

# Package ‘lssdoc’

June 15, 2026

**Title** Render Multilingual Questionnaires from 'LimeSurvey' '.lss'  
Files

**Version** 0.1.0

**Description** Render 'LimeSurvey' '.lss' survey exports as multilingual questionnaire documents in Word ('.docx') or PDF, displaying up to four languages side by side with localized chrome in English, French, German, Spanish and Italian. Includes a rule-based automated audit that flags missing translations, forward filter references, duplicate codes, array-scale inconsistencies and orphan structural references. Designed for anyone working with a 'LimeSurvey' survey: researchers, methodologists, ethics committees, translators and reviewers. Processing is fully local: the source file is the only input and no questionnaire content is uploaded to a third-party service.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en-US

**Depends** R (>= 4.1)

**Imports** cli, rlang, xml2

**Suggests** flextable, knitr, officer, rmarkdown, testthat (>= 3.0.0),  
zip

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**URL** <https://amaltawfik.github.io/lssdoc/>,  
<https://github.com/amaltawfik/lssdoc>

**BugReports** <https://github.com/amaltawfik/lssdoc/issues>

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** no

**Author** Amal Tawfik [aut, cre, cph] (ORCID:  
<<https://orcid.org/0009-0006-2422-1555>>, ROR:  
<<https://ror.org/04j47fz63>>)

**Maintainer** Amal Tawfik <amal.tawfik@hesav.ch>

**Repository** CRAN

**Date/Publication** 2026-06-15 12:10:08 UTC

## Contents

audit_lss . . . . .	2
read_lss . . . . .	3
render_audit . . . . .	4
render_questionnaire . . . . .	6
<b>Index</b>	<b>13</b>

---

audit_lss	<i>Audit a LimeSurvey survey for reviewable anomalies</i>
-----------	---

---

## Description

Inspect a LimeSurvey survey and flag anomalies that can be detected without any AI. The audit guides a human reviewer; it does not silently correct anything. Every finding names a precise location and a severity.

## Usage

```
audit_lss(input)
```

## Arguments

input	Either a path to a .lss file (character string) or a pre-parsed lss object returned by <code>read_lss()</code> . Passing a path parses it on the fly; passing an lss object avoids re-parsing when the same survey is also rendered in the same session.
-------	--

## Details

Checks performed:

- **Missing translations** – a question, help, answer, or subquestion text present in at least one language but empty in another.
- **Empty in all languages** – a translatable text empty in every language.
- **Duplicate codes** – a question variable code repeated in the survey, or an answer/subquestion code repeated within one question.
- **Whitespace in codes** – a question, subquestion or answer code containing leading, trailing or interior whitespace (likely a typo; causes subtle bugs in the data export).
- **Missing options for the type** – a question whose type requires answer options or subquestions but has none (per the type taxonomy).

- **Forward filter references** – a relevance expression that names a variable appearing at or after the filtered question (the value is not yet collected when the filter is evaluated).
- **Array-scale inconsistencies** – an array (single or dual) whose subquestions reference a `scale_id` that has no answer options, or vice versa.
- **Orphan references** – a subquestion or answer pointing to a question that does not exist.

### Value

An object of class `lss_audit`: a list with `file`, `languages`, `summary counts`, and a `findings` data frame (`severity`, `check`, `location`, `language`, `message`). It has a `print()` method and an `as.data.frame()` method.

### See Also

[render\\_audit\(\)](#) to write the same findings to a Word or PDF document.

### Examples

```
# A deliberately flawed demo survey ships with the package, seeded
# with every anomaly the audit detects.
demo <- system.file("extdata", "audit_demo.lss", package = "lssdoc")
audit_lss(demo)
```

---

read\_lss

*Read a LimeSurvey .lss file*

---

### Description

Read a LimeSurvey survey structure export (`.lss`, an XML file) and turn it into a structured `lss` object that the rest of the package can audit ([audit\\_lss\(\)](#)) and render ([render\\_questionnaire\(\)](#), [render\\_audit\(\)](#)). Parsing is fully local: the file is never uploaded anywhere.

