Vita Global Sciences

FROM RESEARCH TO REALITY

phuse·connect
connect·share·advance
The Clinical Data Science Conference

US 2026
22–26 March
Renaissance Austin Hotel

SM12

Automating Analysis Report Programming in SAS with VGSART: A Metadata-Driven Approach

Piyush Makkapati

Swathi Nattam

Bhargav Makkapati

Vita Global Sciences, Waltham, MA

Agenda

Introduction & Background

What is VGSART?

Overall Workflow
- Four Step Process

Functionality Achieved To Date

Testing & Validation

Scalability & Roadmap

Conclusion

Vita Global Sciences

Introduction & Background

Analysis output programming is a time-consuming part of clinical programming that requires careful attention, despite its repetitive nature across studies.

Teams often rely on copying programs forward under accelerated timelines, which can let preventable issues like a missed where condition, an outdated format, or an incorrect header label into deliverables.

VGSART was developed to shift effort from programming the same output logic and toward reviewing outputs with confidence by generating tables from metadata rather than study-specific code edits.

What is VGSART?

- **VGSART** is a metadata-driven SAS framework that standardizes and automates clinical analysis outputs.
- **User's role:**
  - Follow the user guide
  - Define table requirements in structured CSV metadata such as analysis variables, where conditions, formats, titles, footnotes, and other key table elements
- **Framework's role:**
  - Reads the metadata
  - Applies the same standardized logic across studies
  - Generates outputs that align with annotated mock shells
- **Usability goal:**
  - Enable new users to run the full table workflow
  - Minimal assistance required
  - Little to no programming knowledge needed

Vita Global Sciences

Overall Workflow

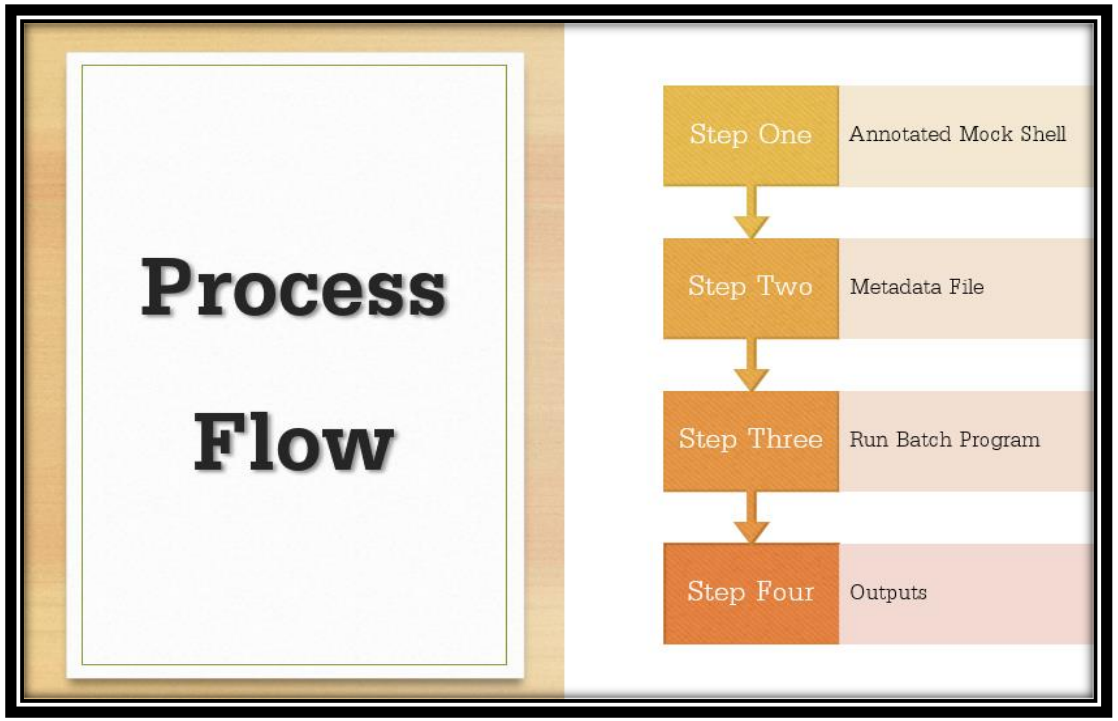


Figure 1: Process Flow

- VGSART generates analysis outputs through a clear, controlled four-step process from defining table structure to producing final outputs.
- The workflow is documented in the user guide
  - Provides detailed, step-by-step instructions for initiating tool use and generating the required outputs

