

A Macro to Avoid P21 FDAC036 Error Message for Regulatory Submission Datasets.

Venkata N Madhira, Shionogi Inc, NJ, USA
Prabhakara Rao Burma, Acerta Pharma, CA, USA

ABSTRACT

As per the CDISC guidelines the clinical trials data must be submitted in SAS® V5 Transport file format. These transport files should be validated using the Pinnacle 21 (P21) validator to ensure CDISC compliance and all errors given by the P21 validator should be addressed by the programming team. One of the most common P21 error messages is FDAC036 (Variable length is too long for actual data). Usually for most of the character variables we typically assign some default length while creating SDTM and ADaM datasets. But if the variable length is greater than its maximum value length, P21 validator will produce an error message (FDAC036). To address FDAC036 error message as well as significantly decrease the size of the file, we developed a macro named ADJLEN, which will be discussed in this paper.

INTRODUCTION

It is most common to assign some default length to a character variable while generating both SDTM and ADAM dataset. The default variable length for a character variable may be greater than the actual maximum value length which causes significant increase in dataset size. In fact, it is the FDA requirement that all submission datasets must follow CDISC guidelines. As per the CDISC guideline all submission datasets must be in SAS V5 Transport file format and the file size should not exceed 5GB. These SAS V5 Transport files need to be validated using P21 validator for CDISC compliance. During this process P21 validator will produce FDAC036 (Variable length is too long for actual data) error message if character variable length is greater than its actual maximum value length. This error message can be addressed by assigning the actual maximum value length to the corresponding character variable.

It is tedious and cumbersome process for a programmer identifying the character variable whose length is not as same as its actual maximum value length since there have been tens of variables in each dataset. Another challenging task is to reassign the actual maximum value length to a character variable for which the actual length is not equal to the maximum value length in each submission dataset. This complicated task can be accomplished by using the macro called ADJLEN as given in this paper.

MECHANISM OF ADJLEN MACRO

Keyword Parameter	Parameter Value	Description
DSN	Input dataset Eg: DM, LB,	%adjlen(dsn=dm,exc=)
EXC	Exclude the list of variables for which the lengths should not be changed. eg: %str("DOMAIN", "STUDYID", "USUBJID")	%adjlen(dsn=dm,exc=%str("STUDYID"))

Table 1.1 Keyword parameters descriptions for the macro ADJLEN.

STEP - I: Creation of CARS dataset for ADJLEN macro illustration.

```
data _1;
  length make $30 model $80 type origin $60;
  make="";
  model="";
  type="";
  origin="";
run;
```

```
data cars;
  set sashelp.cars;
  keep make model type origin DriveTrain;
run;
```

```
data cars;
set _1 cars;
run;
```

STEP - II: Original length of each character variable value in the dataset CARS except DriveTrain (I would just like to keep the same length for this variable as it was in the dataset.)

NAME	TYPE	LENGTH
make	2	30
model	2	80
origin	2	60
type	2	60

STEP - III: Execute adjlen macro.

```
%macro adjlen (dsn=,exc=);
%if %sysfunc(exist(__tempcont chk)) eq 1 or
%sysfunc(exist(__max_count)) eq 1 or
%sysfunc(exist(__tp)) eq 1 %then %do;
%put WARNING: Datasets name should not be identical with the
following names:_____tempcont chk,_____max_count,____tp ;
%abort cancel;
%end;

proc contents data=&dsn out=__tempcont (keep=name length type where=(TYPE eq 2 and
NAME not in (&exc))) 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 &dsn; quit;

proc transpose data=__max_count out=__tp;
run;

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

proc datasets lib=work memtype=data nolist;
delete__tempcont chk__max_count__tp ;
quit;
%mend;

%adjlen(dsn=cars,exc=%str("DriveTrain"));
```

STEP - IV: After execution of ADJLEN macro on cars dataset, the length of each character variable has been reassigned with its maximum value length, where it is applicable as given below.

NAME	TYPE	LENGTH
make	2	13
model	2	40
origin	2	6
type	2	6

STEP - V: Comparison of character variable lengths in CARS dataset Before (Left Image) and After (Right Image) execution of ADJLEN macro.

NAME	TYPE	LENGTH	NAME	TYPE	LENGTH
make	2	30	make	2	13
model	2	80	model	2	40
origin	2	60	origin	2	6
type	2	60	type	2	6

CONCLUSION:

Using ADJLEN macro, the maximum character variable length can be reassigned to the corresponding variables to avoid yielding the P21 FDAC036 error message. This macro will also significantly decrease the file sizes for submissions.

REFERENCES

CDISC SDTMIG V3.2

<https://www.pharmasug.org/proceedings/2015/SS/PharmaSUG-2015-SS08-SAS.pdf>

<https://www.lexjansen.com/pharmasug/2017/SS/PharmaSUG-2017-SS05.pdf>

<https://www.pinnacle21.com/validation-rules/sdtm>

<https://www.fda.gov/downloads/forindustry/datastandards/studydatastandards/ucm384744.pdf>

CONTACT INFORMATION

Venkata N Madhira

E-mail: venkatanmadhira@gmail.com

Sr. Statistical Programmer,

Consultant at Shinogi Inc, NJ, USA

Prabhakara Rao Burma

E-mail: prsas85@gmail.com

Lead SAS Programmer,

Consultant at 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.