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.

Table 1: Keyword parameters description

<table>
<thead>
<tr>
<th>Keyword Parameter</th>
<th>Parameter Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lib</td>
<td>The library name in which your datasets are stored (eg: WORK, Raw...)</td>
<td>%altvarlen (acrossdsn=, lib=WORK);</td>
</tr>
<tr>
<td>acrossdsn</td>
<td>When the value is N (Possible values are Y, N only)</td>
<td>%altvarlen (acrossdsn=N, lib=);</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For suppqual dataset variables, resets the variable length to its maximum value length.</td>
</tr>
</tbody>
</table>
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;
  %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;
  %end;
  proc sql noprint;
    select 'max (length('| compress(name)| |')) as '| compress(name) into: var separated by ','
```sql
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="\n") 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(_______appdsn)) eq 1 or %sysfunc(exist(_________app1)) eq 1 or %sysfunc(exist(_____1)) eq 1
   or %sysfunc(exist(_________dsn1)) eq 1 or %sysfunc(exist(_________appdsn)) eq 1 or %sysfunc(exist(_________app1)) eq 1 or %sysfunc(exist(_______app1)) eq 1 or %sysfunc(exist(_________app1)) eq 1
   or %sysfunc(exist(_______app1)) 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
     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

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.