Vita Global Sciences

Step 1: Annotated Mock shell

- **Annotated mock shell:**
  - Acts as a blueprint for the final output
  - Clearly defines what each table should look like before any programming begins
- **Table details:**
  - Defines table layout
  - Specifies titles and treatment group columns
  - Includes row labels and formatting
- **Row annotations:**
  - Each row is annotated with source data variables e.g., ADSL.AGE, ADSL.SEX, ADSL.RACE
  - Specifies required statistics e.g., means, frequencies
- **Overall impact:**
  - Reduces ambiguity and avoidable rework
  - Supports consistent interpretation of specifications across all team members

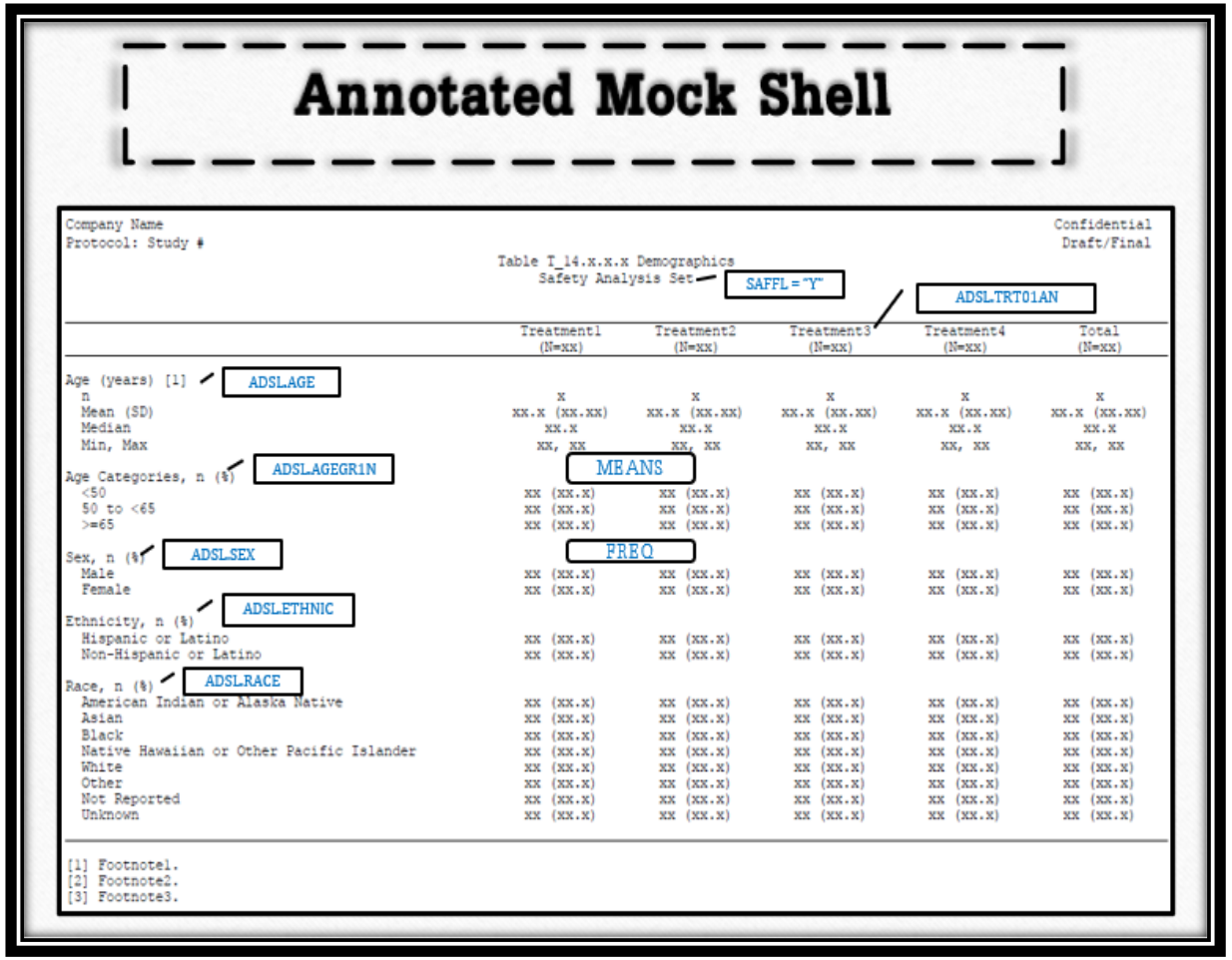
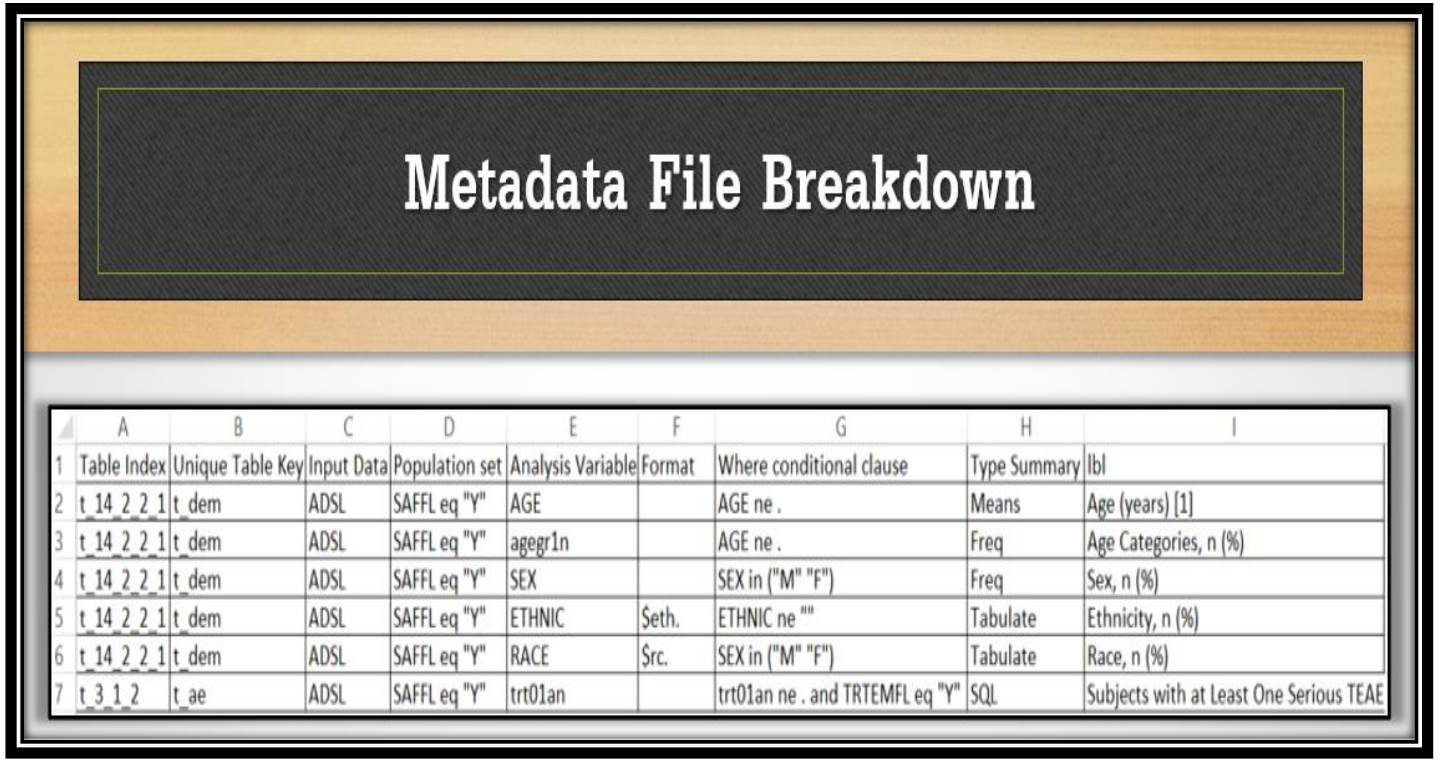


