

Veristat TLF Packager (GUI): An Interactive Tool for Automated Creation of Combined PDFs with Bookmarks and Clickable Table of Contents

Manivannan Mathialagan, Veristat, Chennai, India

ABSTRACT

In many clinical trials, Tables, Listings and Figures (TLFs) are generated across multiple programs and formats (RTF, DOCX and PDF), then manually collated into study packages. This manual process—opening files one by one, copying titles, renaming outputs, building bookmarks and creating a Table of Contents (TOC)—is time-consuming, error-prone and difficult to standardize across projects and programmers.

The Veristat TLF Packager is a Python-based GUI tool designed to automate this packaging step. The tool scans a selected folder, detects RTF, DOCX and PDF files, and automatically extracts TLF titles from RTF/DOCX headers and the first page of PDFs. It presents the outputs in an interactive interface where users can filter, select, edit bookmark text and reorder rows. From this single platform, users can export review sheets to Excel and generate a merged, bookmarked PDF with an optional, multi-page clickable TOC. By centralizing title extraction, review and packaging into one repeatable workflow, the Veristat TLF Packager significantly reduces manual effort.

INTRODUCTION

The generation of Tables, Listings and Figures (TLFs) is a core component of clinical study reporting. These outputs are frequently produced across multiple statistical programs and formats, often requiring manual consolidation before submission. In many organizations, packaging TLFs into a final deliverable involves repetitive, labor-intensive steps such as reviewing titles file by file, renaming outputs, checking ordering, manually adding bookmarks, and building a structured Table of Contents for regulatory or internal review. This process not only consumes valuable programming time but also increases the risk of inconsistency and human error.

The Veristat TLF Packager was developed in this context: to provide a structured, efficient and user-friendly approach to TLF packaging, replacing fragmented manual steps with a standardized, repeatable workflow that can be applied across studies and teams.

DATA PRIVACY AND COMPLIANCE STATEMENT:

This paper does not contain or reference any real study data. All screenshots, examples, and outputs shown in this presentation are generated using dummy datasets created solely for demonstration purposes. No sponsor-specific files, confidential documents, or PHI/PII were used during the preparation of the examples included here.

BACKGROUND AND CURRENT CHALLENGES

FRAGMENTED OUTPUT SOURCES

In a typical clinical study, TLFs are generated independently across programming teams and phases, often in multiple formats (RTF, DOCX and PDF). These outputs follow varying naming conventions and structural patterns depending on the programmer or legacy processes. Once produced, they must be collected, reviewed and packaged into a final deliverable such as a CSR, DSMB/DMC package or interim safety review.

In many cases, statisticians, medical writers or clinical leads may also contribute additional components—such as executive summaries, narrative-style listings, figures or charts—that are created outside the standard SAS programming environment. These materials are frequently produced in formats which do not align structurally with SAS-generated RTF/PDF outputs. This increases the complexity of assembling process.

MANUAL AND INEFFICIENT ASSEMBLY PROCESS

The final packaging step is predominantly manual in many organizations. Programmers often need to:

- ❖ Open each file individually to confirm title accuracy
- ❖ Extract titles manually and confirm consistency with specifications
- ❖ Rename files to ensure ordering alignment
- ❖ Convert files from RTF/DOCX to PDF manually
- ❖ Build bookmarks and TOC entries one by one inside PDF editing software
- ❖ Recheck each iteration whenever outputs are regenerated

This repetitive process increases the likelihood of typographical errors, missed updates, incorrect sequencing and inconsistent bookmark structures. It also becomes increasingly inefficient as the volume of outputs grows during multiple review cycles. For studies with frequent interim reviews or DSMB packages, manual packaging can consume hours or even days per cycle. These challenges motivated the development of a streamlined approach to consolidate outputs efficiently and reliably.

SOLUTION OVERVIEW – VERISTAT TLF PACKAGER

The Veristat TLF Packager was developed to streamline and standardize the final assembly of clinical study outputs. Instead of relying on repetitive manual steps, the tool provides an automated workflow that scans a selected folder, identifies RTF, DOCX and PDF files, extracts TLF titles and presents them in a structured GUI for review. Users can filter, select, reorder and edit bookmark text, then export the selection to Excel or generate a single merged PDF with consistent bookmark structure.

