Package: convertxlsform (via r-universe)

October 21, 2026

Type Package

Title Convert an 'XLSForm' to a 'Word' or PDF Document

Version 0.3.0

Description Converts 'XLSForm' questionnaires, as used by 'Open Data Kit' ('ODK') and 'KoboToolbox', into formatted paper versions for interviewing and review. See <https://xlsform.org> for the 'XLSForm' standard. Output is a 'Word' document ('.docx') or, optionally, a PDF produced from that 'Word' document with 'LibreOffice'. Supports multilingual forms, nested question numbering, skip logic display, and question and choice media.

License MIT + file LICENSE

URL https://github.com/masud90/convertxlsform

BugReports https://github.com/masud90/convertxlsform/issues

Encoding UTF-8

Language en-US

Depends R (>= 4.1)

Imports cli, flextable (>= 0.9.5), magrittr, officer (>= 0.6.9), readxl, rlang, stats, tools, utils, zip

Suggests doconv, knitr, rmarkdown, testthat (>= 3.0.0), withr, writexl

SystemRequirements 'LibreOffice' (optional, for PDF output)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Masud Rahman [aut, cre, cph]

Maintainer Masud Rahman <write2masud+convertxlsform@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-21 19:27:31 UTC

RemoteUrl https://github.com/cran/convertxlsform

RemoteRef HEAD

RemoteSha c27fbdb578a3997b2be33d2de599cd0959af6e5a

Contents

cli 2
convertxlsform 4
docx_to_pdf 6
has_libreoffice 7
xlsform_languages 7
xlsform_outline 8

Index **10**

`cli` *Convert an XLSForm from the terminal*

Description

Runs `convertxlsform()` with options taken from the command line. Use it from a shell without opening R:

Usage

```
cli(args = commandArgs(trailingOnly = TRUE))
```

Arguments

`args`	Character vector of arguments. Defaults to the trailing command-line arguments.

Details

```
Rscript -e 'convertxlsform::cli()' form.xlsx --format pdf --lang bn
```

Put the XLSForm path and the options after the quoted expression. If a script or CI job needs the exit status, use `Rscript -e 'quit(status = convertxlsform::cli())' form.xlsx`. The plain form above also works for normal use.

This function is a terminal interface. Its help and version text go to standard output, and usage errors go to standard error, because that output is its purpose. Progress messages are sent with `message()` and `--quiet` suppresses them (the same as `verbose = FALSE` in `convertxlsform()`).

Value

The exit status invisibly: 0 on success, 1 on a usage error or a failed conversion. The function does not quit R.

Usage

`cli(args)`: `form.xlsx [options]`. Options accept `--name value` and `--name=value`.

- `-o`, `--output PATH`: output file or folder (default: next to the form).
- `-f`, `--format docx|pdf|both`: output format (default: from the extension of `--output`, else `docx`).
- `-l`, `--lang LANG`: language code, name or header suffix.
- `--flat-numbers`: number questions 1, 2, 3 instead of 1.1, 1.2.
- `--no-relevants`: hide skip logic.
- `--no-constraints`: hide constraints.
- `--no-hints`: hide hints.
- `--no-choice-codes`: hide choice codes.
- `--show-names`: show variable names.
- `--include-hidden`: include calculate, start, end and similar rows.
- `--media-root DIR`: folder for media and external files (default: the folder of the form).
- `--title TEXT`: form title.
- `--page-size A4|letter` (default `A4`).
- `--landscape`: landscape orientation.
- `--margins IN`: margins in inches (default 1).
- `--font NAME` (default `Arial`) and `--font-size PT` (default 11).
- `--template FILE`: reference `.docx` for styles.
- `--max-choices N`: longest choice list printed in full (default 30).
- `--repeat-copies N`: copies of each repeat group (default 1).
- `--overwrite`: replace existing output.
- `--quiet`: no progress messages.
- `-h`, `--help`: show the help and stop.
- `--version`: show the package version and stop.

See Also

`convertxlsform()`

Examples

```
path <- system.file("extdata", "minimal", "minimal.xlsx",
                    package = "convertxlsform")
cli(c(path, "-o", tempdir(), "--overwrite", "--quiet"))
```

`convertxlsform` *Convert an XLSForm to a Word or PDF document*

Description

Reads an XLSForm (as used by Open Data Kit and KoboToolbox) and writes a formatted paper version for interviewing and review. The default output is a Word document next to the input file. A PDF is made by converting the Word document with LibreOffice (or Microsoft Word through the doconv package); see `docx_to_pdf()` and `has_libreoffice()`.