Figure 2: Annotated Mock Shell

Vita Global Sciences

Step 2: Metadata Files

- **Metadata:**
  - Central source to define and maintain all analysis definitions
- **Contents:**
  - Captures essential table requirements
    - ✓ Input datasets
    - ✓ Analysis variables
    - ✓ Conditional clauses, etc.
  - Improves transparency for end users and reviewers
- **Metadata population & maintenance:**
  - Populate by following annotated mock shells
    - ✓ Or start from an internal prefilled set (based on FDA-approved mock shells)
  - If study requirements change, update metadata directly
- **Metadata drives:**
  - How analyses run
  - How results are displayed



	A	B	C	D	E	F	G	H	I
1	Table Index	Unique Table Key	Input Data	Population set	Analysis Variable	Format	Where conditional clause	Type Summary	lbl
2	t_14_2_2_1	t_dem	ADSL	SAFFL eq "Y"	AGE		AGE ne .	Means	Age (years) [1]
3	t_14_2_2_1	t_dem	ADSL	SAFFL eq "Y"	agegr1n		AGE ne .	Freq	Age Categories, n (%)
4	t_14_2_2_1	t_dem	ADSL	SAFFL eq "Y"	SEX		SEX in ("M" "F")	Freq	Sex, n (%)
5	t_14_2_2_1	t_dem	ADSL	SAFFL eq "Y"	ETHNIC	$eth.	ETHNIC ne ""	Tabulate	Ethnicity, n (%)
6	t_14_2_2_1	t_dem	ADSL	SAFFL eq "Y"	RACE	$rc.	SEX in ("M" "F")	Tabulate	Race, n (%)
7	t_3_1_2	t_ae	ADSL	SAFFL eq "Y"	trt01an		trt01an ne . and TRTEMFL eq "Y"	SQL	Subjects with at Least One Serious TEAE

Figure 3: Metadata File Breakdown

Vita Global Sciences

Step 3: Run Batch Program

- Once metadata entry is complete, outputs are generated via "Batch Submit."
- VGSART reads the metadata and drives output generation through the macro framework in a controlled sequence.
- The user can choose to either run a specific table or multiple tables at a time depending on the need.

