PharmaSUG 2025 - Paper PO-208

Our Second Brain: A Guide to Building a Note System for New SAS Programmers in Clinical Trials

Zeyu Li, Arrowhead Pharmaceuticals

ABSTRACT

As a clinical trial SAS programmer, the job may at first seem overwhelming with innumerable details necessary to apply programs and analyses. Added to this, months later you could be asked by statisticians or data managers to comment on issues within the data of previous studies. With numerous studies to handle each day, remembering every piece of information from every program and project is not feasible. For that reason, there is a strong need for having an explicit and well-structured note system.

A well-designed note system is your own personal knowledge base, from which you can recall essential information about studies instantly, sometimes even years after they were completed. It also makes you more efficient by providing you with immediate access to reference codes, SAS shortcuts, and reusable templates, thereby avoiding having to do things twice.

INTRODUCTION

Working as SAS programmer in clinical trial, we are constantly confronted with new knowledge—anything from study-related information and regulation mandates to industry insights and technical nuance. Effective management of all this knowledge is necessary to remain current and prevent errors in our output. Without systems in place, excellent ideas are soon lost in scattered notes, emails, or memory.

In this article, I suggest a notetaking system for SAS programmers in clinical trials. I will illustrate the key categories of information worth writing down, the structure for easy retrieval, and benefits in adopting a systematic approach to notetaking. My digital notebook is structured into categories covering the essential aspects of our workday:

- **Day-to-Day Organization** – A repository for daily task list, meeting minutes, and overview of ongoing tasks.

- **Study Overview** – An overview of every clinical study, including study design, key definition, statistical analysis plan, and variables derivation.

- **SAS Reference** – A repository of commonly used SAS code syntax, templates, macros.

- **CDISC Knowledge** – Regulations and examples related to CDISC standards, so that compliance and consistency are achieved.

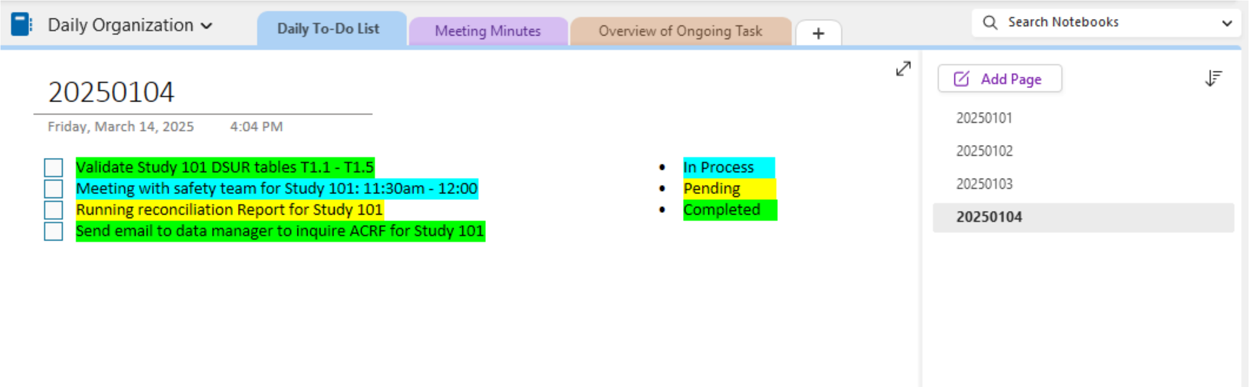
By utilizing a well-organized digital notebook, SAS programmers can increase productivity, eliminate redundancy, and ensure better knowledge retention. This paper will give us a blueprint for creating a personal knowledge system that is our "second brain"—a valuable vault that grows with our professional career and solidifies our statistical programming skills.

DAILY ORGANIZATION

My first section in my notebook is the day-to-day organization, which I use to keep records of tasks, due dates, and commitments. It is the section I most commonly use, as it is my central hub for the day-to-day list of activities to be completed, minutes from meetings, and ongoing work on every study.