Usage

```
convertxlsform(xlsform, output = NULL, format = NULL, lang = NULL, ...)
```

Arguments

`xlsform`	Path to an XLSForm (`.xlsx`, `.xlsm` or `.xls`).
`output`	Output file or folder, see Details. `NULL` (default) writes next to `xlsform`.
`format`	`"docx"`, `"pdf"`, `"both"` or `c("docx", "pdf")`. `NULL` (default) takes the format from the extension of `output`, and is `"docx"` when there is no such extension.
`lang`	Language to use: a code (`"fr"`), a name (`"French"`) or the full header suffix (`"French (fr)"`). `NULL` (default) uses the `default_language` of the settings sheet, or the first language found. See `xlsform_languages()`.
`...`	Further options, all named. Defaults are in brackets.

- `nested_serial_numbers` (`TRUE`): number questions inside groups as 1.1, 1.2; `FALSE` gives flat numbers 1, 2, 3.
- `include_relevants` (`TRUE`): show skip logic ("Ask if: ...").
- `include_constraints` (`TRUE`): show constraints.
- `include_hints` (`TRUE`): show hints and guidance hints.
- `include_choice_name` (`TRUE`): show the choice code after each choice label.
- `show_variable_names` (`FALSE`): show the variable name of each question.
- `include_hidden` (`FALSE`): include hidden types such as `calculate`, `start` and `end`.
- `media_root` (folder of `xlsform`): where images, audio, video and external choice files are looked up.
- `title` (`NULL`): form title; overrides `form_title` of the settings sheet.
- `page_size` (`"A4"`): `"A4"` or `"letter"`.
- `orientation` (`"portrait"`): `"portrait"` or `"landscape"`.
- `margins_in` (1): page margins in inches, above 0 and at most 2.
- `font_family` (`"Arial"`): font for the document.
- `font_size` (11): base font size in points, 6 to 20.
- `template` (`NULL`): a reference `.docx` whose styles are used.

- max_choices (30): choice lists longer than this get a code box layout instead of a printed list.
- strings (NULL): named list overriding connective phrases such as "Ask if".
- repeat_copies (1): number of blank copies printed for each repeat group, 1 to 20.
- font_family_cs (NULL): font for complex scripts; chosen from the language when NULL.
- overwrite (FALSE): replace existing output files.
- verbose (TRUE): print progress messages.

Value

The path or paths of the files written, as a character vector, invisibly.

Where the file goes

- output = NULL: same folder and base name as xlsform, with the new extension.
- output is an existing folder: <base name>.<ext> inside it.
- output is a file path: used as given. Its extension (.docx or .pdf) sets format when format is NULL, and must agree with format otherwise.
- Two formats (format = c("docx", "pdf")) write two files with the same base name. With a file path in output, its extension is replaced.

An existing file is never replaced unless overwrite = TRUE.

See Also

xlsform_outline() to inspect a form before converting it, docx_to_pdf(), and cli() for the terminal.

Examples

```
path <- system.file("extdata", "minimal", "minimal.xlsx",
                    package = "convertxlsform")
out <- file.path(tempdir(), "minimal.docx")
convertxlsform(path, output = out, verbose = FALSE, overwrite = TRUE)


# A multilingual form: another language, flat question numbers, US letter
path_ml <- system.file("extdata", "multilang", "multilang.xlsx",
                       package = "convertxlsform")
convertxlsform(path_ml, output = tempdir(), lang = "bn",
               nested_serial_numbers = FALSE, page_size = "letter",
               verbose = FALSE, overwrite = TRUE)

# PDF needs LibreOffice
if (interactive() && has_libreoffice()) {
  convertxlsform(path, output = tempdir(), format = "pdf",
                 verbose = FALSE, overwrite = TRUE)
}
```

docx_to_pdf *Convert a Word document to PDF*

Description

Converts a .docx file to PDF with LibreOffice, or with Microsoft Word through the doconv package. LibreOffice is free and runs on all platforms. Each call uses a fresh temporary LibreOffice profile, so it works while a LibreOffice window is open and in parallel runs.