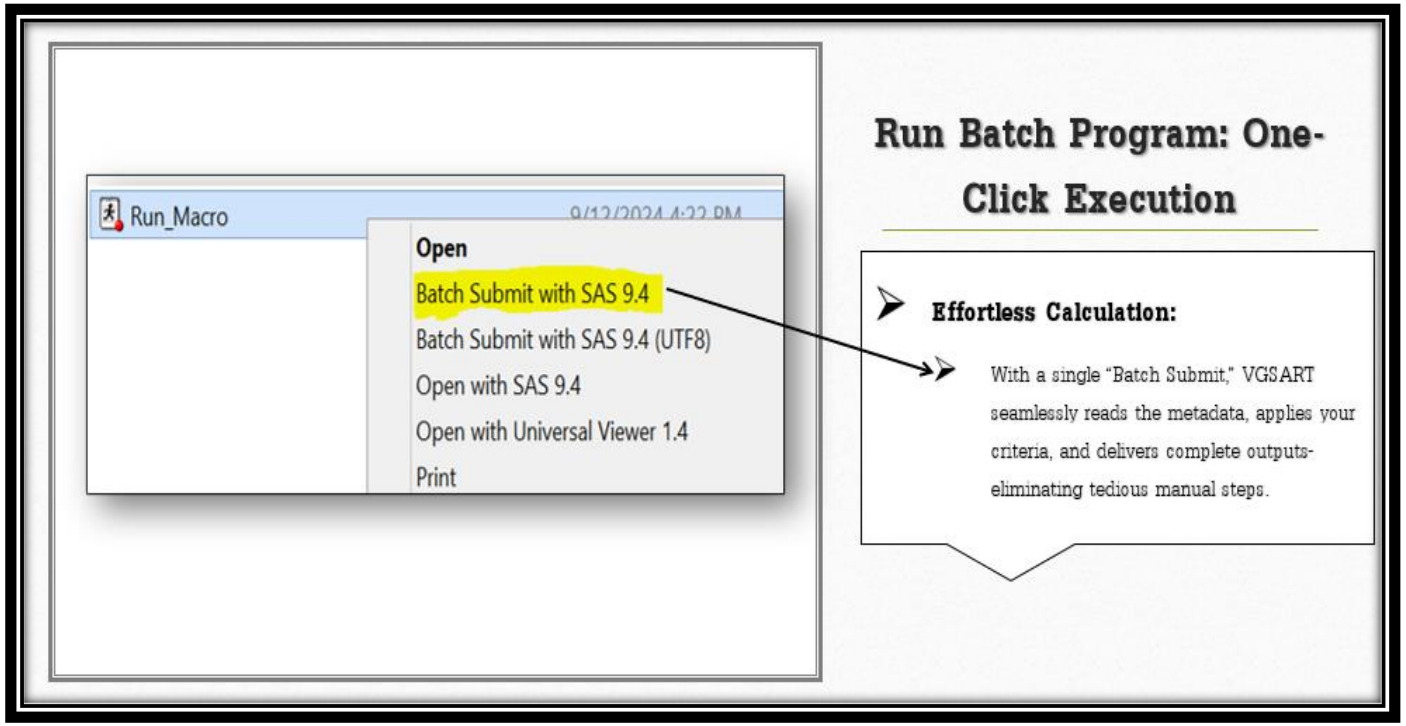


Figure 4: Batch Execution

Vita Global Sciences

Internal Execution Flow

- **Run execution:**
  - Run macro triggers the VGSART macro suite
  - Statistical methods can be specified at both table and variable levels
- **Macro Flow:**
  - **Stats macros:**
    - ✓ Categorical: counts and percentages
    - ✓ Continuous: descriptive statistics
    - ✓ SQL-based counts
  - **Pre-processing macro:**
    - ✓ Derive treatment group combinations
    - ✓ Apply where conditions and population flags
    - ✓ Calculate analysis totals (Big N)
  - **Post-processing macro:**
    - ✓ Apply sequence ordering
    - ✓ Add sub-labels
    - ✓ Calculate page counts for reporting datasets
  - **Reporting macro:**
    - ✓ Generates the final layout based on reporting metadata options

```
%include "&macroPath\freq_Macro.sas";
%include "&macroPath\means_Macro.sas";
%include "&macroPath\sql_Macro.sas";
%include "&macroPath\R_macros.sas";
%include "&macroPath\processing.sas";
```

Figure 8: Set of integrated macros

Preprocessing and Post processing

Program	Macro	Associated Tables	Purpose
Processing	%PP(trt_var=, cmb_trts=, display_trts=)	ALL	Preprocesses treatment groups by combining as specified within parameters, applies where conditions & page flags, calculates total counts (BigN) for each group, and creates preprocessed aDaM datasets ready for analysis.
	%POSTP	ALL	Applies sequence orders, sub labels, page count calculations for the final datasets used by reporting macros.

