

A Document Batch with a Row Report

Twelve fictional records, two templates and no silent omissions

01

02

03

How to use this guide

Coordinators generating small batches of letters and labels from structured records.

An original local merge case produces eight planning letters and nine desk labels, with per-template rejection reports, Unicode and escaped-text examples, and complete row reconciliation.

- Define separate letter and label contracts.
- Generate valid outputs while recording partial failures.
- Preserve CSV records, Unicode names and data-as-text boundaries.
- Reconcile all twelve inputs across twenty-four decisions.

Scope and limitations

- The generator writes HTML and reports locally; it does not email, upload or verify delivery.
- Character limits are preliminary exercise rules, not proof of final visual fit.
- No attendance, qualification, official certificate, tagged PDF or universal font compatibility is promised.

Published by KitForma. Edition 1.0. Sources checked 1 October 2026. People and businesses in worked examples are fictional. This is an independent publication, not endorsed by the platforms mentioned.

Copyright 2026 KitForma. Usage, support and refund terms are included in LICENSE.txt in the purchased package.

Contents

How to use this guide	2
Contents	3
Make every input row accountable	4
Sources and tools	6

01

Make every input row accountable

The fictional River Lab coordinator needs two kinds of workshop documents: a short planning letter and a desk label. Twelve fictional records are available, but some contain missing or unsuitable values. The goal is not to force all twelve through both templates. The goal is to produce the valid outputs, explain every rejection, and reconcile the result so no person or document disappears silently from the batch. This workbook treats the row report as part of the deliverable.

A planning letter and a desk label have different requirements. The letter needs a name, a valid event date, and a note that fits the exercise's declared limit. The label needs a name that fits its short layout and a room value. A missing date can therefore reject a letter without rejecting the corresponding label. This is a deliberate partial-failure case. A single global valid flag would conceal the fact that one record can be suitable for one output and unsuitable for another.

The supplied generator uses Python 3's standard library to create HTML files and separate JSON reports. It sends no email, opens no account, and creates no certificates or qualifications. The letters describe a fictional planned workshop, not verified attendance or an earned credential. The output is intended for local inspection and optional browser printing. A PDF generated later from that HTML requires its own visual and reader checks; the script's success does not prove a PDF export passed.

Start by assigning an immutable row identifier such as R001. Keep it separate from the person's displayed name. Names can contain spaces, accents, punctuation, or identical spellings across different people; they are poor file identifiers. The exercise uses twelve unique row IDs so every

output and rejection can be traced to one input record. This small decision supports reliable filenames, reruns, and reconciliation before any template design begins.

Sources and tools

[S1] RFC 4180: common CSV format

<https://www.rfc-editor.org/rfc/rfc4180>

Checked: 2026-10-01

[S2] Microsoft: accessible source document practices

<https://support.microsoft.com/en-us/accessibility/word/make-your-word-documents-accessible-to-people-with-disabilities>

Checked: 2026-10-01

[S3] W3C PDF3: reading and tab order

<https://www.w3.org/WAI/WCAG22/Techniques/pdf/PDF3>

Checked: 2026-10-01

Inside the complete guide

Define each template's contract

Read the CSV as records, not lines

Map fields without turning data into markup

Reject a row with a reason someone can fix

Design overflow checks that lead to a decision

Generate into a clean run directory

Reconcile counts by output kind

Review representative files and the release package

Capstone and answer key