as a personal opinion. You may decide on some other policy.

Whatever you decide, codify your policy in its own Section or Subsection or whatever. Place it near the section where you translated “Personal Note” and “Author’s Note”.

### 9.3.9 Lost in Translation

After describing all of the previous issues, create a new Section. Give it the name, “Lost in Translation,” or something similar.

In this section you will discuss any common English metaphors, humor, connotation, or other difficult to translate text. Try to balance brevity and completeness. Devote a Subsection — or even Subsubsections — to each specific issue.

### 9.3.10 ... “Yes, we mean now.” ...

Throughout the manuals, the DocTeam has used the following sentences:

If you haven’t read the `<Foo>` manual, go read it. Yes, we mean now.

This sentence will be tricky to translate, since it contains non-translatable connotations. Therefore, create a Subsection for this issue in your “Lost in Translation” section. Present the “... Yes, we mean now...” sentences, then present a translation, along with any explanation you feel necessary.

Here’s what those two sentences, sitting alone in their own paragraph, mean:

The first sentence uses the English conditional followed by an imperative. We, as the LyX team, are commanding the reader to go back to another manual. For example, the *Intro* manual is a prerequisite for all of the other manuals. The conditional clause preceding the command means, “You do not need to perform this command twice.”

The second sentence adds force to the command. Culturally, the imperative tense of a verb in English is not necessarily forceful. The way we wrote that command, “go read it,” is firm, yet polite. The reader may choose to ignore it. By following with, “Yes, we mean now,” we imply two things. First, we add some “resistive force” to the command. That second sentence reinforces the command, making it a bit harder to ignore. Second, the sentence itself implies a certain sense of urgency. You cannot merely wait until later to fulfill that command. The brief paragraph, and its sudden end, add still further subtle reinforcement to the command, “go do the required reading before using this manual.”

Note that all of this commanding and reinforcing is nevertheless in a polite format. Furthermore, it is in a subtle form. We are commanding the reader to do something, but in an indirect fashion. This way, the reader does not feel like we are bullying him.

### **9.3.11 Firm Convincing vs. Rudeness**

In the same part of the Supplement that you place the “...Yes, we mean now...” translation, discuss the English version’s use of “firm convincing.”

You see, here in America, we often say that everything is permitted unless explicitly banned by law. As a result, manuals for computer software are frequently ignored and the software subsequently blamed for not being “user-friendly.” This is where the use of “firm convincing” comes in.

We who wrote the manuals added sentences insisting that the reader not ignore certain parts of the documentation. We wrote in a manner that was polite, yet firmly asserted that the user was misusing the software if he did not read the manual correctly. We did not, however, want to sound threatening, coercive, or bullying.

In your culture, cajoling the reader into using the manuals correctly may not be necessary. It may, in fact, be outright rude. Additionally, translating the firm-but-convincing bits may not work. The translation may sound weird, or rude, or hostile. Therefore, you and your translation team will face many sentences that you cannot translate.

You, the Translation Project Chief, must discuss this issue. Try and find parts of the original manuals where some friendly but firm convincing does not translate properly. Use these cases as the basis for examples of the problem. Be sure to then offer a solution (i. e. how you want such sentences translated.) If stumped, ask for help on the LyX Developer’s List.

### **9.3.12 Anything Else?**

You can add more sections to the Supplement if you need to discuss other issues. There may be policies or guidelines that you want to set for your Translation Team. Be careful, however! Keep the Supplement *short*.