



AUTOMATED CHARACTER LENGTH AUDITS TO OPTIMIZE SUPPVAR INTEGRATION



Ram Mohan Konda
Navitas Life Sciences
21 February, 2026



[Website](#)



[LinkedIn](#)



AGENDA

1

**INTRODUCTION AND NEED FOR
AUTOMATION**

6

**REGULATORY ALIGNMENT AND
VALIDATION SUMMARY**

2

CASE STUDY OVERVIEW

7

EXCEPTIONAL HANDLING

3

MACRO ARCHITECTURE OVERVIEW

8

**CHALLENGES AND FUTURE
DIRECTIONS**

4

**AUTOMATED FLAGGING AND
REPORT GENERATION**

9

CONCLUSION AND Q&A

5

**KEY RESULTS AND
PERFORMANCE IMPACT**

STUDY OVERVIEW AND OBJECTIVES

PURPOSE

To automate auditing of character variable lengths across clinical trial datasets and identify variables exceeding 200 characters to ensure SUPPVAR compliance.

OBJECTIVES

- Reduce manual review effort.
- Enhance data accuracy and traceability.
- Enable regulatory-compliant data submissions.
- Ensure consistent variable length thresholds across domains.

WHY AUTOMATION?



Audit Variable
Lengths Fast



No Manual Checks
Needed



Early SUPPVAR
Compliance



Avoid Human
Errors



Early Flagging of
SDTM Issues



Dynamic Library
Scanning



Reusable Across
Studies



Faster Reporting &
Docs



Easy SUPPVAR
Preparation

CASE STUDY

DRY RUN 1 | INITIAL AUDIT OF CHARACTER LENGTHS

Title: Dry Run 2: Cleanup and Splitting Long Values

Background:

- Phase III oncology trial data had multiple character variables exceeding 200 characters.
- Risk of data truncation and loss, impacting data integrity and compliance.
- Challenge: Need to identify and handle long character variables efficiently across 32 datasets.
- Example: Long text value (over 200 characters):

Dataset	Variable	Example Value (Truncated)	Length
AE	AETERM	Severe headache with dizziness and nausea lasting over 3 days, accompanied by blurred vision, sensitivity to light, sound, and movement, coupled with a general sense of fatigue and inability to perform daily activities without significant discomfort or worsening of the existing symptoms.	288
CM	CMTRT	Extended-release acetaminophen and codeine combination tablets, containing 300mg acetaminophen and 30mg codeine phosphate, administered twice daily for chronic lower back pain management as part of a multi-modal recovery and therapy plan.	238
DV	DVTERM	Implantable cardioverter-defibrillator device with remote monitoring feature, including serial number and history of previous discharge events, programmed for ventricular tachycardia detection and shock delivery protocols.	228

CASE STUDY

DRY RUN 2 | CLEANUP AND SPLITTING LONG VALUES

Approach:

- Trimmed AETERM to 200 characters.
- Moved overflow text to AE supplemental dataset variable AETERM_Overflow in SUPPAE.

Example:

- Trimmed AETERM = " Severe headache with dizziness and nausea lasting over 3 days, accompanied by blurred vision, sensitivity to light, sound, and movement, coupled with a general sense of fatigue and inability to " (195 chars)
- AETERM_Overflow = " perform daily activities without significant discomfort or worsening of the existing symptoms." (remaining text)

Impact:

- Flagged variables reduced from 3 to 0.
- Saved review time and ensured data integrity.

Effort:

- 20 hours programming + 10 hours testing.
- Saved 1 hour per repetitive study.