One of the biggest challenges to SAS programmers involved in clinical trials is juggling multiple projects with various timelines and deliverables. Missing a deadline due to poor organization can have disastrous consequences, affecting not only individual productivity but also the study timeline. Keeping all the tasks in one place will ensure that the programmer will be able to see the priorities clearly, reducing the chances of missing critical deadlines.

I divide my section on daily organization into three main parts to reduce this risk. First, my to-do list includes checklists for daily and weekly tasks that are arranged according to urgency and priority. Color-coded statuses like "Pending," "In Progress," and "Completed" make it easy for me to quickly evaluate my workload. Second, my meeting notes include brief synopses of the main points discussed as well as any necessary follow-up work. Lastly, I can sort assignments by study, record any difficulties I have had, and record important study updates that could have an impact on dataset programming and document preparation with the aid of my study-specific task tracker. I can manage my workload more effectively, meet deadlines, and keep my daily tasks clear by keeping all this information in one location.

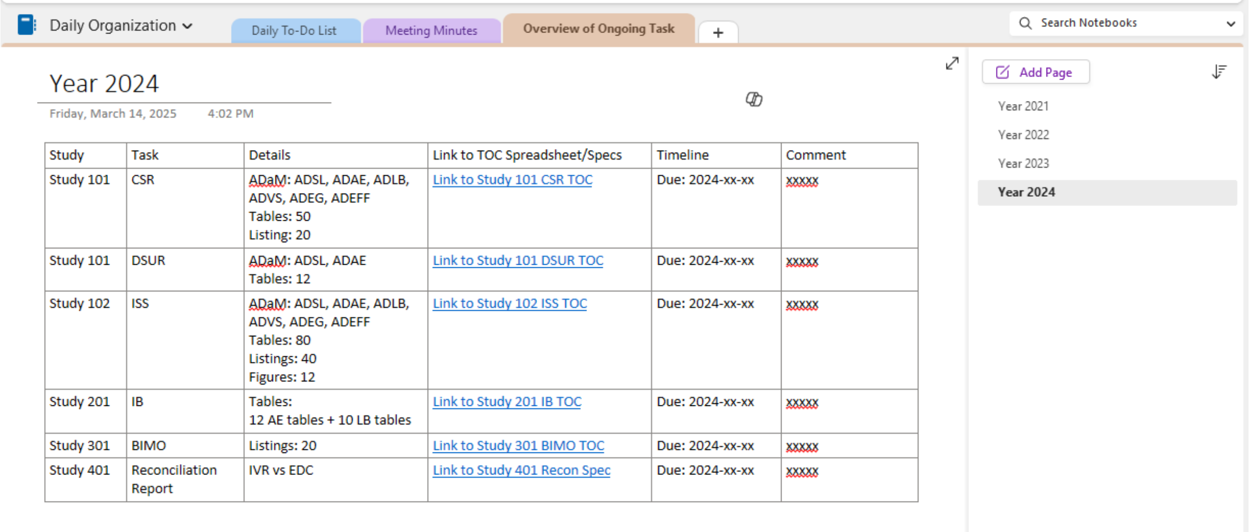


20250104

Friday, March 14, 2025 4:04 PM

- □ Validate Study 101 DSUR tables T1.1 - T1.5
- □ Meeting with safety team for Study 101: 11:30am - 12:00
- □ Running reconciliation Report for Study 101
- □ Send email to data manager to inquire ACRF for Study 101