### Usage

```
read_lss(file)
```

### Arguments

`file` Character. Path to a `.lss` file. Must be a single string pointing to an existing file, otherwise a classed error is raised (`lssdoc_bad_path`, `lssdoc_file_not_found`).

### Details

The `.lss` format is a LimeSurvey XML export. Since DBVersion 4xx/7xx the translatable text lives in dedicated localization sections (`*_l10ns`), keyed by language, while the structural sections hold identifiers and settings. `read_lss()` reads every section into a tidy data frame without mutating any user-facing identifier or text. A field that is present but empty (e.g. `<help/>`) is read as `""`; a field that is absent from a row is read as `NA`.

**Value**

An object of class `lss`: a list with the survey languages, metadata, and one data frame per `.lss` section. Structural sections (surveys, groups, questions, subquestions, answers, question\_attributes, conditions) stay separate from the localized text sections (survey\_language\_settings, group\_l10ns, question\_l10ns, answer\_l10ns), which carry the per-language titles, labels, and help texts. All values are read verbatim as character.

**Examples**

```
# A synthetic four-language demo survey ships with the package.
demo <- system.file("extdata", "demo_survey.lss", package = "lssdoc")
lss <- read_lss(demo)
lss$languages
```

---

render\_audit

*Render the audit as a focused Word or PDF document*


---

**Description**

Build a short, action-oriented document containing only the audit findings: the same cover page as the full questionnaire document, summary counts, then one table per severity (errors, warnings, notes) listing every finding with its location and message. Use it for QA follow-up or to share issues with a colleague without distributing the full questionnaire.

**Usage**

```
render_audit(
  input,
  output,
  languages = NULL,
  logo = NULL,
  logo_width = 1.5,
  logo_height = 0.75,
  font = NULL,
  font_code = NULL,
  colors = NULL,
  authors = NULL,
  description = NULL,
  chrome_lang = NULL
)
```

**Arguments**

<code>input</code>	Either a path to a <code>.lss</code> file (character string) or a pre-parsed <code>lss</code> object returned by <code>read_lss()</code> . Passing a path parses it on the fly; passing an <code>lss</code> object avoids re-parsing in a workflow that already inspected the audit.
--------------------	--

output	Character. Path to the file to create. The extension determines the output format: .docx writes a Word document directly, .pdf writes a Word document into a temporary location and converts it locally via LibreOffice (or Word, on Windows). Any other extension is rejected with <code>lssdoc_bad_output_ext</code> .
languages	Character vector of language codes used on the cover page. NULL (default) keeps all languages of the survey in their declared order.
logo	Optional path (character) to a PNG or JPEG image displayed at the top of the cover page. NULL (default) keeps the cover logo-free.
logo_width, logo_height	Image dimensions in inches. Defaults 1.5 and 0.75, tuned to a 2:1 logo. Resize or pre-crop your image to fit a different aspect ratio.
font	Optional body font name (character). NULL (default) keeps Calibri. See <a href="#">render_questionnaire()</a> for guidance on overrides.
font_code	Optional monospace font (character) used for code-like content (variable codes, raw expressions). NULL (default) keeps Consolas.
colors	Optional named list of hex color overrides for the editorial petrol-blue palette. NULL (default) keeps the package palette intact. Same shape and accepted names as in <a href="#">render_questionnaire()</a> .
authors, description	Optional cover-page credit block (authors) and free-form note (description). NULL (default) for both. Same shapes as in <a href="#">render_questionnaire()</a> .
chrome_lang	Language used for the document chrome (column headers, row labels, audit section). One of "en", "fr", "de", "es", "it". NULL (default) follows <code>languages[1]</code> when supported, otherwise falls back to "en".

**Value**

The output path, invisibly.

**See Also**

[audit\\_lss\(\)](#) to inspect the same findings in the console; [render\\_questionnaire\(\)](#) for the full questionnaire document.

