ALA CORE FORUM 2026 · LOUISVILLE, KY

Automating MARC Holdings Generation with AI: Lessons from the Field

Built with an AI coding assistant; runs without one.

Matthew Codey · USC Libraries

USC University of Southern California

More detail, the figures and a PDF of this poster
matthewcodey.com/poster

1 The problem, and the approaches I tried

The problem

Many libraries' serials holdings are stored only as human-readable, free-text **866** fields rather than as MARC-encoded **853/863** pairs. Converting by hand is slow, conventions vary, and hand-typed data is prone to typos and other errors.

> **The goal:** a tool that is portable across collections, works on messy holdings as they are (no manual clean-up), and never silently loses anything.

866 · as a person typed it

```
v. 27 no. 4-v. 31 no. 4
a  a  b  b  a  a  b  b
(April 1992-April 1996)
 j     i     j     i
```

subfield	853 caption	863 held
$a volume	v.	27-31
$b issue	no.	4-4
$i year	(year)	1992-1996
$j month	(month)	04-04

```
853 31 $8 1 $a v. $b no. $i (year) $j (month)
863 40 $8 1.1 $a 27-31 $b 4-4 $i 1992-1996 $j 04-04
```

APPROACH 1

Write the script by hand

I adapted a script by USC colleagues (Chung & Pendyala, 2023) over six months while teaching myself Python. The hardest part was a single, very long regular expression.

APPROACH 2

Ask a model to read the statements

Send the 866s to a large language model through its API, and ask it to pick out the volumes, issues and dates.

APPROACH 3 · CHOSEN

Use AI to build the tool

Describe in plain language what the tool should do, and have an AI coding assistant (Claude Code) write a deterministic Python application.

	1 · BY HAND	2 · A MODEL READS	3 · AI BUILDS THE TOOL
Coding knowledge	The most	The least	Some
Getting started	Over six months	Minimal prompting	A first working tool from one prompt
Cost	Nothing	Tokens for every statement	About $20/month (Pro plan) to build; nothing to run
Same output every time	Yes	Not guaranteed	Yes
Other collections	Poorly. It needs clean, consistent 866s	Not tested far enough	Designed for it
Understanding the code	Fully, line by line	No code to read	Partly: a black box, checked by tests

2 The approach chosen: AI-assisted development of a deterministic tool

How the Workbench reads a statement

1 · Cut into labeled pieces

```
v.  27  no.  4   -   v.  31  no.  4
VOL  #  ISS  #   —  VOL  #  ISS  #

(  April  1992  -  April  1996  )
    CHRON  YEAR  -  CHRON  YEAR
```

2 · Its shape

```
VOL ISS — VOL ISS(CHRON YEAR-CHRON YEAR)
```

```
v. <start_vol> no. <start_iss>-v. <end_vol> no. <end_iss>
(<start_month> <start_year>-<end_month> <end_year>)
```

1. Statements with the same shape are one group, whatever their numbers.
2. Each group gets a matching rule with a named slot for each value.
3. A cataloger confirms what each value means, once per group.
4. The parser reads every statement; a pattern fills only what it cannot know.

What the cataloger controls

- **On a pattern:** what each value is, its caption and level; skip a shape; test or edit the rule; reuse the library on another collection.
- **On one record:** edit an 866 before it is read, split it, set this record's 853 (frequency, numbering, issues per volume, missing captions), skip it, leave a note.
- **For the whole file:** holdings level, subfield convention, strict reading, what to do with a value MARC has no place for, and whether to keep each 866.

Every decision is listed in a downloadable log.

The working rules

1. **Measure before you change.** Count first, then fix.
2. **Nothing disappears silently.** Encoded, named, or held.
3. **One reader of holdings.** Patterns only fill its gaps.
4. **If a fix needs the tests rewritten, stop.** Rethink the fix.
5. **Write the changelog for catalogers,** not for coders.