MACRO ARCHITECTURE OVERVIEW



Parameter
Validation



Input Checks



Library
Verification



Dataset
Scanning



Threshold
Comparison



Exceptional
Handling



Flag Aggregation

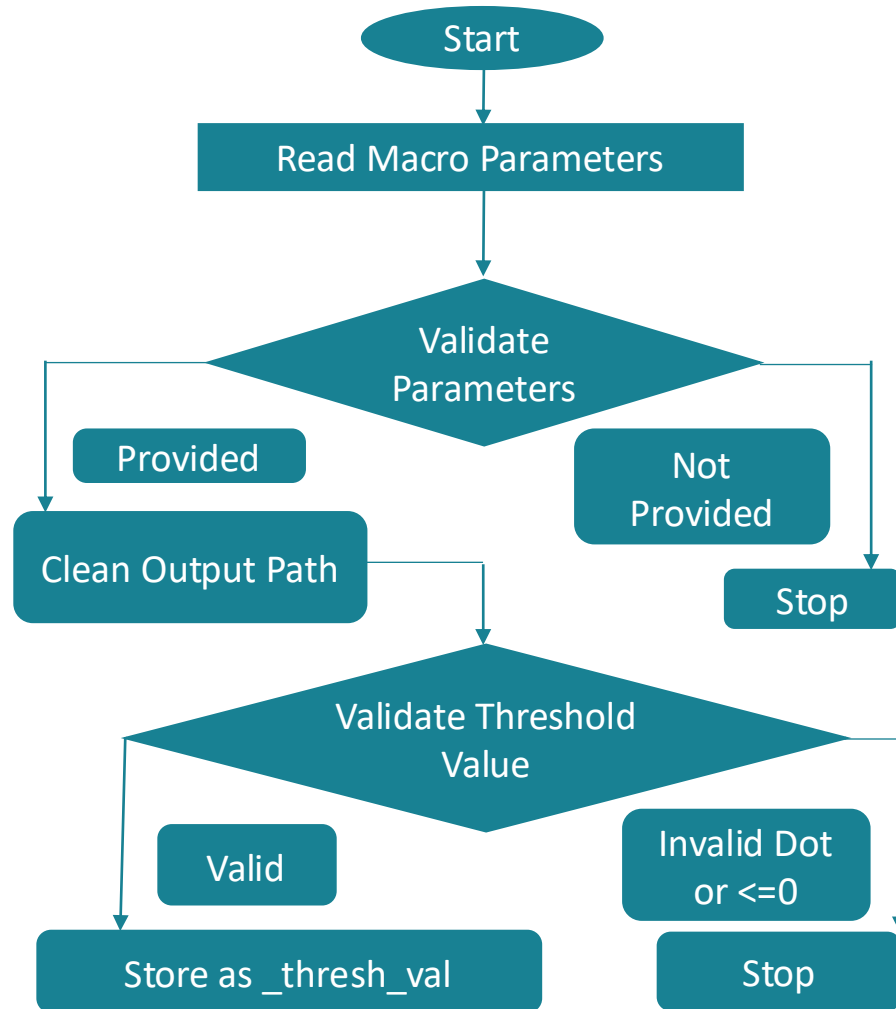


Excel Report
Generation



Macro
Completion

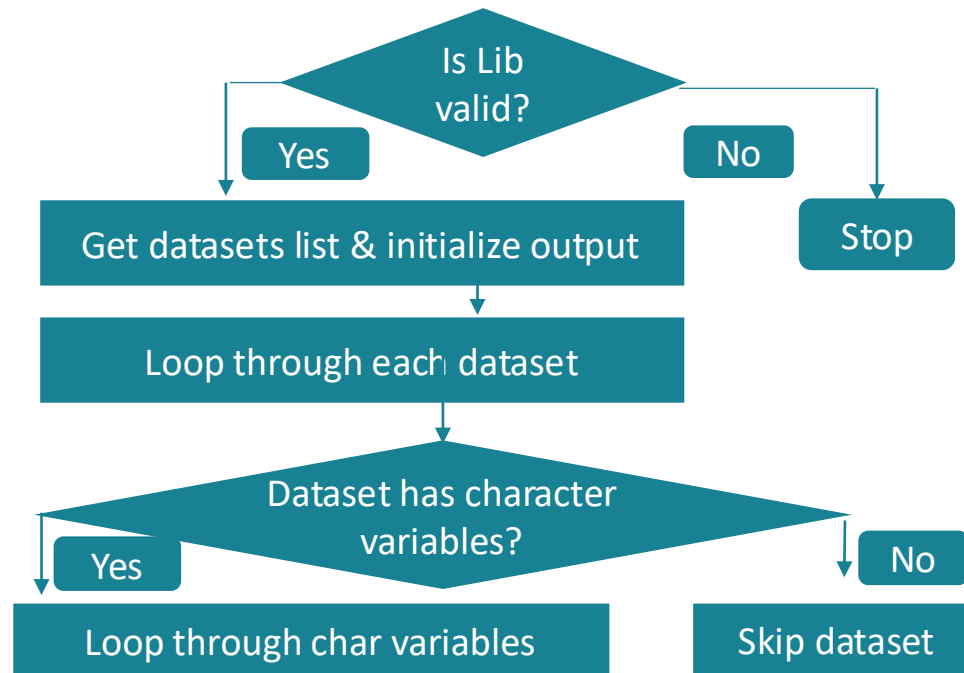
MACRO INITIALIZATION AND PARAMETER VALIDATION



```

%macro audit_all_char_length(
    lib=mylib,
    outpath=%sysfunc(pathname(work)),
    threshold=200,
    excelout=YES,
    outlib=work,
    outds=all_flagged_vars);
%put NOTE: Starting audit_all_char_length macro...;
%put NOTE- LIB=&lib OUTPATH=&outpath THRESHOLD=&threshold
            EXCELOUT=&excelout OUTLIB=&outlib OUTDS=&outds;
%local dslist i ds charlist j var maxlen outpath_clean nprocessed nflagged
            datetime_stamp filename_excel _thresh_val _error_flag;
%if %length(&lib)=0 %then %do;
    %put ERROR: LIB= is required.; %return; %end;
%if %length(&outpath)=0 %then %do;
    %put ERROR: OUTPATH= is required.; %return; %end;
%if %length(&outlib)=0 %then %do;
    %put ERROR: OUTLIB= is required.; %return; %end;
%if %length(&outds)=0 %then %do;
    %put ERROR: OUTDS= is required.; %return; %end;
%let outpath_clean = %sysfunc(dequote(&outpath));
