

PhUSE US Connect 2019
Paper CT16
A Macro Tool to Reduce Dataset Size for the Regulatory Submissions

Venkata N Madhira, Shionogi Inc, Florham Park, USA
Harish Yeluguri, Shionogi Inc, Florham Park, USA
Prabhakara Rao Burma, Acerta Pharma, CA, USA

Abstract

As per the study data technical conformance guide the clinical trials data must be submitted in SAS® V5 transport file format. These transport files have size limitations such as each individual submission dataset should not exceed 5 GB, and it is recommended that the each character variable value length should be as same as the maximum value length of corresponding variable across all domains within the study except variables in suppqual datasets. For suppqual datasets, each variable value length should be set to its maximum value length. In order to accomplish these complicated tasks, we created a macro tool called ALTVARLEN. Using this macro tool we can expedite the submission process by reducing the dataset sizes following study data technical conformance guide (March 2018).

Introduction

It is a requirement by the FDA that the clinical trials data must be submitted in SAS® V5 transport file format, and the size of transport file should not exceed 5 GB. In fact, there are two factors which are directly proportional to dataset size. (i) Assigning superfluous default length to character variables while writing the dataset specification. (ii) More number of observations and/or rows in a dataset. The primary approach is to reduce dataset size is remove unnecessary extra length to character variables. We have some guidelines from study data technical conformance guide to reset the character variable length, such as (i) each character variable value length should be as same as the maximum value length of corresponding variable across all domains within the study, except variables in suppqual datasets (ii) For suppqual datasets, each variable value length should be set to its maximum value length. In order to reset the character variable length following study data technical conformance guide, we created a macro called ALTVARLEN.

Advantages of the Macro Tool ALTVARLEN

- It is very easy to use (just a couple of parameters to be passed).
- This macro follows the study data technical conformance guide.
- This macro resets the character variable length to its maximum value length across the domains within specified library except for suppqual datasets.
- For suppqual dataset variables, resets the variable length to its maximum value length.
- Provides each dataset character variable length summary before and after execution of ALTVARLEN macro in html window where the character variable length had been reassigned.
- It saves a lot of programmer's precious time and expedites the submission activities.

Table1: Keyword parameters description

Keyword Parameter	Parameter Value	Description
lib	The library name in which your datasets are stored (eg: WORK, Raw...)	%altvarlen (acrossdsn=, lib=WORK);
acrossdsn	When the value is N (Possible values are Y, N only)	%altvarlen (acrossdsn=N, lib=); • For suppqual dataset variables, resets the variable length to its maximum value length.

acrossdsn	When the value is Y (Possible values are Y, N only)	%altvarlen (acrossdsn=Y, lib=); This macro resets the character variable length to its maximum value length across the domains within specified library except for suppqual datasets.
-----------	--	---

