

Package: lyxport (via r-universe)

October 11, 2026

Version 1.0.191

Author Mark Bravington <markb2@summerinsouth.net>

Maintainer Mark Bravington <markb2@summerinsouth.net>

Depends R (>= 4.3.0)

Imports mvbutils, Rcpp

LinkingTo Rcpp

Title LyX to MSWord etc

Description Tools for smooth Lyx export via pandoc, currently to MSWord but to potentially other formats

SystemRequirements LyX, pandoc, ImageMagick

Encoding UTF-8

License GPL-2

Collate lyxport.R Cloaders_lyxport.R

Config/pak/sysreqs libmagick++-dev gsfonts

Repository <https://markbravington.r-universe.dev>

Date/Publication 2026-08-09 00:45:42 UTC

RemoteUrl <https://github.com/markbravington/lyxport>

RemoteRef HEAD

RemoteSha 20ff7efe82c6e1939147768139af71ac6db66d62

Contents

lyxport-package	2
lyxprefhack	2
lyxzip2word	7
requote_lyx	10
tex2utf8	11
tidy_initials	12
Index	14

Description

The lyxport R package is for exporting LyX documents 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, appendices, and bibliography. 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 behind-the-scenes fiddly code in order to save you lots of manual post-tinkering.

Once you've installed the package in R, you need to set up a few things in LyX itself, so that you can export directly from LyX. The easiest way is to run [lyxprefhack](#) in R; you need to read its documentation *first* to see what the LyX system requirements are, and to find out what to do if [lyxprefhack](#) doesn't succeed. You can instead do (most of) the setup manually in LyX. After that, you don't need to run R or load the **lyxport** package in order to export to MSWord from LyX; though you might want to use lyxport from R for its other minor features, described below.

When setup is successful, you should see in LyX an "MSWord (lyxport)" option in "File->Export", and a "lyxport" item at the end of "Help->Specific manuals", which contains **full package documentation** (which you may wish to consider at least opening...). To just see the features without, like, having to, like, actually read *documentation*, do "File->Open example" and filter for "lyx", to open "lyxport-demo.lyx". You can try exporting with LyX's built-in MSWord option, and with lyxport. I haven't tested every LyX feature; it's mostly just stuff I need. More things might get added.

Under the hood, the conversion work is done by [lyxzip2word](#), which is normally called by LyX on your behalf. However, if you want to use lyxport to do improved exports to *other* formats besides MSWord, then you might need to experiment with running [lyxzip2word](#) directly from R.

There are also some helper utilities which you can run from R:

- [requote_lyx](#) (qv) to help sort out quotation marks, e.g. in case your document includes imports from other formats such as plain-text.
- [tidy_initials](#) (qv) and [tex2utf8](#) to tweak bibliography files to be MSWord-ready. These are called automatically by [lyxzip2word](#), unless you tell it not to, but you might also find them useful in their own right.

Description

This function should be called just after you (re)install the **lyxport** package (and/or LyX itself), after which you probably won't need to call it again until you update LyX. It tries to modify the "preferences" and "ui/stdmenus.inc" files in your LyX Userdir to add better MSWord export, with shortcut "W" in "File->Export"; see DETAILS. It also creates a LyX help file and an example that you can access straight from LyX.

Before running lyxprefhack, you need to check that the LyX requirements are met, as explained next. Then you should probably close all running LyX instances before running lyxprefhack.

If lyxprefhack works, everything is easy. If it fails, don't panic; you can make the necessary changes to LyX preferences etc yourself, as explained below. It's not possible for me to guess all the quirks of how people have set up LyX on their different OSES, so failure is a possibility.

Before changing your "preferences" and "ui/stdmenus.inc" files, lyxprefhack makes backups, with obvious names. If things get severely broken, you can restore those files yourself manually.

Note: Everything here has been tested on Windows only, though David Miller has provided some helpful and colorful feedback about Linux and Macs. On Macs at least, the precise locations/names of LyX menu items and/or clicky buttons might be a bit different. Seek and ye shall find, maybe.