Figure 9: Processing Macro Descriptions

Vita Global Sciences

Step 4: Outputs

- Outputs can be generated in RTF or PDF and are ready to be validated before submission
- VGSART has successfully produced a comprehensive set of safety and clinical summary tables, including:
  - TEAEs and AE summaries by SOC/PT with standard breakdowns by seriousness, relatedness, severity, and dose
  - Standard safety summaries such as demographics, subject disposition, medical history, and protocol deviations
- Supports more advanced safety outputs, including:
  - Change-from-baseline summaries for vital signs, physical examinations, and ECG by visit and dose
  - Shift tables for selected laboratory parameters

Softworld
Protocol: Study Name

Confidential
VGSART Draft

Table 14.7.4.1 Shift Table of Select Laboratory Parameters by Dose
Safety Analysis Set

Laboratory Test Dose Group Worst Toxicity Grade	Baseline Toxicity Grade				
	Grade 0	Grade 1	Grade 2	Grade 3	Grade 4
Alanine Aminotransferase (U/L)					
TRT 100 mg (N=6)	3	3	0	0	0
Grade 0	1 (50.0)	2 (100.0)	0	0	0
Grade 1	1 (50.0)	0	0	0	0
Grade 2	0	0	0	0	0
Grade 3	0	0	0	0	0
Grade 4	0	0	0	0	0
TRT 200 mg (N=8)	3	5	0	0	0
Grade 0	1 (50.0)	9 (90.0)	0	0	0
Grade 1	1 (50.0)	1 (10.0)	0	0	0
Grade 2	0	0	0	0	0
Grade 3	0	0	0	0	0
Grade 4	0	0	0	0	0
Alkaline Phosphatase (U/L)					
TRT 100 mg (N=6)	6	0	0	0	0
Grade 0	4 (100.0)	0	0	0	0
Grade 1	0	0	0	0	0
Grade 2	0	0	0	0	0
Grade 3	0	0	0	0	0
Grade 4	0	0	0	0	0
TRT 200 mg (N=8)	7	1	0	0	0
Grade 0	10 (90.9)	1 (100.0)	0	0	0
Grade 1	1 (9.1)	0	0	0	0
Grade 2	0	0	0	0	0
Grade 3	0	0	0	0	0
Grade 4	0	0	0	0	0

Baseline is defined as the last available measurement taken prior to the first dose of study drug.
National Cancer Institute (NCI) Common Terminology Criteria for Adverse Events (CTCAE) Version 5.0 used to grade toxicities.

Figure 6: Shift Table of Select Lab Parameters by Dose

Vita Global Sciences

Functionality Achieved to Date

- VGSART has progressed to the point where it can automate majority of safety analysis tables in a consistent, efficient way.
- The framework supports various unique templates across multiple table families
  - Including variations by treatment groups, columns, populations, and where conditions
  - Does not limit how many tables a user can generate at a time
- Consists of SAS macro programs that take care of generating the analysis outputs
- Functional GUI using R Shiny has been created to support routine work while using VGSART
  - Still being tested and refined
  - Supports importing/editing metadata with embedded tooltips
  - Snapshot creation for traceability
  - SAS execution directly within the interface

Functionality Achieved to Date (GUI Visuals)