/* Validate threshold */
%let _error_flag=0;
data _null_;
    _v = input(symget("threshold"), best.);
    if _v <= 0 or _v = . then call symputx("_error_flag", 1);
    else call symputx("_thresh_val", ceil(_v));
run;
%if &_error_flag = 1 %then %do;
    %put ERROR: THRESHOLD must be a positive integer.;
    %return;%end;
%let threshold=&_thresh_val;
  
```

LIBRARY VALIDATION + GET DATASET LIST + GET CHARACTER VARIABLES



```

%if %sysfunc(libref(&lib)) ne 0 %then %do;
    %put ERROR: Library &lib is not assigned.;
    %return;
%end;

proc sql noprint;
select memname
into :dslist separated by ' '
from dictionary.tables
where libname="%upcase(&lib)";
quit;

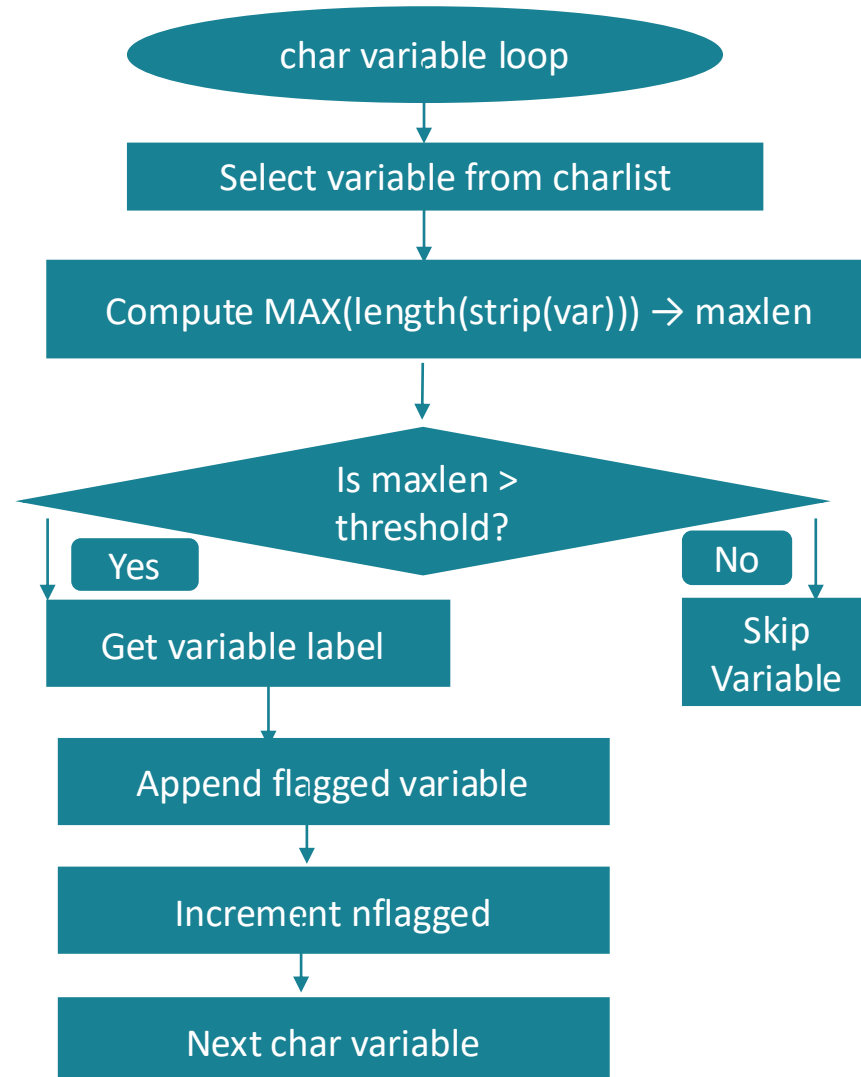
%let nprocessed=0; %let nflagged=0;
data &outlib..&outds;
    length Dataset $128 Variable $128 MaxValueLength 8 VarLabel $200;
    call missing(of _all_); stop;
run;

%let count=%sysfunc(countw(&dslist));
%do i=1 %to &count;
%let ds=%scan(&dslist,&i);
%let nprocessed = %eval(&nprocessed + 1);

proc sql noprint;
select name
into :charlist separated by ' '
from dictionary.columns
where libname="%upcase(&lib)" and memname="%upcase(&ds)" and type="char";
quit;

%if "&charlist" = "" %then %goto next_ds;
%let varcount=%sysfunc(countw(&charlist));
  
