

DualBookmarker: A Python Tool to Create Dual Bookmarks for Annotated CRF

Xianhua Zeng, Taimei Intelligence Pharmaceutical, Shanghai, China

ABSTRACT

The annotated Case Report Form (aCRF) is a required document for FDA submissions that maps each CRF field to its corresponding variable in the SDTM dataset. According to the CDISC SDTM Metadata Submission Guidelines version 2.0 (SDTM-MSG v2.0), the aCRF should be dual bookmarked: (1) bookmarks by chronology and (2) bookmarks by CRF topics or forms, so that reviewers can navigate the document either by visit or by topic. Manually creating dual bookmarks is tedious, time-consuming, and error-prone, particularly for studies with many visits and forms. Existing automation approaches typically rely on commercial Adobe Acrobat plug-ins combined with SAS, spanning multiple environments and requiring paid licenses. This paper introduces DualBookmarker, a pure-Python tool that automatically generates SDTM-MSG v2.0-compliant dual bookmarks for an aCRF using only the open-source PyMuPDF library. It runs on any platform, supports both GUI and CLI modes, and includes built-in QC checks that verify visit count, domain count, and cross-reference consistency before writing the result back to the aCRF.

The application is available for download on my GitHub:

<https://github.com/XianhuaZeng/Projects/raw/master/DualBookmarker/DualBookmarker.exe>

INTRODUCTION

The annotated Case Report Form (aCRF) is a PDF document that maps each CRF field to the corresponding variable in the SDTM dataset and is a required component of regulatory submissions to agencies such as the FDA and EMA. Per SDTM-MSG v2.0, the aCRF must include a printable Table of Contents and dual bookmarks – by chronology and by topic/form – hyperlinked to the appropriate CRF page so reviewers can navigate either way without flipping through hundreds of pages. Bookmarks by chronology follow the study Schedule of Activities (SOA), with study-level pages such as Adverse Events and Concomitant Medications grouped last under “Running Records”; bookmarks by topic are ordered alphabetically (or in the order the forms appear), and within each topic the applicable visits appear chronologically.

Previous papers in the industry have demonstrated that dual bookmarking can be semi-automated by combining SAS with Adobe Acrobat plug-ins such as ISIToolBox or AutoBookmark Pro. These approaches work but carry drawbacks: they depend on commercial plug-ins (license cost, version compatibility), they require SAS. DualBookmarker takes a different approach – it is a single Python script that depends only on PyMuPDF, runs on any platform, and produces SDTM-MSG v2.0-compliant dual bookmarks in seconds.

The following is the GUI of DualBookmarker.

