# `lyxport` documentation

## 1 Overview

The `lyxport` R package is for exporting LyX to MSWord--- which I sometimes have to do, under duress--- and perhaps other formats. Unlike LyX's built-in "MS Word Open Office XML" export, `lyxport` does proper cross-referencing including tables, figures (correctly sized), lists, equations; and it gets the appendices and bibliography all Working Nicely. Most of the heavy lifting is still done by Pandoc, as in LyX's built-in export option; but Pandoc--- wonderful though it is!--- doesn't get everything right even with well-known filters (as you have probably discovered yourself by now, else you mightn't be reading this). So the package contains a lot of *my* behind-the-scenes fiddly code in order to save *you* lots of manual post-tinkering.

The use-case I have in mind is basically my own: you have prepared a long and lovely paper in LyX that generates a perfect and pretty PDF. But for some reason you have to submit it to some journal that insists on miserable misbegotten manuscripts in MSWord. Sigh--- I sincerely feel your pain! There might be several rounds of back-and-forth between you and the journal, so you reeeally don't want to have to repeatedly do manual edits of the Word version of an 80% successful conversion using an editor you hate. The conversion should look reasonably decent, citations and cross-refs should all be correct, etc; but journals always fart around with the appearance of tables etc, so there's no need to get *every* detail of appearance exactly matching between PDF and MSWord. And that's the level that `lyxport` aims at.

To see the features, open "examples/eqntest1.lyx". I haven't tested every LyX feature; it's mostly just stuff I need. More things might get added.

### 1.1 Setup

Once you've installed the package, run `lyxprefhack()` to set things up for direct use from LyX. Then you should see an "MSWord (lyxport)" option in File->Export, and a "lyxport" item in LyX's Help menu (this file). You can also see a PDF version via `RShowDoc("lyxport-docu",package="lyxport")`.

After setup, you won't normally use this package from R yourself; it will just be invoked from LyX. (The exception is if you want to use `requote_lyx` or any other future "offline" support function— currently that's the only one). However, a keen user could experiment with the core function `lyxzip2word` (qv) directly, e.g. to produce other formats besides MSWord.

### 1.2 Do I need to modify my LyX document?

Currently the only mod you usually need to make to a LyX document, is adding one line of ERT to define the bibliography style (section 2). However, there are general limitations on the input LyX, some of which are documented below.

### 1.3 Are there are any other useful functions in `lyxport`?

The helper function `requote_lyx` tries to resolve inconsistent use of straight quotes, etc; you do have to use that manually from R, probably only once per LyX file. For example, I used it when creating this document, which was based on importing plain text files of R documentation.

## 2 Bibliographies

Citations seem to work fine, ie whatever you specify in LyX for the PDF version also appears in the Word version (after a *lot* of hacking on my part!). But for the bibliography itself, Pandoc does not understand "styles" in the same way that biber/biblatex does (or whatever animal of the Latex zoo it should be; I don't understand this stuff, and am happy to keep it that way!). AFAICS none of the ways you can alter bibliography appearance in LyX/Latex will be understood. Instead, you'll need to specify a "CSL style" manually in your LyX source, as follows. First, you will need to get a suitable CSL file from the internet, eg via `https://editor.citationstyles.org/searchByExample` and store it somewhere (see below for thoughts about where). Second, you'll need some ERT to tell `lyxport` (which will then tell Pandoc) about the CSL. Here's two options[1].

```
%% CSL journal-of-applied-genetics.csl
```

or

```
%% CSL ./journal-of-applied-genetics.csl
```

The first— i.e., with no path— is generally better; the CSL file is available for "global" use by `lyxport` on any LyX document. For that, you need to store the CSL file at the top of your biblio tree, ie in the folder "<top>" where "<top>/bibtex/bib/" contains the actual dot-bib file(s) you are using as biblio sources.

The second assumes you store the CSL file "locally", in the same folder as the LyX document that's mentioning it. In that case, you'll need some extra ERT and to insert the file (Insert->File->Child document, type "Verbatim" or "Program Listing") to ensure it is available during export, like this.

```
\ifdefined\theundefinable
```

Program Listing: journal-of-applied-genetics.csl

```
\fi
```

The ERT stops the CSL file itself from appearing in the rendered document (eg in PDF form), but it's still included in the zip file during File->Export->Lyx-zip.

[1] If you are reading this document in LyX, note that these are not real ERT— they are actually program listings inside red framed boxes. And the "include" box is a fake graphic.That's because they are not "for real" in this document. Don't copy-and-paste these; just put appropriate stuff in ERTs yourself.