- In Process
- Pending
- Completed



Year 2024

Friday, March 14, 2025 4:02 PM

Study	Task	Details	Link to TOC Spreadsheet/Specs	Timeline	Comment
Study 101	CSR	ADaM: ADSL, ADAE, ADLB, ADVS, ADEG, ADEFF Tables: 50 Listing: 20	Link to Study 101 CSR TOC	Due: 2024-xx-xx	xxxxx
Study 101	DSUR	ADaM: ADSL, ADAE Tables: 12	Link to Study 101 DSUR TOC	Due: 2024-xx-xx	xxxxx
Study 102	ISS	ADaM: ADSL, ADAE, ADLB, ADVS, ADEG, ADEFF Tables: 80 Listings: 40 Figures: 12	Link to Study 102 ISS TOC	Due: 2024-xx-xx	xxxxx
Study 201	IB	Tables: 12 AE tables + 10 LB tables	Link to Study 201 IB TOC	Due: 2024-xx-xx	xxxxx
Study 301	BIMO	Listings: 20	Link to Study 301 BIMO TOC	Due: 2024-xx-xx	xxxxx
Study 401	Reconciliation Report	IVR vs EDC	Link to Study 401 Recon Spec	Due: 2024-xx-xx	xxxxx

STUDY OVERVIEW

When doing programming tasks in clinical trials, we typically need to extract critical information from several documents, such as the Statistical Analysis Plan (SAP), Protocol, and Annotated Case Report Form (ACRF). These documents are typically long and contain several sections unrelated directly to programming. Manually scanning them each time we need some information may be time-consuming and wasteful.

To make this easier, I have a space in my notebook reserved for the Study Overview, and I methodically note the most useful information from these documents. Instant reference is made available, and redundancy in searching is removed, which saves time significantly while writing and validating SAS programs.

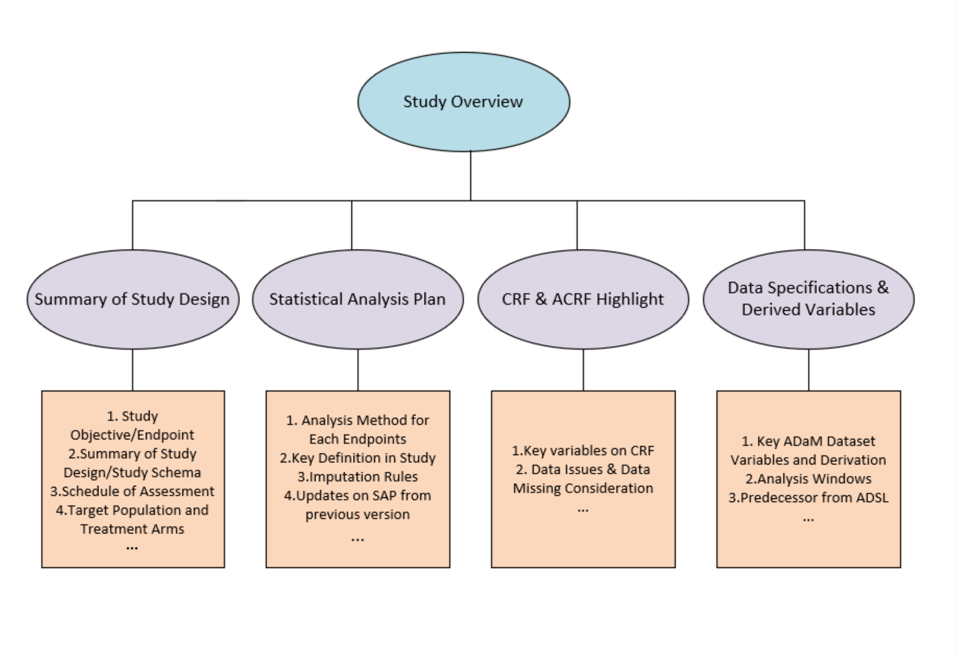
Without having to sort through extensive procedures, Statistical Analysis Plans (SAPs), or Annotated Case Report Forms (ACRFs), the Study Overview section of my notebook is indispensable for rapidly accessing vital information from study documents. Four main components make up this section, which guarantees that important study information is readily available when programming and validating data.

The first element is the high-level summary of the goal, target population, and treatment arms of the study found in the Summary of Study Objective and Study Design. Along with the general study design and schema, this overview also covers specifics on primary, secondary, and exploratory endpoints. Knowing this helps me to rapidly grasp the structure of the study without constantly reviewing the protocol.