**Examples**

```
## Not run:
# One-shot (path -> .docx)
render_audit(
  system.file("extdata", "demo_survey.lss",
             package = "lssdoc"),
  tempfile(fileext = ".docx")
)

# PDF output -- same call, just pass a .pdf path
render_audit("survey.lss", "qa.pdf")

## End(Not run)
```

---

render\_questionnaire    *Render a LimeSurvey questionnaire to a Word or PDF document*

---

## Description

Build a professional questionnaire document from a LimeSurvey survey, displaying up to four languages side by side. Each question becomes a compact flextable with a meta header (variable code, type, mandatory, filter) shown once, language column headers, the question text per language, and the subquestion or answer-option rows underneath – codes on the left, labels per language on the right. Headings, a metadata cover page, an optional table of contents, and an optional audit summary tie the document together. Rendering uses the suggested packages **officer** and **flextable**; both must be installed.

## Usage

```
render_questionnaire(
  input,
  output,
  languages = NULL,
  template = c("cards", "table"),
  layout = c("auto", "side-by-side", "stacked"),
  show_audit = TRUE,
  show_help = TRUE,
  show_attrs = c("prefix", "suffix", "other_replace_text", "validation"),
  show_technical_attrs = FALSE,
  page_format = c("auto", "A4-portrait", "A4-landscape", "A3"),
  show_toc = TRUE,
  show_index = TRUE,
  show_quotas = TRUE,
  show_header_title = TRUE,
  show_source = TRUE,
  show_item_heading = FALSE,
  show_raw_filter = FALSE,
  show_groups = TRUE,
  show_welcome = TRUE,
  show_endtext = TRUE,
  show_description = TRUE,
  show_consent = TRUE,
  show_privacy_settings = FALSE,
  show_admin_settings = FALSE,
  title = NULL,
  logo = NULL,
  logo_width = 1.5,
  logo_height = 0.75,
  font = NULL,
  font_code = NULL,
  colors = NULL,
```

```

    authors = NULL,
    description = NULL,
    chrome_lang = NULL,
    variable_names = c("brackets", "underscore"),
    base_size = 10L
)

```

## Arguments

input	Either a path to a .lss file (character string) or a pre-parsed lss object returned by <code>read_lss()</code> . Passing a path parses it on the fly. Passing an lss object avoids re-parsing when the same survey is rendered with different options (e.g. multiple language subsets) in the same session.
output	Character. Path to the file to create. The extension determines the output format: .docx writes a Word document directly, .pdf writes a Word document into a temporary location and converts it locally via LibreOffice (or Word, on Windows). Any other extension is rejected with <code>lssdoc_bad_output_ext</code> .
languages	Character vector of language codes to display, <b>in the order they will appear as columns</b> . NULL (default) keeps all languages found in the .lss file in the order of the <languages> section. Acts both as a subset filter and an ordering: e.g. <code>c("en", "fr")</code> puts English first, <code>c("fr", "en")</code> puts French first. <code>languages[1]</code> is treated as the primary language (TOC entries, group fallback). Requesting a language absent from the survey is an error ( <code>lssdoc_unknown_language</code> ).
template	Output style. One of "cards" (the default) or "table". <ul style="list-style-type: none"> <li>"cards" renders one detached pair of tables per item (meta table + item table), stacked vertically, with content languages displayed side-by-side in the item table.</li> <li>"table" renders a single dense table covering the whole document: every variable is one tinted Question row carrying No   Variable   Type   Mand.   Filter, followed by one or more white Value rows. Group banners (and a group's description, when the author wrote one) become section rows; the column header repeats on every page. The body font steps down for three or four languages so the columns stay within the portrait content width.</li> </ul>
layout	Reserved for future use. Currently "auto" only.
show_audit	Logical. If TRUE (default), include an audit summary section near the top and inline markers on questions that carry findings. Set to FALSE for a clean reading copy.
show_help	Logical. If TRUE (default), include question help texts under the question text.
show_attrs	Character vector of question attributes to surface under the question text when present. Default keeps the attributes that change how respondents see the item: "prefix", "suffix", "other_replace_text", "validation". Add "exclude_all_others" or "exclude_all_others_auto" to also surface the row-level exclusivity flags (debug-style). Pass character(0) to hide all.
show_technical_attrs	Logical. If TRUE, include technical attributes such as <code>answer_order</code> and <code>location_*</code> . FALSE (the default) hides them.