## 3 Other formats

My requirement is for MSWord, so that's what I've concentrated on. Things might or might not work in other formats; you can experiment inside R with the `outext` and `panoutopts` arguments of `lyxzip2word`. Almost all the work is in generating a nice native-Pandoc document, and the final export step is just up to Pandoc, so there is a good chance things will work.

FWIW I personally would prefer to use ODT rather than MSWord if I could (since ODT is open-source and the MSWord *program* can import ODT), but unfortunately LibreOffice's maths importing is broken (as of v7.6 and for some time before that). [One example: the vertical bar, which I use extensively for conditional probability. But there's other things too.] "Barring" that for-me-deal-breaking limitation, 'lyxzip2word' can export quite nicely to ODT, except figure sizes are not respected--- whether that's down to Pandoc's ODT writer, or to a limitation of ODT itself, I don't know. You'd need to manually resize all the figures within LO :/ Maybe there's an option in native Pandoc to specify figure sizes, in which case I could probably add code to do that--- but there's no point unless the ODT maths things get fixed.

## 4 How it works

In LyX, there needs to be a new "File Format" (in Preferences) which is an alias for the existing MSWord format. (The same would apply to any other desired type of output.) Then there needs to be a "Converter" from "Lyx-zip (archive)" format to the new alias. When you pick "File->Export->[new format alias]", LyX will realize that the only way to produce the new format is to first export to Lyx-zip (producing, yes, a zip file), then run the new Converter. The latter launches R, and runs 'lyxport::lyxzip2word' which does everything else, starting from that zip file. "Everything else" means: LyX -> Latex -> pre-process -> Pandoc-tex-to-native -> post-process -> Pandoc-native-to-MSWord-or-other.

This package is very effective despite *my* limitations: I don't understand Pandoc's internal document structure, I have zero wish to learn new programming languages or arcane file formats, and I have a pretty limited understanding of Latex--- much of the point of LyX being to avoid having to remember all the Latex gruesome details! However, the good news is that LyX exports a highly structured and limited form of Latex (unless you insert really nasty ERT...) which makes it easy to "partially parse" the file using 'grep' etc to find constructs that need attention. Similarly, Pandoc exports a highly-structured "native" format which IMO is much easier to partially-parse than JSON--- for example, indentation is highly consistent, so I can usually find the range of lines to 'gsub' etc by matching indentation size. Then I monkeyed around (a lot) until things worked.

Because I rely on LyX's specific and tidy structuring of its Latex exports, 'lyxzip2word' simply won't work on generic Latex documents.

In slightly more detail, the steps are:

- From inside LyX, LyX exports to LyX-zip, containing all Lyx source files and all graphics in a single file.
- LyX calls a converter script to convert to target format (an "alternative MSWord"), which launches R and runs 'lyxzip2word'.

See 'lyxzip2word' for more detail on the actual conversion steps.

## 5 Pandoc options

Pandoc uses an output-format-specific "template" to control some aspects of its output, eg fonts. You might want to change the template; for example, Pandoc's default choices for MSWord "bold" and "monospace" look pretty weird to me. I'm not sure how all that works, nor how you tell Pandoc which template file to use (might require modify the export call inside 'lyxzip2word', to allow an argument).

## 6 Limitations

There are probably lots more than this. This first list seems fairly permanent— it's structural stuff.

- Labels must start with the right prefix for the thing they are labelling: "tab:" for table-floats, "fig:" for figure-floats, "eq:" for equations, "sec/subsec/par:" for sectioning, and "enu:" for lists. LyX will do this automatically for you when you set up a label, unless you perversely force it not to; so, don't do that.
- In order to "count", tables & figures have to be in floats with labelled captions. Each float can contain several actual tables or figures, but only one label. Requiring a label is reasonable; how else would you alert the reader to the existence and role of the table inside the main text?! Numbered sections and numbered equations don't need labels, unless they are cross-referenced.
- Figures: No absolute paths. (Otherwise, Lyx-zip-export uses lots of subfolders and I can't find the files.) I'm not sure about relative paths that go "upwards" either (eg "../sister/image.png").
- Equations inside tables won't be labelled.
- TOC & lists of Figures, etc. Since those are page-number-specific, they won't translate 100% between Latex & MSWord anyway.
- Multi-page figures using subfloat/ContinuedFloat *do* work, but you shouldn't put anything in the subfloat captions per se because they will not be printed (deliberately, Becoz Reasons). However, you can put things in the main caption of each continuation float.
- Your local "layouts" (LyX modules) won't be available unless you copy them somewhere else; see section 8. This is a LyX bug that might get fixed.