```

COMPUTE MAX VALUE LENGTHS + FLAG VARIABLES



```

%do j=1 %to &varcount;
%let _vlabel = ;
%let var=%scan(&charlist,&j);
proc sql noprint;
  select max(length(strip(&var))) into :maxlen
  from &lib..&ds;
quit;

%if %sysevalf(&maxlen > &threshold) %then %do;

proc sql noprint;
  select coalesce(label,"")
  into :_vlabel trimmed
  from dictionary.columns
  where libname="%upcase(&lib)"
  and memname="%upcase(&ds)"
  and name="%upcase(&var)";
quit;

data flagged_tmp;
  length Dataset $128 Variable $128 MaxValueLength 8 VarLabel $200;
  Dataset = "&lib..&ds";
  Variable = "&var";
  MaxValueLength = &maxlen;
  VarLabel=symget('_vlabel');

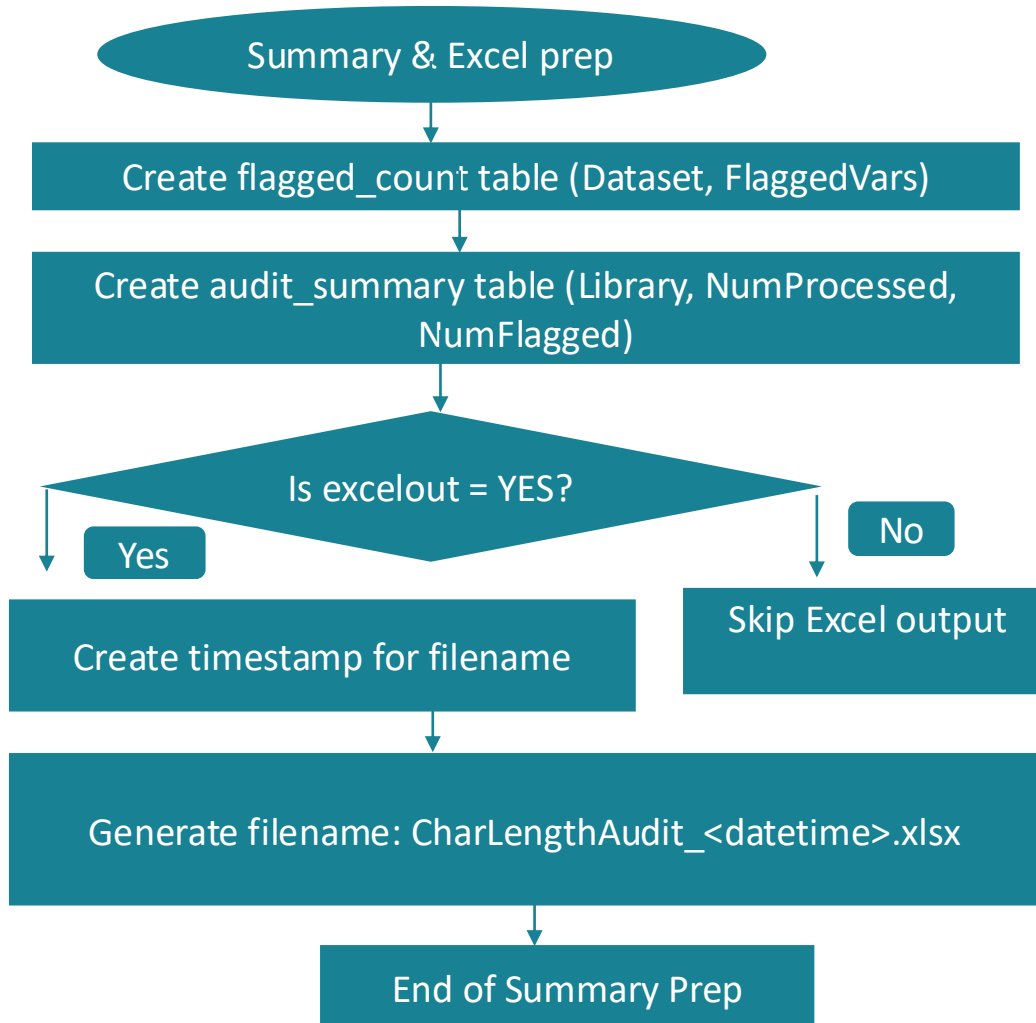
run;

proc append base=&outlib..&outds data=flagged_tmp force; run;

%let nflagged = %eval(&nflagged + 1); %end;
%end;

%next_ds;
%end;
  