Requirements: #1 From LyX, you need to check that the following built-in LyX exports work:

- Export to "Lyx Archive" (dot-zip on Windows, dot-tar-dot-gz on Linux/Macs)
- Export to "MS Word Office Open XML" (or very similar name)
- Export to PDF (if this fails, you're in big trouble!)

If you need a file to practise on, "File->Open_example" and filter to "Welcome.lyx". It's best to immediately *save* it somewhere else, before you try to export.

(This is largely a check on other programs needed by LyX export; are they installed correctly, from LyX's PoV? Usually yes, but not always...)

#2 Make sure that LyX has saved a "preferences" file in your Userdir (see "Help->About_LyX" for whereabouts; you can open the folder to check). Experienced LyX users will already have a "preferences", but it *isn't* automatically created in a new LyX setup. If not, you'll have to get LyX to make one. I suggest "Tools->Preferences->Identity", change your name, and click "Apply". You should see a "preferences" file in your Userdir. You can now change your Identity back if you want, and "Apply" again.

(Note: I have tried to automate this step, but so far failed; I suspect LyX bugs. It's not hard to do it yourself, though.)

#3 You also need to make sure that LyX can run R, and vice versa. If the folders with the R and LyX executables are already in your PATH— and they probably are— then you're probably fine. But you still have to check. (If you don't know what "your PATH" means, then it's a Google moment.) Notwithstanding claims that Linux & Mac users never have any problems here, I am a bit skeptical...

From R, you can check with `Sys.which("lyx")` (or possibly "LyX" on Linux/Macs, IDNK); if that comes back non-empty, it's a jolly good sign. If not, you need to add the LyX executable folder to your PATH— either "globally", or automatically when R starts, eg via ".Rprofile" and [Sys.setenv](#)— the details are up to you.

From LyX, I'm not sure how to check directly, but lyxprefhack tries to check for you. If it reports a problem, then in LyX go to "Tools->Preferences->Paths->PATH_prefix" and add the R

executable folder to the end; use forward-slashes, even on Windows. See Appendix C3 of LyX's "User Guide" for mor info. Remember to "Apply" or "OK". And if you know in advance that R *won't* be in your PATH automatically when LyX starts (and there can be sensible reasons for that), then do the same thing.

#4 Not *absolutely* essential, but ImageMagick is needed to make sure graphics come out the right size (as best as I can). On Windows, this is installed automatically along with LyX, so it should be there. IDNK whether the same courtesy is extended on Linux or Macs. If not, best to install it; make sure it's on your PATH, or add its path to "Tools->Preferences->Paths->PATH_prefix" as per #3.

#6 Make sure that you have an editor/viewer for MSWord documents in your PATH, or in "...PATH_prefix" as above. I use LibreOffice, but presumably you could use MSWord itself; details are up to you.

#7 Work out where your LyX Userdir is (eg "Help->About_LyX"; there are other ways for experienced users). It is needed as an argument to lyxprefhack. (However, for advanced users: there is no need to do this if you normally set the userdir via an envar (environment variable), eg "LYX_USERDIR_24x" or "LYX_USERDIR_23x" as per LyX's "Customization" manual, because lyxprefhack will look for that envar automatically.)

This list isn't exhaustive. There might be other LyX features that aren't properly set up on your system (eg bibliographic stuff, exotic Latex packages), but AFAIK if you can do a PDF export for your document-of-choice, then [lyxport](#) should work too.

.DID.IT.WORK?

lyxprefhack will error-stop if various checks fail. If not, it will do a self-test at the end, to see if it can export the example "lyxport-demo.lyx". That might also fail. Or not! You might see some warnings about "'py' is not recognized as an internal or external command, operable program or batch file."; you can safely ignore them, AFAIK.

If lyxprefhack gave you an error message, then no it didn't work, and hopefully the message is informative enough to help. Check the Requirements above, and if they're OK, try making the (few) changes manually in LyX, as per the next section.

If you get a happy message saying that export worked, then you are probably fine! But you should still check for the mannual in LyX, as in the list below.