The tool extracts titles directly from RTF/DOCX headers or from the first page of PDFs, applies automated ordering based on TLF numbering, and supports RTF/DOCX-to-PDF conversion using Microsoft Word. A key strength is the ability to create a fully formatted, clickable Table of Contents (TOC) with hyperlink navigation. The resulting deliverable is submission-ready, reduces risk of sequencing or formatting errors and cuts packaging time from hours to minutes—particularly valuable for frequent DSMB/DMC review cycles and interim deliverables.

KEY FEATURES OF THE TLF PACKAGER

AUTOMATED FILE SCANNING AND METADATA EXTRACTION

The tool automatically scans a selected folder and detects all RTF, DOCX and PDF files, eliminating the need to manually track or re-import outputs after each run. It extracts TLF titles directly from RTF/DOCX headers and from the first page of PDFs, enabling reviewers to confirm accuracy without opening individual files. This ensures rapid validation of output completeness and prevents missing or duplicated deliverables during multiple rerun cycles.

INTERACTIVE GUI FOR REVIEW AND REORDERING

Users can filter, search, select or deselect outputs, reorder using Move Up/Move Down controls and edit bookmark text inline. A live file count displays selected RTF, DOCX and PDF documents, ensuring immediate visibility into packaging scope. Auto-refresh detects newly generated outputs in real time.

CONSISTENT BOOKMARK AND PDF PACKAGING

The tool applies automated ordering using a sorting algorithm based on Table, Listing, Figure and Appendix numbering. RTF/DOCX files can be converted to PDF using Microsoft Word with a single click. When packaging, selected files are merged into a single bookmarked PDF, preserving exact pagination and content integrity.

ADVANCED TABLE OF CONTENTS (TOC) GENERATION

The Packager generates a fully formatted TOC with wrapped long titles, dot leaders and clickable hyperlinks to each TLF. Users may enable or disable TOC creation depending on deliverable purpose. The final PDF is submission-ready and reduces the risk of manual bookmark or sequencing errors. This automation ensures a clear and professional navigation structure that is consistent across CSR, DSMB/DMC and interim deliverable packages.

EXCEL EXPORT AND DOCUMENTATION SUPPORT

All metadata (filename, format, titles, bookmarks, selected order) can be exported to an Excel review sheet. This enables independent QC review and facilitates documentation for audit trails, validation and version control. The exported Excel can also be shared with cross-functional stakeholders, enabling collaborative review and faster feedback cycles without accessing SAS or raw output files.

STANDALONE EXECUTABLE DEPLOYMENT

The tool is packaged as a Windows executable using PyInstaller, enabling users to launch the application without installing Python or external libraries. This makes the solution accessible to statisticians, medical writers, and reviewers without programming knowledge.

ARCHITECTURE AND TECHNICAL IMPLEMENTATION

SYSTEM ARCHITECTURE OVERVIEW

The Veristat TLF Packager is developed in Python using the PyQt5 framework for the GUI interface and leverages several specialised libraries for document handling and metadata extraction. Title extraction logic is implemented individually for RTF, DOCX and PDF formats, using PyMuPDF (fitz) for PDF text extraction and python-docx for DOCX processing. RTF headers are parsed using custom text parsing logic to identify title rows embedded in table structures.

The tool follows a modular component design to improve readability and long-term maintenance. File detection and auto-refresh components independently monitor changes within the selected workspace folder, enabling real-time update of the GUI without manual refresh actions.

The screenshot displays the Veristat TLF Packager interface. At the top, the title bar reads "Veristat TLF Packager". Below the title bar, there is a search bar with the text "Type to filter ro...", a checked "Include TOC page" checkbox, and a status indicator "Selected: 0 RTF | 1 DOCX | 33 PDF". A note states: "Bookmark text is editable (double-click), and Include column can be checked/unchecked." Below this is a table with columns: "Include", "Type", "Filename", "Title 1", "Title 2", "Title 3", and "Bookmark". The table lists 13 rows of files, including PDFs and a DOCX file. A modal dialog box is open in the center, displaying a message: "Output PDF generated: P:/Projects/Veristat/BSP Demo Project/BSP/Output/Development TLFs/tables_shells/TLFs_Merged_20251126_T2156.pdf". Below the table, there are several action buttons: "Browse Folder", "Select All", "Export to Excel", "Move Up", "Move Down", "Convert to PDF", "Pack & TOC", and "Terminate". At the bottom, there is a text field for "Output PDF Name" containing "TLFs_Merged_20251126_T2156.pdf" and a "Log" section showing a list of file operations and their completion status.