ALTVARLEN Macro Code

```
%macro altvarlen (acrossdsn=, lib=);
```

```
%macro varlenwdsn ;
```

```
%if %sysfunc(exist(__tempcont__chk)) eq 1 or %sysfunc(exist(__max_count)) eq 1 or %sysfunc(exist(__tp)) eq 1
```

```
or %sysfunc(exist(_____dsn1)) eq 1 or %sysfunc(exist(_____newlength)) eq 1 or %sysfunc(exist(_____new_old_dsn)) eq 1
```

```
or %sysfunc(exist(_____dsnlength)) eq 1 %then %do;
%put WARNING: Datasets name should not be identical with the following
```

```
names: __tempcont__chk, __max_count, __tp, _____dsn1
_____newlength, _____new_old_dsn, _____dsnlength;
```

```
%abort cancel;
```

```
%end;
```

```
proc sql noprint;
```

```
create table _____dsn1 as
```

```
select memname
```

```
from dictionary.tables
```

```
where compress(upcase(libname)) eq "&lib." and index(compress(upcase(memname)), "SUPP") >
```

```
0;
```

```
select count (distinct memname) into : dsncount
```

```
from _____dsn1;
```

```
quit;
```

```
data _____dsn1;
```

```
set _____dsn1;
```

```
cnt+1;
```

```
call symput ("dsn" || strip(put(cnt, best.)), strip(memname));
```

```
run;
```

```
proc datasets lib=work memtype=data nolist;
```

```
delete _____dsn1;
```

```
quit;
```

```
%do i= 1 %to &dsncount;
```

```
proc contents data=&&lib..&dsn&i out=__tempcont (keep=name length type where=(TYPE eq 2))
```

```
noprint;
```

```
run;
```

```
proc sql noprint;
```

```
select 'max (length(' || compress(name) || ')) as ' || compress(name) into: var separated by ' , '
```

```

        from ____tempcont;
quit;

proc sql noprint;
    create table ____max_count as
    select &var
    from &&lib..&&dsn&i;
quit;

proc transpose data=____max_count out=____tp;
run;

proc sql noprint;
select compress(_name_)||'|' character('||strip(put(col1,best.))||'|') into: variable separated by ', ' from ____tp;
    alter table &&lib..&&dsn&i modify &variable;
quit;

proc contents data=&&lib..&&dsn&i out=_____newlength (where =(type eq 2) keep=name LENGTH type)
noprint;
run;

proc sql noprint;
    create table _____new_old_dsn (where=( old_length ne new_length)) as
    select "&&dsn&i" as Domain length =20 , a.name, a.length as old_length label="Variable Length BEFORE
Execution of Macro",
    b.LENGTH as new_length label="Variable Length AFTER Execution of Macro"
    from ____tempcont as a right join _____newlength as b
    on compress(upcase(a.name)) eq compress(upcase(b.name)) ;
quit;

proc append base=_____dsnlength data=_____new_old_dsn;
run;

proc datasets lib=work memtype=data nolist;
    delete ____tempcont_chk ____max_count ____tp _____newlength ;
quit;

%end;

ods listing close;
ods html;
    title1 "Variable Length in &lib Library SUPPQUAL data sets Before and After execution of
ALTVARLEN macro";
    proc print data=_____dsnlength (where =(domain ne " _____APP1")) split="~"
NOOBS;

        run;
    ods html close;
ods listing;

proc datasets lib=work memtype=data nolist;
    delete _____dsnlength ____tempcont _____new_old_dsn;

```

```

quit;

%mend varlenwdsn;

%macro varlenacrdsn ;

%if %sysfunc(exist(__tempcont__chk)) eq 1 or %sysfunc(exist(__max_count)) eq 1 or %sysfunc(exist(__tp)) eq
1
    or %sysfunc(exist(__dsn1)) eq 1 or %sysfunc(exist(__var1)) eq 1 or
%sysfunc(exist(__1)) eq 1
    or %sysfunc(exist(__appdsn)) eq 1 or %sysfunc(exist(__app1)) eq 1 or
%sysfunc(exist(__tp1)) eq 1 %then %do;
    %put WARNING: Datasets name should not be identical with the following
names: __tempcont__chk, __max_count, __tp, __dsn1
    __var1, __1, __appdsn, __app1, __tp1;
    %abort cancel;
%end;

proc sql noprint;
    create table __var1 as
    select count (name) as varcount, name, memname
    from dictionary.columns
    where compress(upcase(libname)) eq "&lib." and type eq "char" and
index(compress(upcase(memname)), "SUPP") eq 0
    group by name
    having varcount > 1
    order by name;
quit;

data __var1;
    set __var1;
    by name;
    cnt1+1;
    call symput ("dsncount", strip(put(cnt1, best.)));
    call symput ("dsnord" || strip(put(cnt1, best.)), strip(memname));
    call symput ("varord" || strip(put(cnt1, best.)), strip(name) );
run;

%do i =1 %to &dsncount;
    proc sql noprint;
        create table __1 as
        select max(length(&&varord&i)) as varmaxlen, "&&dsnord&i" as domain length = 15,
"&&varord&i" as variable length = 15
        from &&dsnord&i;
    quit;

    proc append base=__appdsn data=__1;
    run;
%end;

proc sql noprint;
    create table __app1 as