Overall, the biggest limitation is Pandoc's Latex reader, but there are other problems too. For example: the "cases" environment gets spurious RH paren on export, LibreOffice/ODT maths has got problems, ... Anyway, ere are some current limitations that I *might* fix in future:

- No Boxes (yet). In particular, resizebox and presumably its friends don't work (unfortunately, sinceI often use them to get tables to fit). Perhaps I should add code to delete them from the intermediate Latex files, so that at least something comes out.
- Only one Bibliography is produced.

- Equation numbering is either unsectioned (1,2,3,etc) or uses "section" as the prefix (1.1,etc). In principle it should also accept "chapter" or "part" for the prefix, but I haven't added that yet. Perhaps it should allow user-specified arbitrary prefixes, but it doesn't.
- Equations, Figures, and Tables in Appendices are numbered (A1,A2,...,B1,B2)--- no choice (and it's my preferred way). ?Should I allow user-specced Apx encoding, eg S1, S2 instead of A, B? I don't know how to enforce any of this in LyX/Latex anyway.
- Citations are all author/year, rather than square-brackety.

# 7 Requirements

This all needs Pandoc, ImageMagick (which at least on Windows should be installed automatically inside LyX's folder structure), and, of course, LyX itself.

The folders for the executables of Pandoc, LyX, and Rscript should be in the system path. If they aren't normally, then you can tell LyX to set them up automatically just within each LyX session, via "Tools->Preferences->Paths->PATH prefix".

I am going to assume that R_LIBS_USER and so on correctly set the libraries, ie the path(s) to R package folders.

# 8 The LyX userdir

There's a bug in LyX 2.4.2.1 (at least) which means that, if you have a personal LyX userdir (which surely everyone does), it can't be made visible to LyX during the export operation. Normally this doesn't matter, because the userdir is mostly about interactive session stuff such as keybindings, but it does mean that any special modules in your "<userdir>/layouts" folder will not be visible. If you can't live without them, you will just have to copy them to your "<LyX-system-dir>/layouts" folder.

For those who really want to know (be warned: do you really?), the problem is this: `lyxzip2word` needs to call a standalone LyX instance in order to export a Latex version of the source. (Most of the needed files--- the LyX source itself, any include-files, graphics--- are first exported into the Lyx-zip archive which is produced automatically, but the Latex version needs to be exported separately.) There is a handy CLI option "lyx --export latex <blah>" for that. Unfortunately, the latter seems to fall over if the userdir is set, either on the invocation line via "-userdir <blahblah>", or via the "LYX_USERDIR_24x" envar. Then LyX will crash with a SIGSEGV (and exit code 11 in Windows). When LyX calls Rscript, it *sets* that envar first--- which is normally good, coz it means that a converter could look at any userdir settings. However, when Rscript then calls LyX again, it will crash. So Rscript (ie `lyxzip2word`) has to *unset* that envar before calling LyX, which means that the second instance of LyX cannot know about your modules. You were warned!

# 9 FAQ

**Q:** *Why did you use R rather than writing a series of* Pandoc *filters in SDGL to translate FSDOIG format, like you're supposed to?*

**A:** Because I know R, and I have no interest in learning SDGL nor figuring out the intricacies of FSDOIG format. Also, modern R is really good for regexy manipulations--- people often don't realize that.

**Q:** *You clearly had to fix lots of issues. Why didn't you report them as bugs and wait for someone else to sort them out, like a logical systematic person would?*

**A:** Because I needed this for my own purposes, and fairly quickly. Other people are busy, and might not have the same view of what constitutes a "bug" or "something worth fixing" as I do (fair enough). Like people say: "if you want something done properly, do it yourself". That said, I really ought to get round to reporting the bugs at some point--- but reporting bugs is quite time-consuming. If you try exporting "eqntest1.lyx" with LyX's built-in MSWord exporter, you'll get some idea...

**Q:** *Your approach based on counting indents etc is terribly fragile and Offensive To Logic. It is much better to build entire parse trees and then transmogrify the glorts using percortical DeMoivre contextual fermions.*

**A:** Errr, maybe so. But my approach works, and it works *now*, and *I* need it now!

**Q:** *Hmmm, I prefer to write all my documents in raw Latex because it is the Only True Way for ninth-level ninjas like myself; GUIs like LyX are for lesser mortals, monotremes, slime moulds, etc. However, I reluctantly concede that* `lyxport` *does a good job of exporting to MSWord, and I note that it actually starts from a Latex file (as exported by LyX). Can I make use of it on my own Latex files?*

**A:** How very gracious of you. But to answer your question: no, not directly on general Latex files. There might be other tools Out There for that. `lyxport` relies on the tightly-structured flavour of Latex exported by LyX. Your only chance would be to import your Latex into LyX, then export. It *might* all work, but expect pain.

**Q:** *Is this really a FAQ?*

**A:** Do I look like Mickey Sodding Mouse?