

A Macro Tool to Find and/or Split Variable Text String Greater Than 200 Characters for Regulatory Submission Datasets.

Venkata N Madhira

Senior SAS Programmer, Shionogi Inc.

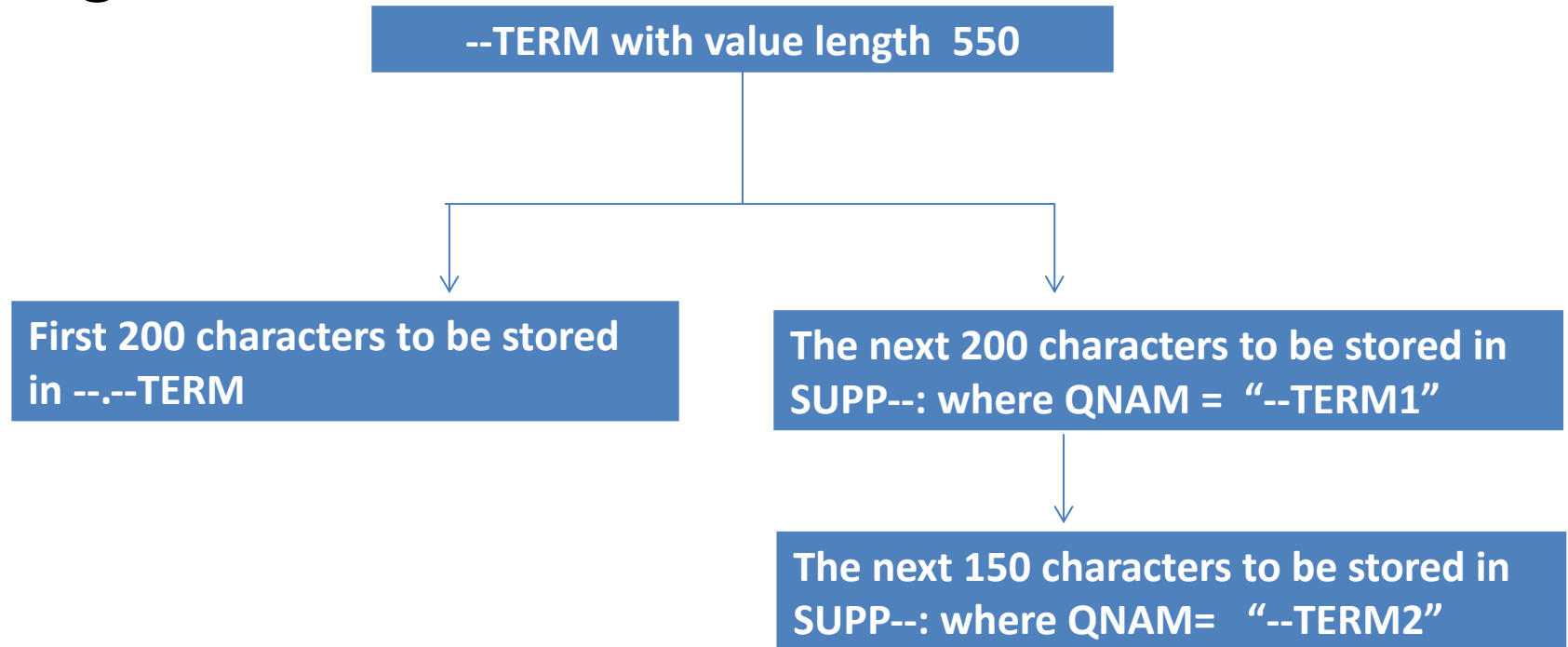
PhUSE US Connect 2018

Introduction

- For the studies started after 17DEC2016, it is the FDA requirement that the clinical trials data should be submitted electronically and following CDISC guidelines.
- All submission datasets must comply with CDISC guidelines (Version 5 SAS transport file format).
- V5 SAS transport file is compatible for the variable with value length less than or equal to 200 characters.
- One of the challenging tasks in following CDISC guidelines is, ensuring the variable text string in a submission dataset should not exceed 200 characters.

Splitting Variable Using CDISC Guidelines

For general observation class domains:



-- is Domain Name (Eg: MH,DV..)