The second component is Key SAP Information, where I document the primary analysis methods and statistical models outlined in the SAP. This section also includes notes on any special derivations or calculations required for analysis datasets. By keeping these details organized, I can ensure that my programming aligns with the study's pre-specified analysis plan.

The final component is Data Specifications & Derived Variables, which includes key dataset variables, notes on imputations, and CDISC compliance considerations. This section ensures that derived variables are documented clearly, reducing inconsistencies and facilitating adherence to regulatory standards.

By structuring my Study Overview section in this way, I can efficiently reference study-specific details, minimize redundant searches, and maintain consistency in my programming approach.

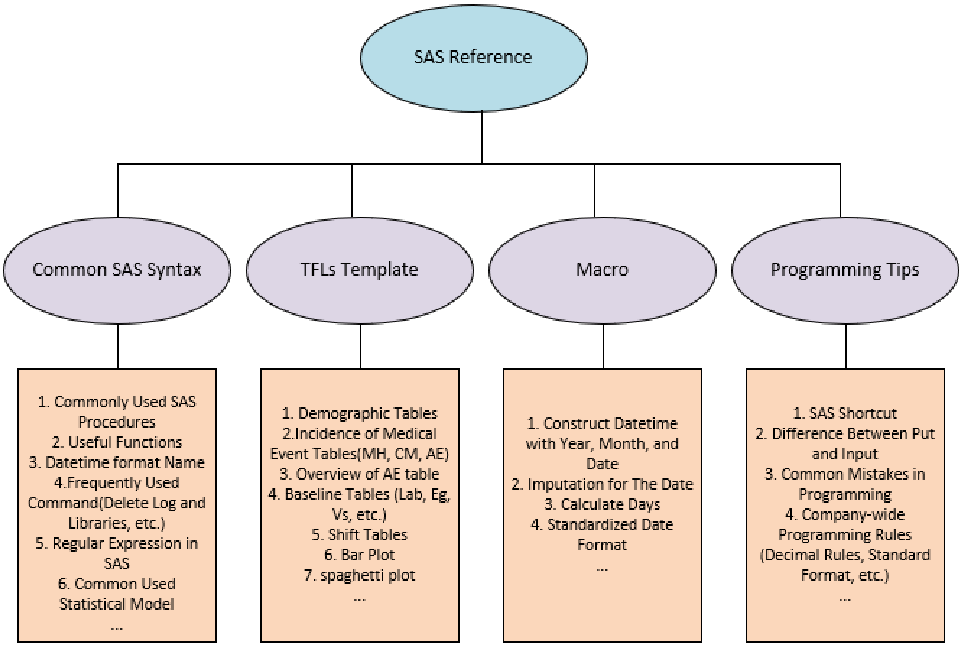

Study Overview
Summary of Study Design
Statistical Analysis Plan
CRF & ACRF Highlight
Data Specifications & Derived Variables
1. Study Objective/Endpoint
2.Summary of Study Design/Study Schema
3.Schedule of Assessment
4.Target Population and Treatment Arms
...
1. Analysis Method for Each Endpoints
2.Key Definition in Study
3.Imputation Rules
4.Updates on SAP from previous version
...
1.Key variables on CRF
2. Data Issues & Data Missing Consideration
...
1. Key ADaM Dataset Variables and Derivation
2.Analysis Windows
3.Predecessor from ADSL
...

SAS REFERENCE

Statistical programming efficiency is best optimized by keeping a tidy SAS Reference page in your electronic notebook. Such a page is your own repository of knowledge with reusable code, programming tips, and templates where programmers save themselves tedious programming effort along with accelerating their work through efficiency. Because such an "SAS vault" is built up over time, more of the code do not need to be rewritten for each Table, Listing, and Figure (TLF) or each dataset, but instead use existing material already created:

The utility of such a reference section is in large part dependent on how it is organized. The key to maximizing availability is to organize material in some way that is sensible. I divide this section in my notebook into four broad categories: Commonly Used SAS Syntax, TFL Templates, Macros, and Programming Tips. The Commonly Used SAS Syntax section includes frequently used commands, such as data manipulation techniques, statistical procedures, and regular expression, etc. Having these at hand prevents wastage of time in programming for repetitive tasks. TFL Templates section maintains table, list, and figure designs for consistency in various studies and preventing repeated programming. Macros subcategory maintains macro programs as entities which can be reused again and again and perform repetitive tasks automatically without any errors caused by manual intervention. Lastly, the Programming Tips section keeps expertise, best practices, and troubleshooting advice handy in solving common issues that arise during clinical programming.