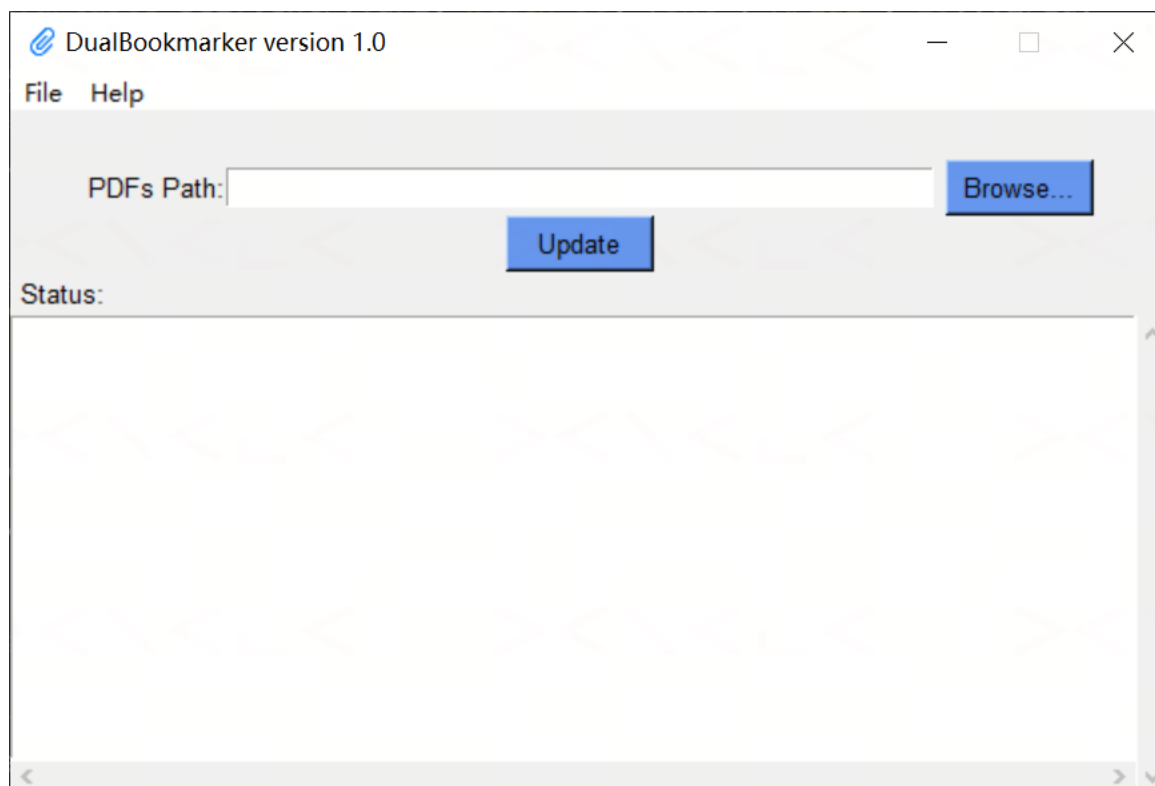


Figure 1. GUI of DualBookmarker

FUNCTIONALITIES

DualBookmarker takes two input PDFs from the same folder and writes the dual bookmarks back to the first one. The two inputs are:

1. aCRF.pdf – the annotated CRF containing only the unique CRF forms (one page per form), already bookmarked by form name. This is the file that will receive the new dual bookmarks.
2. AllCRF.pdf – the full CRF containing all pages for all study visits, already bookmarked by visit → form. This file is used only as a reference to learn the visit/form structure and is not modified.
3. Given these two inputs, DualBookmarker performs four steps:
 - Extract – Reads the existing bookmark tree from both PDFs via PyMuPDF and writes them to two intermediate text files (aCRF.txt, AllCRF.txt) in the jpdfbookmarks format. Cover-page and self-duplicate bookmarks are cleaned automatically.
 - Remap – Builds two synchronized bookmark trees and writes them to Final.txt: BY VISIT (chronological, mirroring the SOA from AllCRF.pdf) and BY DOMAIN (alphabetical by form, mirroring the form list from aCRF.pdf). Both trees point to pages in aCRF.pdf, so every link lands on the correct annotated page.
 - QC checks – Validates the remapped tree against three automated checks: visit count, domain coverage, and cross-references.
 - Apply – Writes the remapped bookmark structure back into aCRF.pdf in place, replacing the existing bookmarks, using PyMuPDF's `set_toc` API.

KEY FEATURES

1. Pure Python – only depends on PyMuPDF (open-source, MIT licence). No Adobe Acrobat plug-in, no SAS, no Unix `pdftotext` command required.

2. Cross-Platform – works on Windows, macOS, and Linux. The packaged executable is lightweight and self-contained.
3. Dual Mode – a Tkinter GUI for manual users with a “Browse and Update” workflow, and a CLI for automated pipelines (CI/CD or submission QC).
4. Built-in QC – three checks are run on every update: (1) visit count match between BY VISIT and AllCRF.pdf, (2) domain count match between BY DOMAIN and aCRF.pdf, and (3) cross-reference check that every visit name referenced in BY DOMAIN exists in BY VISIT. Each result is reported as [QC PASS] or [QC FAIL]. The tool also refuses to re-run on an aCRF.pdf that has already been dual-bookmarked, avoiding accidental double-processing.

TESTING

To run DualBookmarker via the GUI:

1. Place aCRF.pdf and AllCRF.pdf in the same folder.
2. Open DualBookmarker, click Browse to select the folder, then click Update.

During the run the status box reports each step in real time. When the run completes successfully the GUI displays a clickable hyperlink to the updated aCRF.pdf, as shown below.