If things didn't work but you were able to set up LyX manually, there are three things to check:

- Does "Help->Specific_manuals" give you "lyxport" option? Can it open the file? (It's not essential; the same file can be viewed from R, as per ?lyxport. But it's very handy, and you should check this even if lyxprefhack reports success.)
- Does "File->Open_example" (then filter for "lyxport") open up the "lyxport-demo" example?
- Is there an "MSWord (lyport)" option in "File->Export_as"?
- Does it export the "lyxport-demo" successfully?

If you're sure that the Requirements are OK, and you've tried the above including manual setup but still have problems, please email me to let me know what happened (details are good!). I'm not going to say "bug report" because it may not be my fault :) It's not a zero-sum game BTW ; it might not be your fault, either!

Manual setup in lyx: There isn't much, actually, but it is fiddly— hence lyxprefhack! If that fails and you want to know more, read this...

Reading LyX's "Customization" manual, in p'tic sections 3.1 "Formats" and 3.3 "Converters" should help if you get stuck with the descriptions below.

#1 In "Tools->Preferences->File_Handling", you need to add one "File format" and one "Converter". For the former, select the "File Formats" dialog and click "New". You will get a bunch of empty items. Complete them as follows:

Format MSWord (lyxport)

(Tick the boxes for "Document format", "Show in export menu", and "Vector graphics format", in that order.)

Short name wordx

Extensions docx

MIME application/vnd.openxmlformats-officedocument.wordprocessingml.document

Shortcut W

Editor Custom, swriter

Viewer Custom, swriter

Copier <blank>

Default Output Formats:

With TeX fonts PDF (pdflatex)

Japanese PDF (dvipdfm)

With non-TeX fonts PDF (XeTeX)

You might be able to use a different Editor and Viewer, eg MSWord itself— but I don't have that installed on my system, so I use LibreOffice ("swriter") instead. It's good, but not perfect (math is not right). You could look at LyX's built-in format for "MS Word Office Open XML", or whatever it's called, for guidance.

Then "Apply" or "OK".

#2 For the Converter (which you can only create after you've defined the new File Format), you have to have an existing converter selected (doesn't matter which one) and then starting changing things before you get the option to Add— you can't just go from an empty definition. This can seem a bit nerve-wracking because it feels like you might break the existing one if you're not careful. And you might. So don't.

From format LyX Archive (zip)

To format MSWord (lyxport)

Converter Rscript --no-save --no-restore --verbose -e lyxport::lyxzip2word(FROM_LYX=TRUE)
 \$\$i \$\$r \$\$p 1 > docxconv.log 2>&1

Extra flag <blank>

Converter File Cache <I prefer to leave this unchecked, but you might not>

Security:

Forbid used of needauth converters <unchecked>

Use needauth option <check>

Then Add, and OK or Apply.

Now "MSWord (lyxport)" should be available from "File->Export_as" menu. It might even work!

#3 To make visible the LyX helpfile for [lyxport](#), you need to it to LyX's menu system. First, create a folder "doc/" in Userdir. Then copy into it the file "lyxport-docu.lyx" from the "examples/" folder of this R package. Now you need to tell LyX that it's available, via editing a file that lives in the "ui/" folder of Userdir: (on Windows...) the file is "stdmenus.inc". Somewhere in it, there should be a "HELP MENU" with two sub-menus. Add the following line to the end of the "Examples" sub-menu (you can also include a line "Separator" immediately before it):

Item "lyxport" "help-open lyxport-docu"

#4 "lyxport-demo" example comes as 4 files in the "examples" folder of the R package. You can copy them to a folder "examples/lyxport" in your Userdir. I think they should be immediately visible to LyX via "File->Open_example".

Usage

```
lyxprefhack( userdir=NULL, self_test=TRUE)
```

Arguments

userdir	Where your config files live; see section 2 of LyX's "Customization" manual. R will prompt you for it if you don't supply it (and if it also can't be auto-deduced; see below), so you can copy it from "Help->About" in LyX; single backslashes or forward-slashes are OK on Windows.
self_test	set FALSE to omit the self-test, which does take a few seconds (e.g. if you know it will work, or you know it will fail...).