page_format	Page format. One of "auto" (the default), "A4-portrait", "A4-landscape", or "A3". "auto" is template-aware and never follows the language count (four languages fit on A4 portrait): the "cards" template gets A4 portrait, while the dense "table" template gets A4 landscape, which it needs to stay readable. Pass an explicit value to override; selecting a wider page makes every panel (meta table, item table, audit/quota tables and the dense codebook table) expand automatically to fill the content width, in both templates.
show_toc	Logical. If TRUE (default), include a table of contents listing the groups (skipped automatically when the survey has fewer than two groups). For per-variable navigation, use show_index.
show_index	Logical. If TRUE (default), append a variable index at the end of the document listing every item code with its number, sorted alphabetically.
show_quotas	Logical. If TRUE (default), append a quotas section (after the end text, before the variable index) listing each sampling quota: its localized name, status (active, limit and action when full), the membership condition resolved to question codes and answer labels, and the localized "quota full" message. Skipped when the survey defines no quotas.
show_header_title	Logical. If TRUE (default), show the survey title at the top right of every page (one line per displayed language, truncated to 80 characters). FALSE keeps only the X/Y page counter at the bottom right.
show_source	Logical. If TRUE (default), show the <b>Source file</b> name and <b>Survey ID</b> rows in the cover metadata table. Pass FALSE to hide both (e.g. when sharing without exposing the LimeSurvey internals).
show_item_heading	Logical. If FALSE (the default), the meta table starts each item directly, for a compact layout. If TRUE, a bold "N. variable" heading is added above each item for scroll-time navigation.
show_raw_filter	Logical. If FALSE (the default), the Filter cell shows only the human-readable form (e.g. Q1 = 1) – editorial codebook style, matching ESS / MOSAiCH / GESIS conventions. Set to TRUE to also surface the raw LimeSurvey relevance expression underneath in small italic gray (e.g. !is_empty(Q1.NAOK) && (Q1.NAOK == 1)), useful for QA cross-checks. The raw form is always shown when the plain form could not be simplified.
show_groups	Logical. If TRUE (default), show the group banners (cards layout) or group rows (table layout). Pass FALSE to flatten the document into a single sequence of items with no section breaks (useful when groups exist only as internal organization).
show_welcome	Logical. If TRUE (default), include the survey's multilingual welcome text (surveyls_welcometext) as a side-by-side block (cards) or embedded row (table).
show_endtext	Logical. If TRUE (default), include the survey's multilingual end text (surveyls_endtext). Same treatment as show_welcome.
show_description	Logical. If TRUE (default), include the survey's multilingual description (surveyls_description) – the "what this survey is about" intro that LimeSurvey shows above the welcome text on the landing page.