Splitting Variable Using CDISC Guidelines

- For Special-Purpose Domain (CO), Trial Design Domain (TS):

--VAL (Parameter Value) with value length 550



The first 200 characters to be stored in parent domain variable (--VAL)



The next 200 characters to be stored in a new variable (--VAL1), in the parent domain



The remaining 150 characters to be stored in a new variable (--VAL2) in the parent domain

-- is Domain Name
(Eg: CO,TS)

How to Split the Long Text ?

- For a programmer, it is tedious and cumbersome process to search each dataset in a library for the variables with text value greater than 200 characters.
- If so, it's a challenging task to split them into additional variables without breaking the word, in a readable manner and complying CDISC standards.

Macro Tool FINDSPLIT

- To get this complicated task done swiftly, created a macro tool FINDSPLIT.
- **The Advantages of this Macro Tool are:**
 - (i) It is very easy to use (just a couple of parameters to be passed).
 - (ii) Finds the variables in datasets with text string greater than 200 characters in a given specific library irrespective of number of datasets.
 - (iii) Provides the summary in html window.

Macro Tool FINDSPLIT

(iv) Splitting occurs for the variables, if the text string is greater than 200 characters in any dataset within the specified library.

(v) Splitting occurs in a readable manner and comply with CDISC standards for the submission purpose.

(vi) It saves a lot of programmer's precious time and expedites the submission activities.

Mechanism of FINDSPLIT Macro

- **%findsplit** (libnme=, split=)

Keyword Parameter	Parameter Value	Description
libnme	The library name in which your datasets are stored (eg: WORK, Raw...)	Eg: %findsplit (libnme=work, split=)
split	When keyword parameter value is N (Possible values are Y, N only)	Eg: %findsplit (libnme=work, split=N) Variable value length information in each dataset with more than 200 characters within the specified library will be given in html window.

Mechanism of FINDSPLIT Macro

Keyword Parameter	Parameter Value	Description
split	When keyword parameter value is Y (Possible values are Y, N only)	%findsplit (libnme=work, split=Y) (a) Variable value length information in each dataset with more than 200 characters within specified library will be given in html window.

Mechanism of FINDSPLIT Macro

Keyword Parameter	Parameter Value	Description
split	When keyword parameter value is Y (Possible values are Y, N only)	(b) Stores the variable long text value into additional variables in a readable way with each new variable length is less than or equal to 200 characters. The scope of this process is all datasets within a specified library .

FINDSPLIT Macro Tool Output

- **%findsplit** (libnme=work, split=N)

Below is the Variable Information With Length Greater than 200 in work Library Datasets

Library Name	Dataset Name	Variable Name	Variable Maximum Length
work	_1	avalca	335
work	_1	avalcb	734
work	_2	avalcc	296
work	_2	avalcd	356

FINDSPLIT Macro Tool Output

%findsplit (libname=work, split=Y)
_1 Dataset Before Macro Execution:

avalca	(335)	avalcb	(734)
		calculation of additional study days within subdivisions of time in a clinical trial may be based on one or more sponsor-defined reference dates not represented by RFSTDTC. In such cases, the Sponsor may define SupplementalQualifier variables and the define.xml should reflect the reference dates used to calculate such study days. If the sponsor wishes to define day within element or day within epoch, the reference date/time will be an elementstart date/time in the Subject Elements dataset represent the timing of an observation relative to the sponsor V3.1 was the first fully implementation-ready version of the CDISC Submission Data Standards that was directlyreferenced by the FDA for use in human clinical studies inv	
all unplanned visits, but this method provides no differentiation between the unplanned visits and does not provide chronological sorting. Methods that provide a one-to-one relationshipbetween visits and values of VISITNUM, that are consistent across domains, and that assign VISITNUM values that sort chronologically require more work			