```

SUMMARY TABLE AND EXCEL REPORT CREATION



```

proc sql;
create table flagged_count as
select Dataset, count(*) as FlaggedVars
from &outlib.&outds
group by Dataset;
quit;

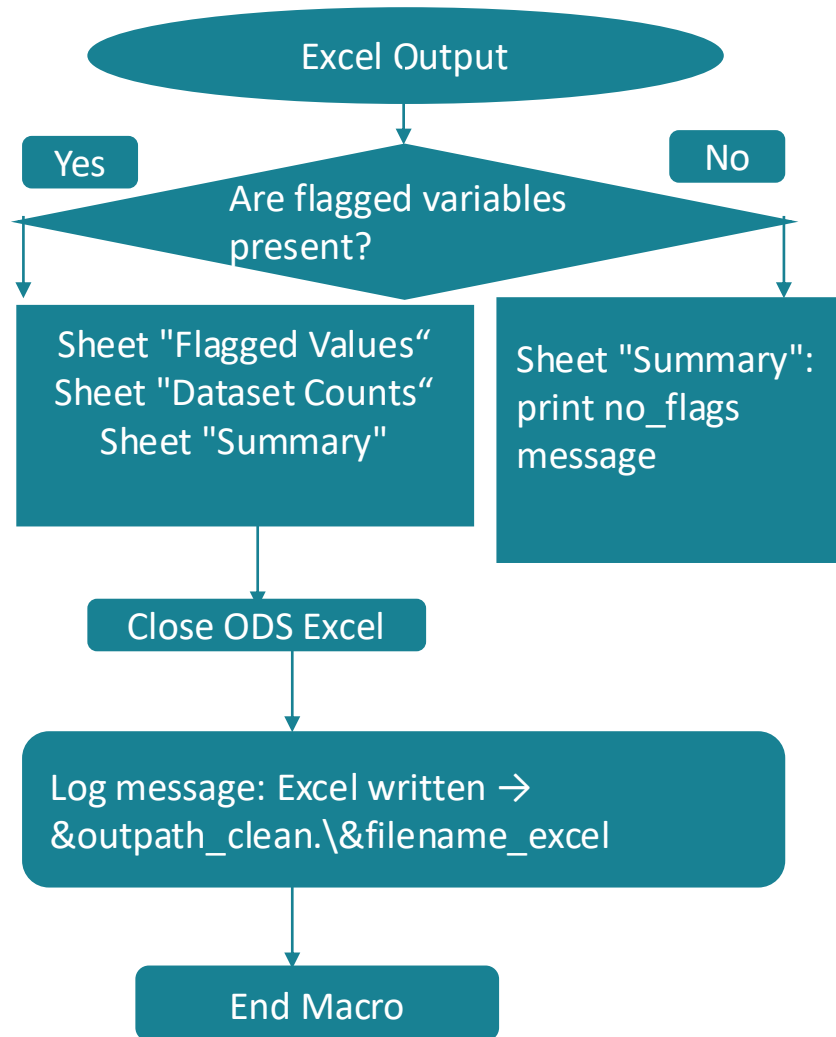
data audit_summary;
length Library $40 NumProcessed NumFlagged 8;
Library="&lib";
NumProcessed=&nprocessed;
NumFlagged=&nflagged;
run;

%if %upcase(&excelout)=YES %then %do;

data _null_;
call symputx("datetime_stamp", compress(put(datetime(), datetime20.), ":- "));
run;

%let filename_excel = CharLengthAudit_&datetime_stamp..xlsx;
  