show_consent	Logical. If TRUE (default), render a data protection and consent block in the front matter (before the welcome text): the survey's privacy policy notice (surveyls_policy_notice) and its consent checkbox label (surveyls_policy_notice_label), side by side across languages, with the checkbox drawn as an empty box. Skipped when the survey turns the policy notice off or carries no notice text.
show_privacy_settings	Logical. If FALSE (the default), omit the survey-level privacy / tracking flags from the cover. Set to TRUE to surface anonymized, save partial, datestamp, ipaddr, and refurl rows – useful for ethics committee submissions.
show_admin_settings	Logical. If FALSE (the default), omit the survey-level administrative settings. Set to TRUE to surface alias, end URL with description, and active flag rows.
title	Optional override of the survey title shown on the cover and the top-right header. NULL (default) uses the per-language titles from the .lss survey settings. Pass a single string to use the same title in every displayed language, or a named character vector keyed by language code (e.g. c(fr = "Mon titre", de = "Mein Titel")) for per-language overrides.
logo	Optional path (character) to a PNG or JPEG image displayed at the top of the cover page. NULL (default) keeps the cover logo-free, matching the neutral style of survey-methodology references (ESS, MOSAiCH, Panel). The .lss file does not embed a logo, so this image must be supplied by the caller.
logo_width, logo_height	Image dimensions in inches. Defaults 1.5 and 0.75, tuned to a 2:1 logo. Resize or pre-crop your image to fit a different aspect ratio.
font	Optional body font name (character). NULL (default) keeps Calibri, which is pre-installed on every recent Windows Office and metric-substituted with Carlito (OFL) on Mac and Linux LibreOffice, so column widths stay stable across platforms. Pass any string to override (e.g. "Source Sans 3", "IBM Plex Sans", or a corporate brand font); install the font on the machine that opens the document, otherwise the reader's application substitutes its own fallback face.
font_code	Optional monospace font name (character) used for code-like content: the variable column in each meta table, the raw relevance expression under each filter cell, and the variable index entries. NULL (default) keeps Consolas; pass "JetBrains Mono" or "IBM Plex Mono" for sharper code style.
colors	Optional named list of hex color overrides for the editorial petrol-blue palette. NULL (default) keeps the package palette intact. Accepted names: "primary" (group filets, headers text and item top borders), "accent" (hyperlinks, ORCID iD, URL auto-links, group under-line in cards), "band" (light header backgrounds), "band_dark" (the meta-table dark header in cards), "zebra" (the very-light tint on Question rows in the table template), "grid" (the 0.5 pt border color), "text", "muted". Each value must be a hex string ("#XXXXXX" or "#XXX"). Unknown keys are rejected (lssdoc_bad_colors). Useful for honoring an institutional brand: e.g. colors = list(primary = "#5C9F1A", accent = "#7FA82E") produces a LimeSurvey-green document.
authors	Optional credit block for the questionnaire's designers, displayed on the cover page below the subtitle. Each author is shown centered on its own line as Name

-- Affiliation; when an ORCID iD is provided, a smaller monospace line below shows ORCID 0000-0000-0000-0000 as a hyperlink to <https://orcid.org/<id>>.

Accepts:

- NULL (default): no authorship block.
- An **unnamed character vector** (`c("Jane Doe", "John Doe")`): each entry becomes a line with no affiliation.
- A **named character vector** (`c("Jane Doe" = "HESAV")`): names are authors, values are affiliations. Use "" to render an author without affiliation.
- A **list of named lists** for the full form, e.g. `list(list(name = "Jane Doe", affiliation = "HESAV", orcid = "0009-0001-2345-6789"), list(name = "John Doe", affiliation = "HESAV"))`. The name field is required; affiliation and orcid are optional.

description	Optional free-form text (single string) shown on the cover page below the authors block. NULL (default) omits the block. Useful for a citation hint, a funding acknowledgement, a methodology note, or a link to a related publication. Line breaks (\n) split the block into separate centered lines; http:// and https:// tokens are rendered as clickable hyperlinks.
chrome_lang	Language used for the <b>chrome</b> of the document (column headers, row labels, navigation titles, type labels, Value descriptors, audit section). One of "en", "fr", "de", "es", "it". NULL (default) follows languages[1] when supported, otherwise falls back to "en". Independent from languages, which controls the survey's content columns: e.g. chrome_lang = "en" with languages = c("fr", "en") produces an English-labelled document with French and English content. Spanish and Italian translations should be reviewed by a native speaker before publishing an official document.
variable_names	How response-variable names are written, so the document matches the data file the reader holds. One of: <ul style="list-style-type: none"> <li>• "brackets" (default) – the exact column names of the <b>CSV / Excel data export</b>, so the variable index reproduces the raw data file column for column: parent[subq], parent[subq][1] (dual scale), parent[CH] / parent[other] (multiple choice), parent[59842] (ranking, by answer id), parent[_Ccomment] (list-with-comment).</li> <li>• "underscore" – the sanitized code form used by the Expression Manager / relevance equations and the SPSS / Stata / R exports (parent_subq, parent_subq_1). The two-dimensional arrays (array of numbers / texts with a second axis) and ranking questions are expanded so every produced column appears as its own entry either way.</li> </ul>
base_size	Body type size in points (default 10). One lever scales the whole document: question text, item tables, the meta band, the quotas table, the variable index and the cover metadata all follow it, while headings and answer/help text keep their relative offsets. Useful for a roomier single-language render (e.g. 12). The cover title and subtitle keep their fixed title-page sizes. Accepted range: 7 to 16.