I reckon the best way to run LyX (not just for this), is to set the userdir via an envvar of the form "LYX_USERDIR_24x" (or "...23x" for LyX 2.3, etc), as documented in LyX's "Customization" manual (currently in the section "...multiple configurations", though I only use one config). That means file associations etc all use exactly with the same LyX setup as if you launch LyX directly. If you have set this up already, then leave userdir blank.

Value

As described in DID.IT.WORK and MANUAL.SETUP.IN.LYX:

- Overwrites the file "<userdir>/preferences" ((unless theres no change required). The old "preferences" will be backed up to "old_preferences<N>" (guaranteed not to overwrite any existing backup).
- Adds a "lyxport" option to "Help->Specific manuals", in the file "<userdir>/ui/stdmenus.inc", again making a backup of the latter if any change was needed. If "stdmenus.inc" did not exist in "userdir/ui", then its copied there from (an attempt to find...) LyXs system dir.
- Copies one file from the R installation to LyXs "<userdir>/doc" ("lyxport-docu.lyx") and one set of files to LyXs "<userdir>/examples/lyxport" ("lyxport-demo.lyx" and associated files).

The actual R return-value is probably a string such as "LyX archive ...zip created successfully" (which is good, you can ignore it) or some error output if the self-test failed. If self_test=FALSE you should get 0, which signifies nothing :)

Version proofing: There is a petageek-level subtlety with PATH prefix and R versions, which I will mention here "for completeness"—partly because I find it generally useful, not just for [lyxport](#).

R often needs updating; I don't want to have to remember each time which obscure LyX features I need to also update—nor indeed in lots of other bits of software which might occasionally want to call R. It would be nice if I could set an envvar(s) that holds the current R path (well, I can and I do do that), and ask LyX to use the envvar instead of some absolute path, but that second part doesn't currently work. Instead, I have set up (in Windows) a symlink folder for R whose nominal path is always fixed, and I use that as the PATH_prefix in LyX (and other places, eg the start-menu item for R). Then, when R is updated, I just run a little command-shell script that updates the symlink. I use a similar trick for LyX itself, obvs not to tell LyX about where it lives, but rather to be able to refer to the current LyX consistently from other pieces of software. I did say "petageek"—you were warned!

See Also

[lyxzip2word](#), [lyxport](#)

Examples

```
## Not run:
lyxprefhack()

## End(Not run)
```

lyxzip2word

Convert LyX to MSWord or other format

Description

lyxzip2word starts from a "LyX archive" (zip export from LyX) and converts to MS Word (or potentially other formats) using various tools, mostly Pandoc. See `RShowDoc("lyxport-docu", package="lyxport")` for more information about normal use from LyX, and any requirements of your LyX source file (eg specifying the bibliography format).

You don't normally need to know anything about this function, since it is called automatically from LyX. However, I should document it for maintenance-type reasons. Also, if you want to experiment with exporting to other formats, you might want to use it direct from R, setting the `outext` and `panoutopts` arguments.

Usage

```
lyxzip2word(
  zipfile,
  outext= 'docx',
  panoutopts= outext,
  origdir= dirname( zipfile),
  tempdir= base::tempdir(),
```

```

copy= FALSE,
FROM_LYX= !interactive(),
refdir= NULL,
lyxdir= NULL,
lyx_userdir= NULL,
natnum_pandoc= FALSE, # devil or deep-blue-sea?
crossref_pandoc= TRUE,
extra_filters= TRUE,
verbose= FALSE,
dbglyx= ''
)