By structuring the SAS Reference section this way, programmers can instantly locate information they require, significantly improving workflow effectiveness. Ultimately, this meticulously organized library is a treasure trove, enabling efficient knowledge transfer, rapid programming execution, and problem-solving improvement.


SAS Reference
Common SAS Syntax
TFLs Template
Macro
Programming Tips
1. Commonly Used SAS Procedures
2. Useful Functions
3. Datetime format Name
4.Frequently Used Command(Delete Log and Libraries, etc.)
5. Regular Expression in SAS
6. Common Used Statistical Model
...
1. Demographic Tables
2.Incidence of Medical Event Tables(MH, CM, AE)
3. Overview of AE table
4. Baseline Tables (Lab, Eg, Vs, etc.)
5. Shift Tables
6. Bar Plot
7. spaghetti plot
...
1. Construct Datetime with Year, Month, and Date
2. Imputation for The Date
3. Calculate Days
4. Standardized Date Format
...
1. SAS Shortcut
2. Difference Between Put and Input
3. Common Mistakes in Programming
4. Company-wide Programming Rules (Decimal Rules, Standard Format, etc.)
...

CDISC KNOWLEDGE

CDISC (Clinical Data Interchange Standards Consortium) standards are one of the most crucial aspects of clinical trial programming. These standards ensure that datasets submitted to regulatory agencies, such as the FDA, comply with a formal and uniform structure, facilitating efficient review and approval processes. CDISC standard compliance is among the drivers of the quality and integrity of a study's deliverables. Failure to comply with these guidelines has the potential to cause costly delays, additional programming work, and ultimately non-compliance issues that may hinder regulatory submission.

Having a dedicated CDISC Knowledge section in an electronic notebook is worth its weight in gold for clinical programmers. In my notebook, this section is a reference base of frequent Pinnacle 21 validation problems, reasons, and ADaM and SDTM dataset best-practice resolutions. Having these frequent problems with step-by-step solutions helps programmers avoid errors from happening in the first place. I also have scribbles on frequently referenced CDISC implementation guides, dataset specs, and examples of nicely formatted datasets.

By actively authoring CDISC data, developers can have compliant datasets from the start, avoiding expensive rework and simplifying a successful and efficient regulatory submission process. In the long term, this structured approach optimizes individual and team productivity, which in turn contributes to delivering higher-quality clinical trial deliverables.

CONCLUSION

Maintaining a tidy digital notebook is a good habit for SAS programmers involved in clinical trials. With the increasing complexities of managing more than one study, maintaining CDISC standards, and ensuring statistical programming success, keeping an organized knowledge warehouse can significantly decrease anxiety levels and improve productivity when we are approaching the deadline. By providing a central location for daily planning, SAS reference data, reusable templates, and CDISC guidelines, the programmer can easily locate helpful information, automate the workflow, and avoid redundant efforts.

REFERENCES

Xia, Jeff, and Varughese, Mary. “Microsoft OneNote: A Treasure Box for Managers and Programmers.” *Proceedings of the PharmaSUG Conference*, Rahway, NJ, and North Wales, PA: Merck & Co., Inc.

CDISC, https://www.cdisc.org/

Pinnacle 21, https://www.pinnacle21.com/

CONTACT INFORMATION

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

Name: Zeyu Li
Enterprise: Arrowhead Pharmaceuticals
Phone: +1 (626) 788-8176
E-mail: zeli@arrowheadpharma.com