## Value

The output path, invisibly.

**"LimeSurvey last save" date on the cover**

The cover metadata table carries a row labelled *"LimeSurvey last save"* (or its localized equivalent). It is read verbatim from the `surveys.lastmodified` column of the `.lss`, which is the only timestamp LimeSurvey writes into the export – no other table (questions, question\_l10ns, answer\_l10ns, groups, etc.) carries a per-row modification date. The row is named "last save" rather than "last modified" because LimeSurvey only bumps that field reliably when the user clicks **Save** on a survey-level form (Settings tab); editing a question text, an answer label, or a translation through the Question Editor does **not** consistently update it across LimeSurvey versions. If the date looks stale relative to your most recent edits, the workaround is to open *Survey settings* in LimeSurvey, click **Save** (no other change needed), then re-export the `.lss`. The next render will show the bumped timestamp.

**Field-update prompt in Word**

Opening the rendered `.docx` in Microsoft Word may surface a security-style prompt: *"This document contains fields that may refer to other files. Do you want to update the fields in this document?"*. This is expected: the package marks the page-number and bookmark-reference fields as needing a refresh so the footer shows the correct page count and the table of contents links resolve to the right pages on first open (this is also what makes headless PDF conversion via LibreOffice produce correctly paginated output without a manual F9). Clicking **Yes** is safe – the document has no INCLUDETEXT, INCLUDEPICTURE-linked, or DDE fields; the only external links are the ORCID and DOI URLs in the cover credits, which are static HYPERLINK targets and not fetched on update.

**PDF output**

When output ends in `.pdf`, the function first renders a `.docx` to a temporary location and then converts it locally via LibreOffice headless (or Word on Windows). LibreOffice (`soffice` executable) must be installed and on PATH; otherwise a classed error explains how to install it. Conversion stays on the user's machine: no upload, no network call. LibreOffice headless does not refresh Word field values (TOC, page counts) during conversion, so the table of contents may appear empty in the converted PDF. To obtain a PDF with a populated TOC, render to `.docx` instead, open it in Word (the TOC refreshes automatically) and use File > Save As > PDF.

**See Also**

[render\\_audit\(\)](#) for the audit-only document; [audit\\_lss\(\)](#) to inspect findings in the console without rendering; [read\\_lss\(\)](#) to pre-parse a `.lss` file once and render multiple variants.

**Examples**

```
## Not run:
file <- system.file("extdata", "demo_survey.lss", package = "lssdoc")

# One-shot: parse + render Word document
render_questionnaire(file, tempfile(fileext = ".docx"))

# Same call, PDF output (format inferred from extension)
render_questionnaire(file, tempfile(fileext = ".pdf"))

# Parse once, render several variants without re-parsing
```

```
lss <- read_lss(file)
render_questionnaire(lss, tempfile(fileext = ".docx"),
  languages = "en")
render_questionnaire(lss, tempfile(fileext = ".docx"),
  template = "table",
  languages = c("en", "fr"))

# Branded cover with authors block and palette override
render_questionnaire(
  lss,
  tempfile(fileext = ".docx"),
  template = "table",
  chrome_lang = "en",
  colors = list(primary = "#5C9F1A", accent = "#7FA82E"),
  authors = list(
    list(name = "Jane Doe", affiliation = "HESAV",
      orcid = "0009-0001-2345-6789"),
    list(name = "John Doe", affiliation = "HESAV",
      orcid = "0009-0002-3456-7890")
  )
)

## End(Not run)
```

# Index

audit\_lass, [2](#)  
audit\_lass(), [3](#), [5](#), [11](#)

read\_lass, [3](#)  
read\_lass(), [2](#), [4](#), [7](#), [11](#)  
render\_audit, [4](#)  
render\_audit(), [3](#), [11](#)  
render\_questionnaire, [6](#)  
render\_questionnaire(), [3](#), [5](#)