Figure 10: Snapshot Creation & Selection

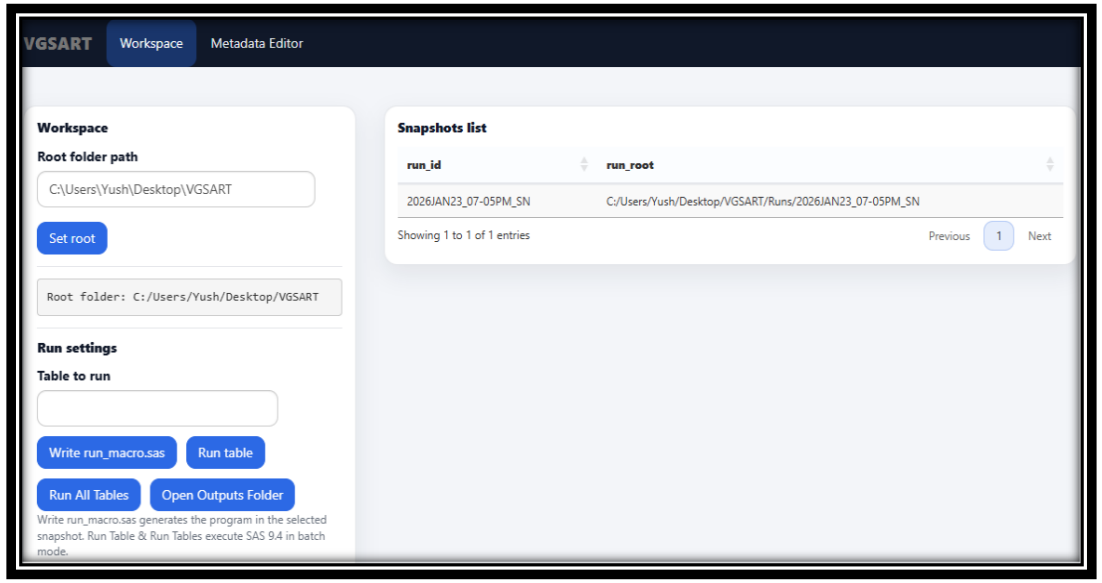


Figure 11: VGSART Workspace Tab

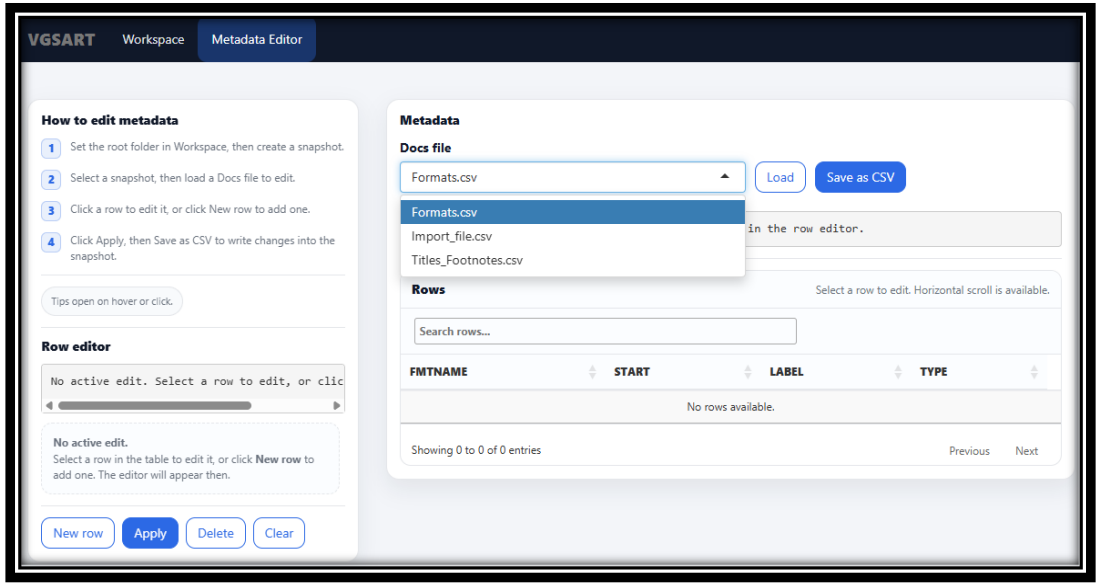
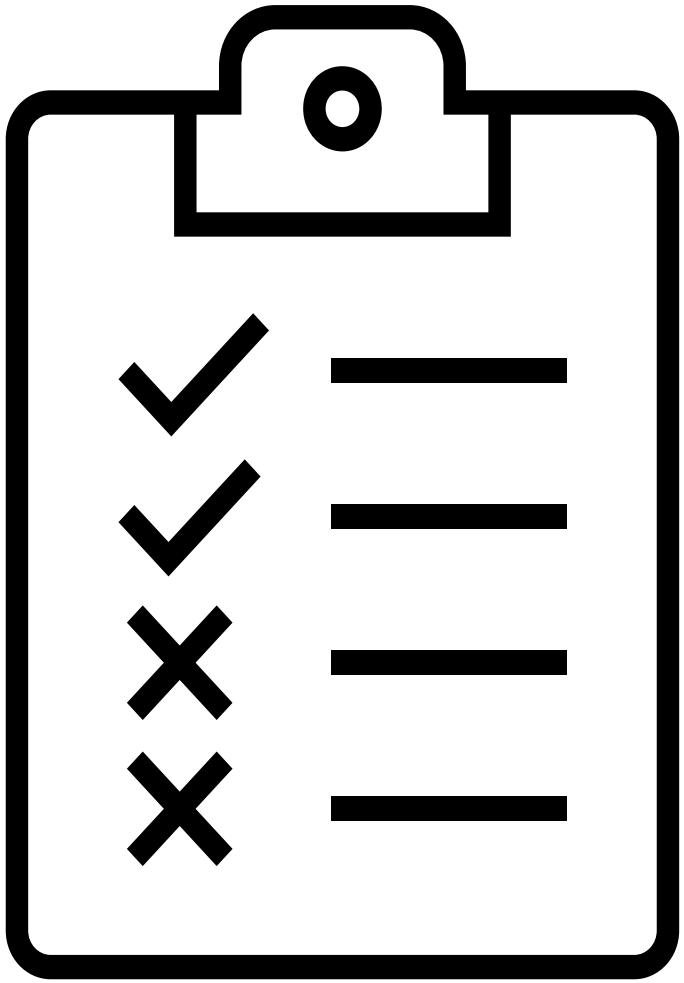