Usage

```
docx_to_pdf(
  docx,
  pdf = NULL,
  engine = c("auto", "libreoffice", "word"),
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

docx	Path to an existing .docx file.
pdf	Path of the PDF to write. NULL (default) uses the path of docx with the extension changed to .pdf.
engine	"auto" (default) tries LibreOffice, then Word through doconv if that package is installed. "libreoffice" and "word" force one engine.
overwrite	Overwrite an existing pdf? Default FALSE.
verbose	Print progress messages? Default TRUE.

Details

If LibreOffice is installed in an unusual place, set the option convertxlsform.soffice or the environment variable CONVERTXLSFORM_SOFFICE to the path of the soffice binary.

Value

The path of the PDF, invisibly.

Examples

```
if (interactive() && has_libreoffice()) {
  docx <- tempfile(fileext = ".docx")
  doc <- officer::read_docx()
  doc <- officer::body_add_par(doc, "Hello")
  print(doc, target = docx)
  pdf <- docx_to_pdf(docx, pdf = tempfile(fileext = ".pdf"))
  file.exists(pdf)
}
```

has_libreoffice *Is LibreOffice available?*

Description

Reports whether a LibreOffice binary can be found, which is needed to convert Word documents to PDF with `docx_to_pdf()`. The search checks the option `convertxlsform.soffice`, the environment variable `CONVERTXLSFORM_SOFFICE`, the `PATH`, and the default install locations for the operating system.

Usage

```
has_libreoffice()
```

Value

`TRUE` or `FALSE`.

Examples

```
has_libreoffice()
```

xlsform_languages *List the languages of an XLSForm*

Description

Finds the languages used in the `label::<language>` style columns of the survey and choices sheets, for example `label::English (en)`, `label::en` or `label::English`.

Usage

```
xlsform_languages(xlsform)
```

Arguments

`xlsform`	Path to an XLSForm (`.xlsx` or `.xls`).

Value

A data.frame with one row per language and columns `suffix` (the text after `::`), `code`, `name` and `default` (TRUE for the language used when no `lang` is requested: the `default_language` of the settings sheet if it matches, otherwise the first language). The data.frame has no rows for a single-language form.

Examples

```
path <- system.file("extdata", "multilang", "multilang.xlsx",
                    package = "convertxlsform")
xlsform_languages(path)
```

`xlsform_outline` *Outline of an XLSForm*

Description

Parses an XLSForm and returns one row for each item that would appear on the paper form: sections, questions, notes and the end of repeat groups. It shows the question numbers, the answer type, the number of choices, and the skip logic and constraints in plain language, so a form can be checked before it is converted.

Usage

```
xlsform_outline(xlsform, lang = NULL, ...)
```

Arguments

`xlsform`	Path to an XLSForm (`.xlsx` or `.xls`).
`lang`	Language to use: a code (`"fr"`), a name (`"French"`) or the full header suffix (`"French (fr)"`). `NULL` uses the `default_language` of the settings sheet, or the first language found. See `xlsform_languages()`.
`...`	Further options: `nested_serial_numbers`, `include_relevants`, `include_constraints`, `include_hints`, `include_hidden`, `media_root`, `max_choices` (lists longer than this get a code box layout), `strings`, `show_variable_names`, `title` and the other conversion options. Page and font options are accepted and have no effect here.

Value

A data.frame with one row per block and columns `row` (Excel row of the form), `serial` (question number, NA when unnumbered), `depth` (number of enclosing groups), `block` (`"section"`, `"section_end"`, `"note"` or `"question"`), `kind` (the question kind, for example `"select_one"`, or `"begin_group"`, `"begin_repeat"`, `"end_repeat"`, `"note"`), `name`, `label`, `required`, `n_choices` (all choices of a select or rank question, not counting an "other" row), `relevant_text`, `constraint_text` (the constraint message when the form has one, else the humanized expression) and `layout` (for select and rank questions: `"list"`, `"columns"` or `"code_box"`, from the number of choices and `max_choices`; NA otherwise). A label may contain line breaks (`\n`).

Examples

```
path <- system.file("extdata", "minimal", "minimal.xlsx",
                    package = "convertxlsform")
out <- xlsform_outline(path)
out[, c("serial", "block", "kind", "name", "relevant_text")]


# A multilingual form: another language, with flat question numbers
path_ml <- system.file("extdata", "multilang", "multilang.xlsx",
                       package = "convertxlsform")
out <- xlsform_outline(path_ml, lang = "bn", nested_serial_numbers = FALSE)
head(out[, c("serial", "name")], 5)
```

Index

cli, 2
cli(), *5*
convertxlsform, 4
convertxlsform(), *2, 3*

docx_to_pdf, 6
docx_to_pdf(), *4, 5*, *7*

has_libreoffice, 7
has_libreoffice(), *4*

message(), *2*

xlsform_languages, 7
xlsform_languages(), *4*, *8*
xlsform_outline, 8
xlsform_outline(), *5*