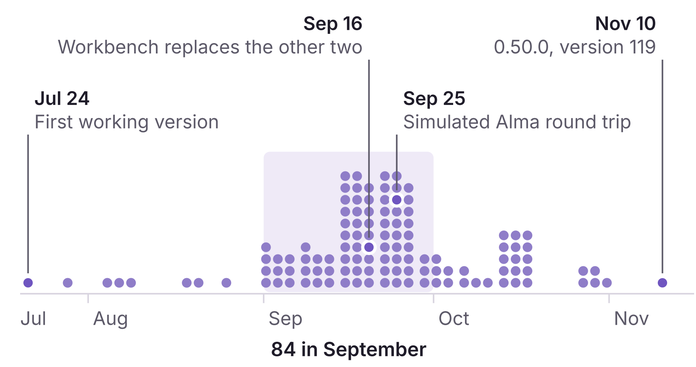
The human in the loop

The AI writes the code; I decide what "correct" means. Using the tool, I found two bugs the tests had missed:

- `Late Summer 2002` was coded as plain Summer, but there may be an Early Summer issue too. Now flagged (0.9.6).
- `v. 58 Suppl. (Sep 2003)` lost "Suppl." when its pattern was confirmed. Supplements are now held (0.32.0).

How it was built

119 numbered versions in 15 weeks, most of them small.



Each dot is one version. The Pattern Detector and Converter each began as a single prompt. The Workbench combined them, then replaced them on September 16.

On real data

A real Alma export of 372 records:

1,057 holdings statements

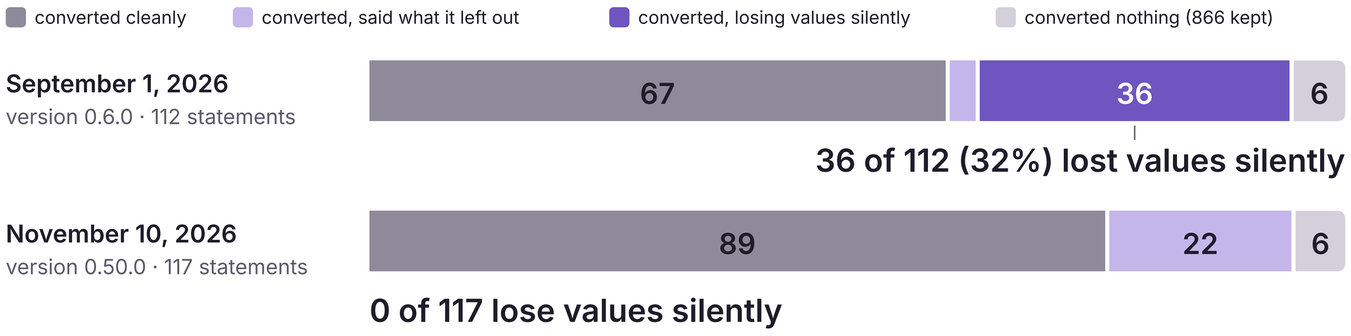
0 values unaccounted for

0 drift, simulated Alma round trip

48/48 of Alma's own 866s reproduced

The turning point: silent losses

In a silent loss, the tool writes the 853/863, removes the 866, and drops a value (a volume, an issue, a year) with nothing on screen to say so.



Zero is what was measured, not a guarantee. A review by a new AI model, Fable 5.2, found one more: `v.3 (1990--1995)` kept only 1990. Fixed in 0.50.0.

Lessons, and what's next

- **A script reads what it was written for and passes over the rest.** So the most important rule is that nothing disappears silently: every value is encoded, named as left out, or held with its 866 kept.
- **Decide whether you can work with code you don't fully understand.** Tests show the tool works, not that it was built well. That takes a software engineer to judge.
- **Where this may be heading:** people build their own tools with AI, and engineers build the guardrails.
- **The models improved as I worked.** Newer ones solved what had stopped older ones, then built features from one prompt.
- **AI is at its best building the tool, not being it.**
- **Next:** more real .mrc files from colleagues, a review by a software engineer, then a 1.0 release.

Try it: tools.matthewcodey.com/workbench · **Source:** github.com/bdmcodey/holdings-workbench
Contact: codey@usc.edu (work) · matthew.codey@gmail.com (personal)

License: © 2026 Matthew Codey, CC BY 4.0 (creativecommons.org/licenses/by/4.0). The USC logo is not covered.
Reference: Chung, M., & Pendyala, P. C. (2023). Enhancing serials holdings data: A Pymarc-powered clean-up project. *Code4Lib Journal*, (58).