APPLICATION WORKFLOW

At a high level, the application follows a structured processing flow:

- scan_folder() - Detects RTF/DOCX/PDF files in the selected directory
- extract_titles() - Extracts TLF titles and builds bookmark text
- refresh_gui() - Updates the table view with file metadata
- export_excel() - Generates Excel review sheet with TLF information
- convert_to_pdf() - Converts RTF/DOCX files to PDF using MS Word
- merge_and_create_toc() - Merges selected files and adds bookmarks/TOC
- package_as_exe() - Packages the tool as a standalone executable for deployment

The GUI provides the core interaction layer, while background threads execute longer-running processes such as PDF conversion and merging, ensuring responsive behaviour during heavy operations such as DSMB runs or high-volume output batching.

EXE PACKAGING USING PYINSTALLER

The TLF Packager is compiled into a standalone Windows executable to support users who do not have Python installed. The tool is packaged using PyInstaller in `--onefile` mode, which bundles all dependencies (PyQt5, PyMuPDF, python-docx, openpyxl, win32com) into a single portable EXE.

The following build command is used:

```
pyinstaller --onefile --windowed --icon app.ico Veristat_TLF_Packager.py
```

```
C:\Windows\system32\cmd.exe
F_Packager.txt
161027 INFO: Graph cross-reference written to P:\BSP_LocalDev\Manivannan.Mathialag\zzzz My SAS Files\Presentation\PHUSE 2026\Tool\EXE Tool\build\Veristat_TLF_Packager\F-Veristat_TLF_Packager.html
161149 INFO: checking PYZ
161150 INFO: Building PYZ because PYZ-00.toc is non existent
161150 INFO: Building PYZ (ZlibArchive) P:\BSP_LocalDev\Manivannan.Mathialag\zzzz My SAS Files\Presentation\PHUSE 2026\Tool\EXE Tool\build\Veristat_TLF_Packager\PYZ-00.toc
163858 INFO: Building PYZ (ZlibArchive) P:\BSP_LocalDev\Manivannan.Mathialag\zzzz My SAS Files\Presentation\PHUSE 2026\Tool\EXE Tool\build\Veristat_TLF_Packager\PYZ-00.toc completed successfully.
163929 INFO: checking PKG
163930 INFO: Building PKG because PKG-00.toc is non existent
163930 INFO: Building PKG (CArchive) Veristat_TLF_Packager.pkg
200462 INFO: Building PKG (CArchive) Veristat_TLF_Packager.pkg completed successfully.
200492 INFO: Bootloader C:\Users\Manivannan.Mathialag\AppData\Roaming\Python\Python313\site-packages\PyInstaller\bootloader\Windows-64bit-intel\runw.exe
200492 INFO: checking EXE
200493 INFO: Building EXE because EXE-00.toc is non existent
200493 INFO: Building EXE from EXE-00.toc
200496 INFO: Copying bootloader EXE to P:\BSP_LocalDev\Manivannan.Mathialag\zzzz My SAS Files\Presentation\PHUSE 2026\Tool\EXE Tool\dist\Veristat_TLF_Packager.exe
200564 INFO: Copying icon to EXE
200623 INFO: Copying 0 resources to EXE
200623 INFO: Embedding manifest in EXE
200687 INFO: Appending PKG archive to EXE
203373 INFO: Fixing EXE headers
208617 INFO: Building EXE from EXE-00.toc completed successfully.
208646 INFO: Build complete! The results are available in: P:\BSP_LocalDev\Manivannan.Mathialag\zzzz My SAS Files\Presentation\PHUSE 2026\Tool\EXE Tool\dist
==== Build Completed ====
Output EXE is available under: dist\Veristat_TLF_Packager.exe
Press any key to continue . . .
```

RESULTS AND BENEFITS

TIME AND EFFICIENCY GAINS

The Veristat TLF Packager demonstrates substantial time savings compared to traditional manual packaging processes. Tasks that previously required several hours of iterative manual review—opening files individually, renaming outputs, applying bookmarks and building a TOC—can now be completed in minutes. In studies with frequent interim reviews or DSMB/DMC cycles, the tool reduces repetitive workload dramatically, enabling rapid turnaround even when outputs are regenerated multiple times per day.