Figure 12: Metadata Files Importing, Editing, & Saving

Vita Global Sciences

Testing & Validation

- **Goal:** Reduce programming time
  - Maintain the same quality outputs
  - Match QC reviewed deliverables with no mismatches in counts, presentation, or table structure
- Tested using a dedicated VGSART folder structure
  - Example FDA-submitted studies as reference
- For each test study -> confirmed number of hours for completion with the project manager
  - Used the same ADaM datasets & populated metadata to match mock shells
  - Compared against verified outputs
  - Tracked amount of time saved
- To test user friendliness: provided access to two users (one with TFL experience and one who primarily worked on SDTMs)
  - **Results in comparison to standard programming:**
    - ✓ **Note:** Both users only followed user guide
    - ✓ TFL programmer matched expected completion time of 25% less time
    - ✓ Non-TFL programmer completed in 18% to ~20% less time



Vita Global Sciences

Scalability & Roadmap

Next phase focuses on expanding what the framework can produce while keeping the same metadata-driven approach.

Coverage: Current priority is to target even more complicated tables

- Expand into efficacy tables
- Figures are a long-term target

Efficiency Goal: ~30-40% reduction in programming time and errors

- Improve macro efficiency
- Simplify metadata fields without reducing flexibility

Usability: Improve day-to-day usability

- Add more features to the GUI
- End goal: any user new to TFL programming should be able to follow along with the user guide
  - Generate tables independently

Future-Proof: Expand to adopt R

Vita Global Sciences

Conclusion

Key Takeaways

Practical
Metadata-driven workflow that generates safety tables from structured metadata instead of study specific program edits.

Familiar
Built around how the team already works by using SAS and structured metadata to keep the workflow familiar, transparent, and easy to review.

Efficient
Matched QC reviewed deliverables and reduced programming time by about 25% and shifted focus towards QC process.

Main takeaway: When table requirements are captured in metadata and execution is standardized, teams can spend less time rewriting code and more time delivering outputs they fully trust.

Vita Global Sciences

Thank You!

Pmakkapati@softworldinc.com Snattam@softworldinc.com
bmakkapati@softworldinc.com

Vita Global Sciences

Questions?

Vita Global Sciences