PharmaSUG 2026 – Paper PO-007

DefinePageChecker: A Python Tool for Verifying Page Number Hyperlinks in Define.xml

Xianhua Zeng, Taimei Intelligence Pharmaceutical, Shanghai, China

ABSTRACT

The accuracy of page number hyperlinks in define.xml files is crucial for regulatory submissions in clinical trials. These files often include page number hyperlinks to ensure accurate cross-referencing in the clinical trial report. Discrepancies in these hyperlinks may lead to incorrect or broken links, compromising the integrity of the report. This paper introduces DefinePageChecker, a Python tool designed to automate the verification of page number hyperlinks in Define.xml files. The tool checks whether the hyperlinks correspond to the correct page numbers and missing annotation pages, ensuring that the report's navigation structure remains intact and reliable.

The app, source code, and test files are available on my GitHub:

https://github.com/XianhuaZeng/PharmaSUG/tree/master/2026/DefinePageChecker

INTRODUCTION

The define.xml file is a cornerstone of electronic submissions to regulatory agencies, including the FDA and EMA. It provides metadata about datasets, variables, and other clinical trial elements, often including hyperlinks to specific pages in the aCRF. Ensuring the accuracy of these hyperlinks is essential to maintain the integrity and usability of the define.xml file.

Manual verification of page number hyperlinks is both time-consuming and error-prone. DefinePageChecker addresses this issue by automating the verification process. Building upon the methodologies originally developed in PharmaSUG 2019's "Automating Quality Checks for define.xml Files", this tool has been further enhanced by leveraging PyPDF2 for hyperlink validation, offering a streamlined and robust solution to identify and resolve hyperlink discrepancies.

The following is the GUI of DefinePageChecker.

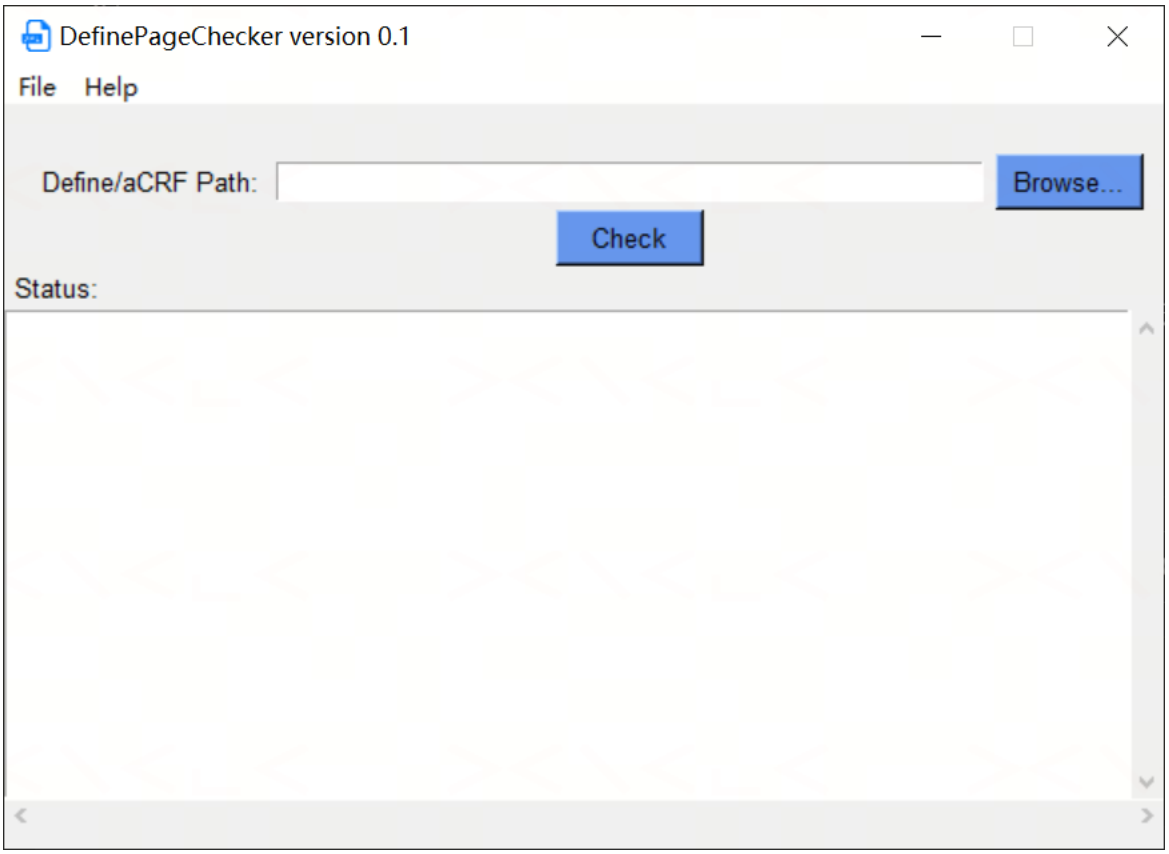

Figure 1. GUI of DefinePageChecker

FUNCTIONALITIES

The DefinePageChecker GUI is designed to be user-friendly and efficient (Figure 1). The core functionalities of the tool include:

1. Automated Verification Process: DefinePageChecker parses the define.xml file, extracts hyperlinks, and validates their correctness against the associated aCRF file.

2. Graphical User Interface (GUI): An intuitive interface enables users to easily interact with the tool, input files, and monitor status updates.

3. Real-Time Feedback: Provides instant status updates within the GUI, such as:
   - "Error: DM.RFXSTDTC not found on page 39."
   - "Note: there are no annotations on page 41."
   - "DefinePageChecker operation is complete. Execution time: 1 seconds."