Table of Contents	
Table 14.1.1.1.1: Patient Disposition - Intent to Treat Population	2
Table 14.1.1.1.2: Patient Disposition - Safety Population	4
Table 14.1.1.1.3: Patient Disposition - By Baseline Lesion Load - Intent to Treat Population	5
Table 14.1.1.1.4: Patient Disposition - By Baseline Lesion Load - Safety Population	6
Table 14.1.3.1.1: Demographics and Diagnosis Characteristics - Intent to Treat Population	7
Table 14.1.3.1.2: Demographics and Diagnosis Characteristics - Safety Population	10
Table 14.1.3.1.3: Demographics and Diagnosis Characteristics - By Baseline Lesion Load - Intent to Treat Population	13
Table 14.1.3.1.4: Demographics and Diagnosis Characteristics - By Baseline Lesion Load - Safety Population	16
Table 14.1.3.2.1: Baseline Clinical Presentation - Intent to Treat Population	19
Table 14.1.3.2.2: Baseline Clinical Presentation - Safety Population	20
Table 14.1.3.2.3: Baseline Clinical Presentation - By Baseline Lesion Load - Intent to Treat Population	22
Table 14.1.3.2.4: Baseline Clinical Presentation - By Baseline Lesion Load - Safety Population	23
Table 14.1.3.3.1: Non-ALD/CALD Medical History - Intent to Treat Population	25
Table 14.1.3.3.2: Non-ALD/CALD Medical History - Safety Population	26
Table 14.1.3.3.3: Non-ALD/CALD Medical History - By Baseline Lesion Load - Intent to Treat Population	27
Table 14.1.3.3.4: Non-ALD/CALD Medical History - By Baseline Lesion Load - Safety Population	28
Table 14.2.1.1.1: Time to Persistent Gd-enhancement - Kaplan-Meier Analysis - Intent to Treat Population	29
Table 14.2.1.1.2: Time to Persistent Gd-enhancement - Kaplan-Meier Analysis - Safety Population	30
Table 14.2.1.1.3: Time to Persistent Gd-enhancement - Kaplan-Meier Analysis - By Baseline Lesion Load - Intent to Treat Population	31
Table 14.2.1.1.4: Time to Persistent Gd-enhancement - Kaplan-Meier Analysis - By Baseline Lesion Load - Safety Population	32
Table 14.2.1.2.1: Time to HSCT or Gene Therapy - Kaplan-Meier Analysis - Intent to Treat Population	33
Table 14.2.1.2.2: Time to HSCT or Gene Therapy - Kaplan-Meier Analysis - Safety Population	34
Table 14.2.1.2.3: Time to HSCT or Gene Therapy - Kaplan-Meier Analysis - By Baseline Lesion Load - Intent to Treat Population	35
Table 14.2.1.2.4: Time to HSCT or Gene Therapy - Kaplan-Meier Analysis - By Baseline Lesion Load - Safety Population	36
Table 14.2.1.3.1: Treatment-Free Survival Analysis - Intent to Treat Population	37
Table 14.2.1.3.2: Treatment-Free Survival Analysis - Safety Population	38
Table 14.2.1.3.3: Treatment-Free Survival Analysis - By Baseline Lesion Load - Intent to Treat Population	39
Table 14.2.1.3.4: Treatment-Free Survival Analysis - By Baseline Lesion Load - Safety Population	40
Table 14.2.1.4.1: MFD-Free Survival Analysis - Intent to Treat Population	41
Table 14.2.1.4.2: MFD-Free Survival Analysis - Safety Population	42
Table 14.2.1.4.3: MFD-Free Survival Analysis - By Baseline Lesion Load - Intent to Treat Population	43
Table 14.2.1.4.4: MFD-Free Survival Analysis - By Baseline Lesion Load - Safety Population	44
Table 14.3.1.1.1: Adverse Event by MedDRA System Organ Class, Preferred Term, and Treatment Arm - Intent to Treat Population	45
Table 14.1.1.1.1: Patient Disposition - Intent to Treat Population	
Table 14.1.1.1.2: Patient Disposition - Safety Population	
Table 14.1.1.1.3: Patient Disposition - By Baseline Lesion Load - Intent to Treat Population	
Table 14.1.1.1.4: Patient Disposition - By Baseline Lesion Load - Safety Population	
Table 14.1.3.1.1: Demographics and Diagnosis Characteristics - Intent to Treat Population	
Table 14.1.3.1.2: Demographics and Diagnosis Characteristics - Safety Population	
Table 14.1.3.1.3: Demographics and Diagnosis Characteristics - By Baseline Lesion Load - Intent to Treat Population	
Table 14.1.3.1.4: Demographics and Diagnosis Characteristics - By Baseline Lesion Load - Safety Population	
Table 14.1.3.2.1: Baseline Clinical Presentation - Intent to Treat Population	
Table 14.1.3.2.2: Baseline Clinical Presentation - Safety Population	
Table 14.1.3.2.3: Baseline Clinical Presentation - By Baseline Lesion Load - Intent to Treat Population	
Table 14.1.3.2.4: Baseline Clinical Presentation - By Baseline Lesion Load - Safety Population	