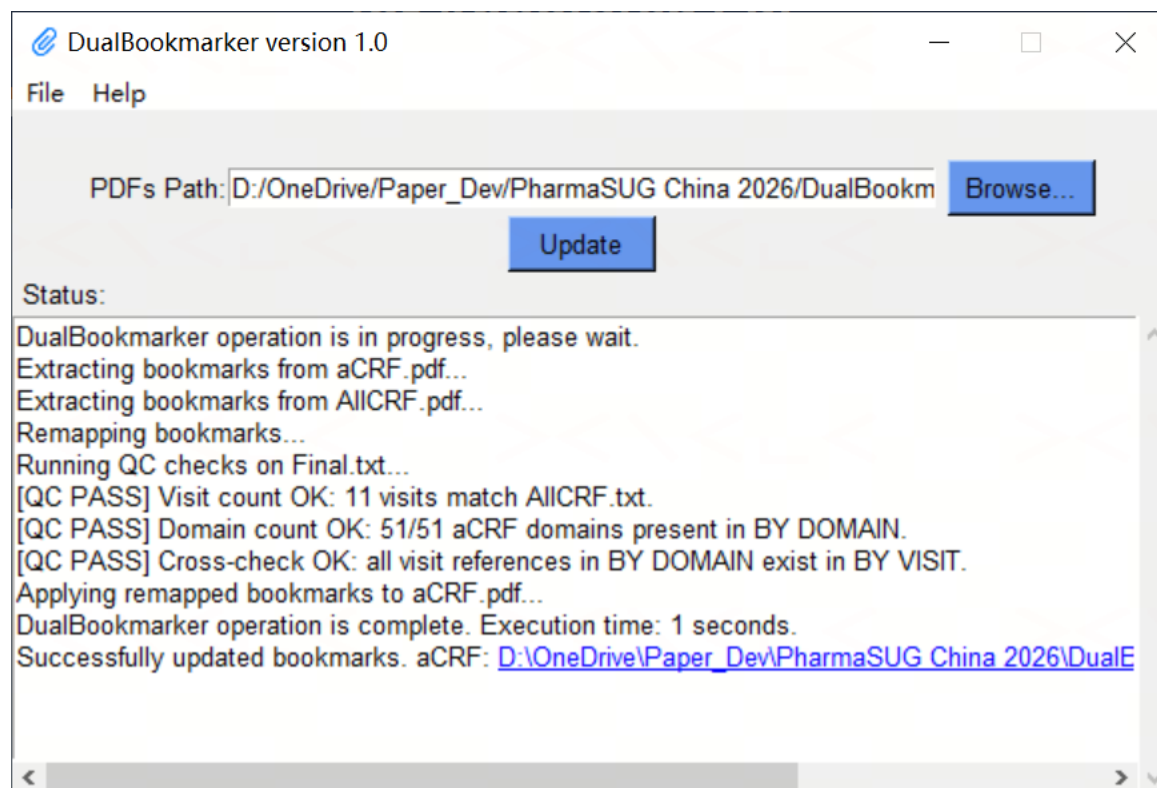


Figure 2. Execution Status of DualBookmarker

Once the operation is complete, the “Status” box displays the execution log, including:

1. Step-by-step progress messages (Extracting, Remapping, QC, Applying)
2. QC results ([QC PASS] / [QC FAIL]) for visit count, domain count, and cross-references

CLI USAGE

DualBookmarker also supports a command-line interface for automated, script-driven workflows. The CLI exposes a single update command with two input modes:

3. Auto-detect mode (-d DIR): automatically locates aCRF.pdf and AllCRF.pdf in the specified directory by filename.
4. Explicit mode (--acrf FILE --allcrf FILE): accepts direct file paths for full control in automated pipelines.

Example CLI commands:

- `python DualBookmarker.py update -d C:\study\crf`
- `python DualBookmarker.py update --acrf C:\study\crf\aCRF.pdf --allcrf C:\study\crf\AllCRF.pdf`

The CLI exits with code 0 on success and code 1 on failure, making it suitable for integration into CI/CD pipelines or submission quality-control workflows.

CONCLUSION

DualBookmarker provides a lightweight, cross-platform, plug-in-free way to generate SDTM-MSG v2.0–compliant dual bookmarks for an aCRF. By replacing the traditional SAS-plus-commercial-plug-in workflow with a single Python script, it reduces manual effort, eliminates commercial dependencies, and produces consistent, QC-verified results suitable for regulatory submission. The tool is a practical addition to the toolkit of any statistical programmer or submission team working with SDTM deliverables.

REFERENCE

Fredrik Lundh. “Tkinter”. Available at <https://docs.python.org/3/library/tk.html>

PyMuPDF Documentation. Available at: <https://pymupdf.readthedocs.io/>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Name: Xianhua Zeng
Enterprise: Taimei Intelligence Pharmaceutical
Address: 6th Floor, Building 24, Phase 3, Caohejing Technology Oasis, No. 1999 Yishan Road,
Minhang District, Shanghai, China, 200233
E-mail: xianhua.zeng@taimei.com
Web: <https://www.taimei.com/>
<http://www.xianhuazeng.com/>