_1 Dataset After Macro Execution:

avalca	avalca1	avalcb	avalcb1	avalcb2	avalcb3
		calculation of additional study days within subdivisions of time in a clinical trial may be based on one or more sponsor-defined reference dates not represented by RFSTDTC. In such cases, the Sponsor	may define SupplementalQualifier variables and the define.xml should reflect the reference dates used to calculate such study days. If the sponsor wishes to define day within element or day within	epoch, the reference date/time will be an elementstart date/time in the Subject Elements dataset represent the timing of an observation relative to the sponsor V3.1 was the first fully	implementation-ready version of the CDISC Submission Data Standards that was directlyreferenced by the FDA for use in human clinical studies inv
all unplanned visits, but this method provides no differentiation between the unplanned visits and does not provide chronological sorting. Methods that provide a one-to-one relationshipbetween visits	and values of VISITNUM, that are consistent across domains, and that assign VISITNUM values that sort chronologically require more work				

Tips for Macro Execution

- Make sure that the dataset names in your library should not be same as given below.

_TEMPCONT1
_TEMPLN1
_TEMPLN2
_TEMPLN_DSN
SASHELPVCOLUMN
_TEMPORARY_VAR_LEN_DSN
_TEMPORARY_VAR_NOLEN_DSN
_TEMPORARY_INDSN_CONTENTS

- If so macro stops execution, and will be notified in log as a WARNING message.

Tips for Macro Execution

- Make sure both keyword parameters should have values for the macro execution.
- Keyword parameter SPLIT should have a value of either Y or N.
- **The macro stops execution if:**
 - Any of the keyword parameters value is missing and the reason will be notified in log as a WARNING message.
 - Keyword parameter SPLIT has a value of Y and the preexistence of following variables : X, LENGTHA in your dataset.
 - and the reason will be notified in log as a WARNING message.

Snippet of the Macro To Split Variable

- data &indsn;
- set &indsn;
- lengtha=lengthn(&aval);
- array varx (*) \$200 &aval.1 - &aval&var1;
- do i=1 to dim(varx);
- if i = 1 then do ;varx(i)= substr(&aval, 1, ifn(index(reverse(substr(&aval, 1, 200))), " ") ne 0 and index(reverse(substr(&aval, 1, 200))), " .") ne 0 , 7
- (200- min(index(reverse(substr(&aval,1,200))), " "), index(reverse(substr(&aval,1,200))), " .")) ,
- (200- max(index(reverse(substr(&aval,1,200))), " "), index(reverse(substr(&aval,1,200))), " ."))));
- x=sum(x, length(varx(i)));
- end;
- if i gt 1 and x< lengtha then do;
- varx(i)= substr(&aval, x+1 , ifn(index(reverse(substr(&aval, x+1,min(lengtha-x-1, 200))), " ") ne 0 and index(reverse(substr(&aval, x+1, min(lengtha-x-1, 200))), " .") ne 0 ,
- (min(lengtha-x-1, 200) - min(index(reverse(substr(&aval,x+1,min(lengtha-x-1, 200))), " "), index(reverse(substr(&aval,x+1,min(lengtha-x-1, 200))), " .")) ,
- (min(lengtha-x, 200)- max(index(reverse(substr(&aval,x+1,min(lengtha-x-1, 200))), " "), index(reverse(substr(&aval,x+1,min(lengtha-x-1, 200))), " ."))));
- x=sum(x, length(varx(i)));
- if x eq lengtha then x=x+1;
- if varx(i) = "###" then call missing(varx(i));
- end;
- end;
- drop i x ;
- run;

Summary of FINDSPLIT Macro

- Using the macro FINDSPLIT, variable with text string greater than 200 characters can be identified and/or split into additional variables in a readable way complying CDISC standards.
- The scope of this macro is for all datasets in a specified library.
- It is very easy to use and saves a lot of programmer's precious time.

Acknowledgement

- I would like to thank Malla Reddy Boda, Associate Director, Shionogi Inc, for his review and invaluable comments.

Thank You

Contact Information

- Venkata N Madhira
- E-mail: sharma.madhira@gmail.com
- Harish Yeluguri
- E-mail: harish.yeluguri@gmail.com

References

- CDISC SDTMIG V3.2
- <https://www.pharmasug.org/proceedings/2015/SS/PharmaSUG-2015-SS06.pdf>
- <https://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM511237.pdf>
- SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.
- Other brand and product names are trademarks of their respective companies.

Questions?