QUALITY, CONSISTENCY AND REDUCED RISK

Automation significantly lowers the risk of human-driven sequencing or title mismatches. With metadata extracted directly from source files, consistency is maintained across re-runs and QC cycles. Clickable bookmarks and a structured TOC improve document usability for reviewers, statisticians, medical writers and regulatory stakeholders. Ensuring standardized formatting across packages improves audit readiness and reduces time spent on rework caused by formatting discrepancies.

IMPROVED CROSS-FUNCTIONAL COLLABORATION

Excel export functionality allows teams to share review sheets easily for sign-off or validation without accessing individual TLF files. This collaborative workflow supports remote review and ensures transparency in version control and decision tracking. The ability to include external non-SAS files such as executive summaries or medical writer-generated content further streamlines multidisciplinary deliverable assembly.

CONCLUSION

In addition to these advantages, the Veristat TLF Packager also helps standardize cross-study workflows, making it easier for new team members to onboard and quickly align with existing processes. The unified interface reduces variation in how outputs are reviewed and delivered, supporting stronger governance and smoother handovers between programming, biostats and medical-writing functions.

The tool's consistent structure for bookmarks, output ordering and TOC formatting enables faster downstream QA checks and reduces ambiguity during submissions or sponsor reviews. Its automated extraction logic also minimizes human error in title handling, which is often a major source of rework in large TLF packages.

By consolidating all extraction, review, export and packaging capabilities into a single environment, the Packager supports predictable timelines even during high-pressure deliverables such as rapid DSMB updates or interim analyses. The presence of a version-controlled GitHub repository encourages continuous improvement, allowing teams to propose enhancements, share test cases and extend the tool to evolving regulatory or sponsor requirements.

Together, these strengths position the TLF Packager as a robust, forward-looking solution that not only accelerates current workflows but also strengthens long-term operational maturity across the clinical programming ecosystem.

A publicly accessible GitHub repository is available for full source code, executable packaging instructions and demonstration materials, supporting transparency, reproducibility and community adoption.

REFERENCES

American Psychological Association. (2023). Publication Manual of the American Psychological Association (7th ed.). <https://apastyle.apa.org/>

PHUSE AI Guidance Working Group. (2025). PHUSE guidance on referencing the use of AI in PHUSE Connect papers and Working Group white papers. PHUSE.

OpenAI. (2025). ChatGPT (GPT-4o, 2 July version) [Large language model]. <https://chat.openai.com/>

ACKNOWLEDGMENTS

The author would like to thank the Veristat Biostatistics and Statistical Programming team for their support, review feedback and collaboration during the development of this tool. Appreciation is extended to colleagues involved in the BSP Harmonization initiative, whose practical inputs helped validate workflow requirements and real-world usability.

The conceptual design, Python implementation, GUI development and testing of the Veristat TLF Packager were fully authored by Manivannan M. Generative AI (ChatGPT, OpenAI, 2025) was used solely to refine written content and enhance clarity in this paper—not for algorithm design, code development or technical problem-solving.

RECOMMENDED READING

Veristat TLF Packager – Full Source Code & Executable (GitHub Repository)
https://github.com/manivannan-mathiazhagan/TLF_Packager_GUI

CONTACT INFORMATION

Your comments and questions are welcome. Please contact the author at:

Author: Manivannan Mathialagan

Company: Veristat

Email: manivannan.mathialagan@veristat.com

LinkedIn: <https://www.linkedin.com/in/manivannan-mathialagan>

GitHub Repository (Tool Source Code): [https://github.com/manivannan-mathiazhagan/TLF Packager GUI](https://github.com/manivannan-mathiazhagan/TLF_Packager_GUI)

Brand and product names are trademarks of their respective companies.