KEY FEATURES

1. Displays errors and progress updates directly within the GUI for real-time insights.

2. Cross-Platform Compatibility: DefinePageChecker supports various versions of Windows and is lightweight for easy deployment.

3. User-Friendly Interface: A straightforward "Browse and Check" workflow eliminates unnecessary complexity.

4. Command-Line Interface (CLI): Supports automated, script-driven workflows with two input modes — auto-detect files from a directory (-d) or specify file paths explicitly (--define / --acrf). Exits with code 0 (no issues) or 1 (issues found) for easy pipeline integration.

TESTING

To run the DefinePageChecker function:

1. Open DefinePageChecker, Use the "Browse..." buttons to select the define.xml file and the corresponding aCRF file.

2. Click the "Check" button to initiate the process.

If the process completes successfully, the GUI will appear as shown below:

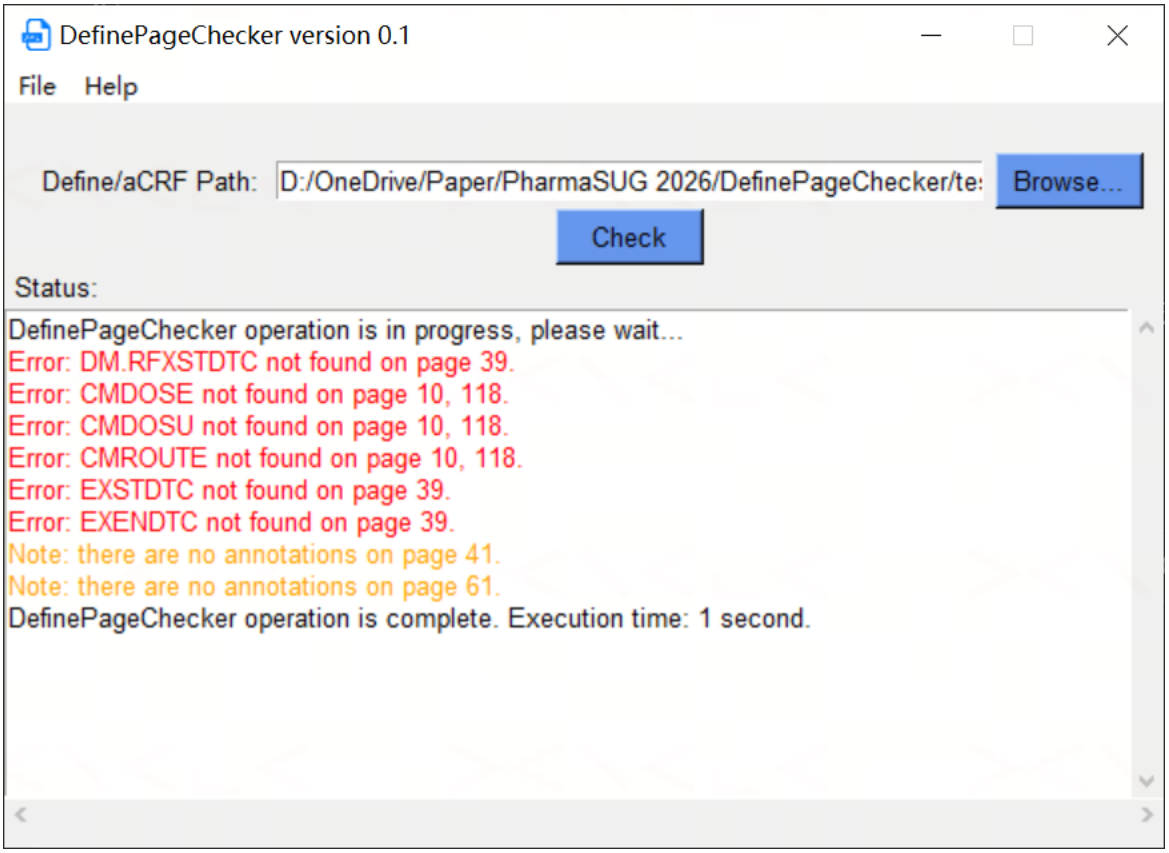

Figure 2. Execution Status of DefinePageChecker

Once the operation is complete, the "Status" box will display the following issues:

1. Error page link

2. Missing annotations page

CLI USAGE

DefinePageChecker also supports a command-line interface (CLI) for automated and script-driven workflows. The CLI exposes a single check command with two input modes:

3. Auto-detect mode (-d DIR): Automatically locates define.xml and the aCRF PDF in the specified directory by filename pattern.

4. Explicit mode (--define FILE --acrf FILE): Accepts direct file paths for full control in automated pipelines.

Example CLI commands:

- `python DefinePageChecker.py check -d C:\submission`
- `python DefinePageChecker.py check --define C:\sub\define.xml --acrf C:\sub\acrf.pdf`

The CLI exits with code 0 when no issues are found, and code 1 when issues are detected, making it suitable for integration into CI/CD pipelines or submission quality-control workflows.

CONCLUSION

DefinePageChecker is a valuable addition to the toolkit of clinical trial professionals, providing a reliable and efficient solution for verifying page number hyperlinks in define.xml files. Its user-friendly GUI and robust functionality make it a practical choice for ensuring the accuracy of regulatory submissions.

REFERENCE

PharmaSUG 2019: Automating Quality Checks for define.xml Files. Available at: https://pharmasug.org/proceedings/2019/AD/PharmaSUG-2019-AD-211.pdf

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

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: http://www.xianhuazeng.com/