```

Arguments

zipfile	Name of input file, normally a LyX-zip archive with a path. Extension is optional, but ".zip" is assumed on Windows, and ".tar.gz" on Unix-style. If (for experimentation only) the extension is ".lyx", then all the other necessary files had better be in tempdir, which had better be the same folder as the in-fact-not-zippy zipfile.
outext	File extension of output.
panoutopts	For pandoc's writer, to tell it what kind of output to produce, ie pandoc's "-t" argument. Normally the default of outext is fine, but there might be other useful options for specific writers, eg "markdown+myfilter-theirfilter".
origdir	where zipfile lives, and where the output should go; deduced from zipfile if missing.
tempdir	where to unpack the zipfile and create temporary files etc.
copy	whether to copy zipfile to tempdir (leaving the original in place), or <i>move</i> it (in which case the original is "lost"). Normally it's fine to just move it, because the zipfile is only a temporary step in the creation of your magnificent MSWord-flavoured document...
FROM_LYX	set TRUE iff called from inside LyX by a Converter, in which case zipfile, origdir, tempdir, and possibly dbglyx will be set according to commandArgs .
refdir	Top of the folder-tree for bibliography-finding. It should have a folder ./bibtex/bib/ underneath it, containing dot-bib files. It is normally found by automatic magic.
lyxdir	Normally fine to leave blank. If not, it should be path of the "Resources" folder used by LyX itself. Default is to deduce it from the location of the LyX executable, found via Sys.which . NB if lyxdir is supplied and the LyX executable *isn't on the search path— which it should be— then an attempt is made to find the LyX executable relative to lyxdir; that may not be sensible.
lyx_userdir	what you'd pass in the "-u <userdir>" option when starting LyX. However, as of Lyx 2.4.2.1, there's a bug which stops that working (in the particular context of lyxzip2word). What this means is, you can't use your own private layouts/modules in your own userdir tree; you'd have to copy them to Lyx's system layouts folder.
natnum_pandoc	whether to turn on pandoc's own numbering scheme (when reading and when writing). More trouble than it's worth so far, hence the default is FALSE.

<code>crossref_pandoc</code>	whether to use the "pandoc-crossref" filter when reading the Latex source. The default is TRUE, but lyxzip2word actually has to spend quite a bit of time fixing incorrect xrefs, and I suspect I could get FALSE to work with just minor mods, leading to one less "dependency". But everything's fine at the moment, so leave well enough alone.
<code>extra_filters</code>	which addition Lua filters to run. Default TRUE means use all Lua files in the installed package's "lua-filters" folder; as of v1.0.188, that was just "page-break.lua", but I may add others and you can manually copy your own faves in there too. FALSE or NULL means no filters; otherwise, a vector of file names (with full path). Since the LyX invoker has (v1.0.188 anyway) no way to set this parameter, you have to put your filters into the aforementioned folder if you want to do this directly from LyX.
<code>verbose</code>	if TRUE, then generate two logfiles, "pandoc.log" and "docxconv.log", in <code>tempdir</code> , for the latex-to-pandoc and pandoc-to-docx conversions respectively.
<code>dbglyx</code>	Only for debugging, obvs. Should be blank, or a positive integer as per "lyx -dbg". Any number will also cause R to print out various things, such as paths.

Details

The steps in the conversion process are (actually there's more than this, this list is out-of-date...):

- Move & extract zip file
- Generate Tex, by running "lyx -export"
- Merge any input/include files
- Add eqn labels: `eqn_labels_for_word()`
- Check and prepare bibliography
- Twiddle any appendices, so as to not confuse pandoc
- Export Tex -> pandoc-native: pandoc
- Fix labelled-eqn column widths: `eqalignfix()`
- Perhaps move the bibliography to before appendices
- Export pandoc-native -> docx: pandoc

Value

Should produce a file "<zipfilename>.<outext>" in folder `origdir`. There will also be various files in `tempdir` (which will be LyX's session `tempdir`, if this was invoked from LyX itself), including logfiles if `verbose=TRUE`. Look carefully thru LyX's "View->Messages" window to see where that LyX `tempdir` is; it changes from one LyX session to the next. The formal R return-value of `lyxzip2word` is TRUE or FALSE according as whether it thinks everything worked.