```

GENERATE EXCEL OUTPUT



```

%let filename_excel = CharLengthAudit_&datetime_stamp..xlsx;

ods excel file="&outpath_clean.\&filename_excel"

options(sheet_interval='none' embedded_titles='yes');

%if &nflagged > 0 %then %do;
ods excel options(sheet_name="Flagged Values");

proc print data=&outlib..&outds noobs; run;

ods excel options(sheet_name="Dataset Counts");
proc print data=flagged_count noobs; run;
ods excel options(sheet_name="Summary");
proc print data=audit_summary noobs; run;
%end; %else %do;
ods excel options(sheet_name="Summary");

data no_flags;
  length Message $200;
  Message="No character values exceeded threshold (&threshold).";
run;

proc print data=no_flags noobs; run;
%end;

ods excel close;
%put NOTE: Excel written ? &outpath_clean.\&filename_excel;%end;

%put NOTE: Completed audit_all_char_length macro.;

%mend;
%audit_all_char_length(lib=mylib,outpath="C:\Users\kondar\Desktop\presentation",
  threshold=200,excelout=YES,outlib=work,outds=all_flagged_vars);
  
```

CHARACTER LENGTH AUDIT OF ALL DATASETS REPORT

For Outputs: Variables Exceeding Threshold Detected

Character Variables > 200 Characters in mylib			
Dataset	Variable	Length	Variable Label
mylib.AE	AETERM	288	Reported Term for the Adverse Event
mylib.CM	CMTRT	238	Reported Name of Drug, Med, or Therapy
mylib.DV	DVTERM	228	Protocol Deviation Term
Character Variables > 200 Characters in mylib			
Dataset		Flagged Variables Count	
mylib.AE		1	
mylib.CM		1	
mylib.DV		1	
Character Variables > 200 Characters in mylib			
Library	Datasets Processed		Flagged Variables
mylib	32		3

CHARACTER LENGTH AUDIT OF ALL DATASETS REPORT

For output: All Variables Within Threshold

Character Variables > 200 Characters in mylib

Message

No Character variables > 200 characters found in any dataset in mylib

KEY RESULTS, REUSABILITY AND REGULATORY ALIGNMENT

Results

- 100% reduction in manual review time.
- Detected all long variables (>200 chars).
- Generated Excel summaries automatically.
- Eliminated human error in QC checks.

Reusability

- Designed for cross-study deployment.
- Parameter-driven for different thresholds.
- Compatible with SAS pipelines and batch processes.

Regulatory Alignment

- Ensures CDISC-compliant SUPPVAR creation.
- Strengthens data transparency for audits.
- Enhances traceability for FDA/EMA submissions.