```

```

        select max(varmaxlen) as varmaxlen, variable
        from _____appdsn
        group by variable;
quit;

proc datasets lib=work memtype=data nolist;
    delete _____var1 _____1 _____appdsn ;
quit;

proc sql noprint;
    create table _____dsn1 as
    select memname
    from dictionary.tables
    where compress(upcase(libname)) eq "&lib." and index(compress(upcase(memname)), "SUPP")
eq 0;

        select count (distinct memname) into : dsncount
        from _____dsn1;
quit;

data _____dsn1;
    set _____dsn1;
    cnt+1;
    call symput ("dsn" || strip(put(cnt, best.)), strip(memname));
run;

        proc datasets lib=work memtype=data nolist;
            delete _____dsn1;
quit;

%do i= 1 %to &dsncount;

    proc contents data=&&lib..&&dsn&i out=___tempcont (keep=name length type where=(TYPE eq 2 ))
noprint;
    run;

    proc sql noprint;
    select 'max (length(' || compress(name) || ')) as ' || compress(name) into: var separated by ', '
    from ___tempcont;
quit;

proc sql noprint;
    create table ___max_count as
    select &var
    from &&lib..&&dsn&i;
quit;

proc transpose data=___max_count out=___tp1;
run;

proc sql noprint;
    create table ___tp as

```

```

select a.*, b.*
from ___tp1 as a left join _____app1 as b
on compress(upcase(a._NAME_)) eq compress(upcase(b.variable));
quit;

data ___tp;
set ___tp;
if compress(upcase(_NAME_)) eq compress(upcase(variable)) then COL1=varmaxlen;
run;

proc sql noprint;
select compress(_name_)||' character('||strip(put(col1,best.))||')' into: variable separated by ', ' from ___tp;
alter table &&lib..&&dsn&i modify &variable;
quit;

proc contents data=&&lib..&&dsn&i out=_____newlength (where =(type eq 2) keep=name LENGTH type)
noprint;
run;

proc sql noprint;
create table _____new_old_dsn (where=( old_length ne new_length) ) as
select "&&dsn&i" as Domain length =20 , a.name, a.length as old_length label="Variable Length BEFORE
Execution of Macro",
b.LENGTH as new_length label="Variable Length AFTER Execution of Macro"
from ___tempcont as a right join _____newlength as b
on compress(upcase(a.name)) eq compress(upcase(b.name)) ;
quit;

proc append base=_____dsnlength data=_____new_old_dsn;
run;

proc datasets lib=work memtype=data nolist;
delete ___tempcont_chk ___max_count ___tp ___tp1 _____newlength;
quit;

%if &i eq &dsncount %then %do;
proc datasets lib=work memtype=data nolist;
delete _____app1;
quit;
%end;

%end;

ods listing close;
ods html;
title1 "Variable Length in &lib Library Datasets (Not SUPPQUAL) Before and After execution of
ALTVARLEN macro";
proc print data=_____dsnlength (where =(domain ne " _____APP1")) split="~"
NOOBS;

run;

```

```

ods html close;
ods listing;

proc datasets lib=work memtype=data nolist;
    delete _____dsnlength ____ tempcont _____ new_old_dsn;
quit;

%mend varlenacrdsn;

%if &acrossdsn eq N %then %do;
%varlenwdsn;
%end;

%if &acrossdsn eq Y %then %do;

%varlenacrdsn;
%end;

%mend altvarlen ;

%*altvarlen (acrossdsn=N, lib=WORK);
%*altvarlen (acrossdsn=Y, lib=WORK);

```

CONCLUSION

Using the macro tool ALTVARLEN, we can reassign the character variable length following study data technical conformance guide, which leads to reduction in dataset size.

REFERENCES

Analysis Data Model Implementation Guide Version 1.1
SAS ®Certification Prep Guide Advanced Programming for SAS ®9
<https://www.fda.gov/downloads/forindustry/datastandards/studydatastandards/ucm384744.pdf>

ACKNOWLEDGMENTS

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

CONTACT INFORMATION

Venkata N Madhira
E-mail: venkatanmadhira@gmail.com
Sr. Statistical Programmer, Consultant at Shionogi Inc, NJ, USA

Harish Yeluguri
E-mail: harish.yeluguri@gmail.com
Sr. Statistical Programmer, Consultant at Shionogi Inc, NJ, USA

Prabhakara Rao Burma
E-mail: prsas85@gmail.com
Principal Clinical SAS Analyst, Acerta Pharma, CA, USA

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.