See Also

[lyxport](#)

Examples

```
## Not run:
# In LyX, open the "lyxport-demo.lyx" example, then...
# ... File->Export->Lyxzip. Then try exporting...
# to a non-MSWord format, via eg...
lyxzip2word( 'lyxport-demo.zip', outtext='html')

## End(Not run)
```

requote_lyx

Clean up LyX quotation marks

Description

This function *tries* to sensibly turn assorted types of quotation marks in a LyX document into LyX "dynamic quotes". The latter make it easy to render the document with any of a number of defined "nationalesque" quotation schemes, just by tweaking a single item in Document->Settings->Language. For example, you can get outer double/inner single quotes, outer single/inner double, guillemets, and so on; see "Quotation marks" in the LyX UserGuide (currently section 3.9.4.2). Without this functionality, a large LyX document can end up having multiple types of quotation marks (especially if it is multi-authored or includes excerpts of other documents), which can't easily be changed or searched for or made coherent.

Not all quotation marks should be changed: for example, straight quotes within listings or ERT should be left alone. `requote_lyx` tries to get that right. However, many funny-looking things come out OK when exported to Latex (which is a necessary step in producing eg nice MSWord documents, using the other functions in this package).

`requote_lyx` mainly aims at double-quotes (since IME these are the commonest defaults for normal quotation), but does some single-quote stuff too:

- Apostrophes are left alone (deliberately; they are tricky!)
- Any explicit single quotes are made dynamic, but their singleness is kept; it's assumed to be deliberate.
- Hard-wired directional single-quote characters are turned into dynamic *double* quotes, just like hard-wired directional double-quotes. Coz that was *probably* the intention of an author who just prefers single-quotes for outer.

`requote_lyx` isn't aiming at perfection, and may well not be foolproof; there might be situations where it doesn't work properly, because LyX used some structuring that I hadn't anticipated. Sorry.

Usage

```
requote_lyx( filename = NULL, lyx = NULL, outfile = NULL)
```

Arguments

<code>filename</code>	optional name of file to read from
<code>lyx</code>	or you can pass the actual text in directly, as a character vector
<code>outfile</code>	optional filename to write the output to.

Value

The modified LyX text will be returned, invisibly. Also, if `outfile` is not `NULL`, the modified LyX text will be written to `outfile`.

tex2utf8

Transform Biblatex bibliography to native UTF8 characters

Description

Biblatex bibliography (dot-bib) files may contain legacy Latex/Bibtex representations of characters, such as `"\c{G}"` for "g". These are normally fine— though some of them are technically incorrect, but may still work— but not always, e.g. when looking for consistent names, as in [tidy_initials](#) (qv). So you can *try* `tex2utf8` to translate such representations into "native UTF8 codepoints". It might help you.

This might even work on more general Latex (ie not on a bib file) but you are on your own there...

Usage

```
tex2utf8(tex, file = NULL, outfile = NULL, debrace = FALSE)
```

Arguments

<code>tex</code>	character vector containing the bibliography contents. Provide just one of <code>tex</code> or <code>file</code> .
<code>file</code>	if supplied, this is used in place of <code>tex</code> (which is ignored), to read the lines in from.
<code>outfile</code>	if supplied, the result will be written here.
<code>debrace</code>	whether to remove superfluous braces around single UTF8 characters. IME these are mostly legacy effects of Latex representation, rather than deliberate statements about upper/lower case (the only legit use I can think of). Extra braces are usually harmless if you are using <code>tex2utf8</code> as a standalone, but they do muck up <code>tidy_initials</code> (qv), so lyxzip2word sets this option <code>TRUE</code> when tidying up the biblio (unless I have given an option to change that, which I haven't yet).

Details

This is harder than it sounds. The key was to find a couple of tables on WWW; see source code for details. `tex2utf8` tries to fix up common Bibtex representation errors (i.e. semi-problems in my own master biblio file, mostly from WWW sources), but probably won't catch everything. And there may be "native UTF8 codepoints" for some characters that aren't in the (original?) Latex list, and won't be transformed. They can of course be produced in Latex by a composite (eg `"\k{n}"`; I have no idea whether that is a real character in some alphabet). They will stay composite in the output.