COMPREHENSIVE LIST OF POTENTIAL MACRO ENHANCEMENTS

Input Parameter	Test #	Test	Expected Outcome
lib	1.01	Supply lib name for analysis	Analysis performed using the specified lib name
	1.02	No lib name is supplied	Relevant error message is written to the log, and the macro exits.
outpath	2.01	Supply outpath for analysis	Analysis performed using the specified outpath.
	2.02	No outpath is supplied	Relevant error message is written to the log, and the macro exits.
threshold	3.01	Numeric threshold	Analysis performed using the specified threshold.
	3.02	Non-Numeric threshold	Relevant error message is written to the log, and the macro exits.
excelout	4.01	EXCELOUT=YES specified	Report generated
	4.02	EXCELOUT=NO specified	Skipping report generation
	4.03	EXCELOUT=" " specified	Skipping report generation
outlib	5.01	OUTLIB = WORK specified	Datasets are stored in work library
	5.02	OUTLIB = " " specified	Relevant error message is written to the log, and the macro exits.
outds	6.01	OUTDS= dataset name	Analysis conducted on the given dataset.
	6.02	OUTDS= " " specified	Relevant error message is written to the log, and the macro exits.

PERFORMANCE IMPACT

Parameter	Manual	Automated
Review Time	1 hour	< 1 Minute
Errors	Possible	0% (Rule-Based)
Compliance	Manual Logs	Auto Reports
Reusability	Low	High
Scalability	Limited to few datasets	Unlimited (32+ datasets)
Accuracy	Depends on human review	Consistent, rule-based checks
Documentation	Manual notes	Auto Excel summaries
Maintenance Effort	High (code edits required)	Minimal (parameter-based)
Traceability	Hard to track manually	Timestamped Excel/SAS Logs
Regulatory Readiness	Variable quality	Fully standardized & auditable

EXCEPTIONAL HANDLINGS



Check
Parameters



Verify Output
Path



Confirm Library
Assigned



Validate Positive
Threshold



Skip if no
Datasets



Warn Length
Limit Exceeded



Report Even if
Empty



Log Steps
Clearly

VALIDATION SUMMARY

Validation Step	Description	Result
Unit Testing	Checked with mock datasets containing char vars	✓ Passed
Manual Review	Compared to PROC CONTENTS output manually	✓ Consistent
Real Case Study	Detected variables >200 chars	✓ Found in SUPP dataset
Excel Output Validation	Confirmed correct formatting	✓ Works
Empty Dataset Handling	Confirmed macro exits with no data	✓ No error
Corrupted Variable Detection	Tested response to corrupted values	✓ Needs improvements
Threshold Check	Threshold validation	✓ Stable
Performance Test	Executed on 50+ dataset to measure runtime	✓ Efficient
Cross-Library Validation	Validated results across multiple SAS Libraries	✓ Consistent
Error Handling Test	Simulated missing parameters & Libraries	✓ Proper error messages
User Acceptance Test	Evaluated by programmers and reviewers	✓ Approved

CHALLENGES



Varied Study
metadata
formats



High memory
usage on big
trials



Complex
variable labels
cause errors



Cannot detect
corrupted
Variables



SAS-only
environment
restriction



Not user-
friendly for non-
coders



Slow with large
SAS libraries



SUPPQUAL
creation not
automated

FUTURE DIRECTIONS



Standardize
metadata
formats globally



AI-based
corruption
detection



Enable
distributed
cloud processing



Optimize
memory with
streaming



Support multi-
language
integration



Auto-generate
SUPPQUAL
datasets



Intelligent label
cleaning



Build no-code
interface

CONCLUSION

-  **Audit Success:** Identified all over-length variables.
-  **Efficiency:** Automated tools cut manual review time.
-  **Integrity:** Validated metadata ensures full traceability.
-  **Compliance:** Fully aligned with CDISC/FDA standards.
-  **Robustness:** Validated parameters prevent system errors.
-  **Reporting:** Automated Excel outputs for stakeholders.
-  **Stability:** Optimized SQL reduces system resource load.
-  **Scalability:** Reusable macro for any study environment.

POWERING POSSIBILITIES IN DRUG DEVELOPMENT

Navitas is a knowledge-driven, tech-enabled CRO offering bespoke solutions across the drug development lifecycle. From strategic advisory services that navigate complex regulatory landscapes to clinical development ensuring compliant, data-driven operations, and post-marketing solutions enhancing product value with safety surveillance, lifecycle management, and real-world evidence—we integrate deep expertise, cutting-edge technology, and industry insights to drive success.





Driving better outcomes through insights



[Website](#)



[LinkedIn](#)



Ram Mohan Konda

[rammohan.konda@navitaslifesciences.co
m](mailto:rammohan.konda@navitaslifesciences.com)