Value

The modified contents.

See Also

lyzip2word, [tidy_initials](#)

Examples

```
# Not compulsory to have an EXAMPLES -- you can put examples into other sections.
# Here's how to make a "don't run" example:
## Not run:
reformat.my.hard.drive()

## End(Not run)
```

tidy_initials

Consistent author names in dot-bib files

Description

In a dot-bib file, the same author can appear with slightly different names in different papers: "A. Psmith", "Alan Psmith", "Alan B. Psmith", "A. Bertram Psmith", and so on. If you are not careful, your citations can come out funny as a result. For example, you might see "Psmith et al. (1999)" but "A.B. Psmith (2004)" even though Alan Bertram is the only Psmith you are citing. With Biblatex and PDFs, you can suppress such nonsense via "uniquename=false" and "uniquelist=false". But with CSL and MSWord etc, it seems to be harder— well, so does pretty much everything, actually. To circumvent the problem, you can call `tidy_initials` on your bib file beforehand (which [lyxzip2word](#) does automatically by default, on a temporary copy of the dot-bib, unless you ask it not to). This will ensure all plausibly-identical authors have exactly the same bib entries (initials only, and the longest possible set), and should eliminate silly citations.

The merging rules, which depend slightly on the `gungho` argument, are as follows:

- "Alan Psmith" and "A. Psmith" are always assumed to be the same (and "Alan Psmith" will be remapped to "A. Psmith", since only initials are retained when there's a discrepancy).
- "Alan Psmith" and "Alicia Psmith" are never merged. That is: if there's a full name rather than initial, it's taken seriously.
- Mismatched initials are never merged, eg "A.B. Psmith" and "A.C. Psmith".
- If `gungho=TRUE`, then missing initials are ignored, so that "A. Psmith" gets merged with "A.B. Psmith". If there's also an "A.C. Psmith", then it's the luck-of-the draw as which one "A. Psmith" will be merged with.
- If `gungho=FALSE`, then merging only occurs when two authors also have the same number of initials; so "Alan Bertram Psmith" gets merged with "A.B. Psmith" but not with "A. Bertram Psmith".

In order to make this surprisingly tricky programming task a bit easier, `tidy_initials` calls [tex2utf8](#), which you might find useful in its own right.

Usage

```
tidy_initials( bib, file=NULL, outfile=NULL, gungho=TRUE)
```

Arguments

<code>bib</code>	character vector containing the bibliography contents. Use one of <code>bib</code> or <code>file</code> .
<code>file</code>	if supplied, this is used in place of <code>tex</code> (which is then ignored), to read the lines in from.
<code>outfile</code>	if supplied, the result will be written here.
<code>gungho</code>	TRUE if you prefer to assume that middle initials are easy-come-easy-go. Saves pain, but in rare cases could merge genuinely different people.

Value

The modified contents, as a character vector. If `outfile` is supplied, that file will be created.

See Also

[tex2utf8](#), [lyxzip2word](#)

Examples

```
# Not compulsory to have an EXAMPLES -- you can put examples into other sections.
# Here's how to make a "don't run" example:
## Not run:
reformat.my.hard.drive()

## End(Not run)
```

Index

* misc

- lyxport-package, [2](#)
- lyxprefhack, [2](#)
- lyxzip2word, [7](#)
- requote_lyx, [10](#)
- tex2utf8, [11](#)
- tidy_initials, [12](#)

commandArgs, [8](#)

lyxport, [4](#), [5](#), [7](#), [9](#)
lyxport (lyxport-package), [2](#)
lyxport-package, [2](#)
lyxprefhack, [2](#), [2](#)
lyxzip2word, [2](#), [7](#), [7](#), [11–13](#)

requote_lyx, [2](#), [10](#)

Sys.setenv, [3](#)

Sys.which, [8](#)

tex2utf8, [2](#), [11](#), [12](#), [13](#)

tidy_initials, [2](#), [11](#), [12](#), [12](#)