Paper 099-2010

Using Proc Contents Output to Perform Quality Control Checks on SDTM Datasets

Jennifer Srivastava, Quintiles, Morrisville, North Carolina

ABSTRACT

This paper focuses on using SAS Proc Contents output as a tool for performing Quality Control (QC) checks on SDTM datasets. Although the code applies to SDTM data, it could easily be used for other applications as well.

The SAS code presented in this paper will be using the Variable Name (NAME), Variable Number or position within the dataset (VARNUM), and Variable Label (LABEL) fields from the Proc Contents output dataset, along with the Data Step and Proc Compare to perform two different types of QC checks on SDTM datasets.

INTRODUCTION

Proc Contents is often employed by programmers as a way to learn more about the details and attributes of a SAS® dataset. In addition to the primary use of Proc Contents, the SAS® programmer can also save the Proc Contents output as a dataset, and use it to perform a variety of Quality Control (QC) checks.

In the pharmaceutical industry, it is becoming increasingly common for submissions to the Food and Drug Administration (FDA) to have data presented in a standard format as defined by the Clinical Data Interchange Standards Consortium (CDISC). The analysis datasets submitted to the Food and Drug Administration (FDA) are typically created or revised to follow the Study Data Tabulation Model (SDTM). For the remainder of the paper, these analysis datasets will be referred to as SDTM datasets.

The code presented in this paper is used to perform QC checks on SDTM datasets. However this code could easily be applied to other types of data as well.

If a programmer uses Proc Contents to create an output dataset, he or she may consider using the NOPRINT option to avoid printing out the contents. The Proc Contents output dataset will contain the 40 variables listed below. As the reader can see, there is a lot of information available in this dataset. This information is often referred to as metadata or data about data. The SAS® code presented in this paper will be using the Variable Name (NAME), Variable Number or position within the dataset (VARNUM), and Variable Label (LABEL) fields.

Alphabetic List of Variables and Attributes from a Proc Contents Output Dataset

Variable	Type	Len	Format	Label
CHARSET	Char	8		Host Character Set
COLLATE	Char	8		Collating Sequence
COMPRESS	Char	8		Compression Routine
CRDATE	Num	8	DATETIME16.	Create Date
DELOBS	Num	8		Deleted Observations in Data Set
ENCRYPT	Char	8		Encryption Routine

Alphabetic List of Variables and Attributes from a Proc Contents Output Dataset (continued)

Variable	Type	Len	Format	Label
ENGINE	Char	8		Engine Name
FLAGS	Char	3		Update Flags (Protect Contribute Add)
FORMAT	Char	32		Variable Format
FORMATD	Num	8		Number of Format Decimals
FORMATL	Num	8		Format Length
GENMAX	Num	8		Maximum Number of Generations
GENNEXT	Num	8		Next Generation Number
GENNUM	Num	8		Generation Number
IDXCOUNT	Num	8		Number of Indexes for Data Set
IDXUSAGE	Char	9		Use of Variable in Indexes
INFORMAT	Char	32		Variable Informat
INFORMD	Num	8		Number of Informat Decimals
INFORML	Num	8		Informat Length
JUST	Num	8		Justification
LABEL	Char	256		Variable Label
LENGTH	Num	8		Variable Length
LIBNAME	Char	8		Library Name
MEMLABEL	Char	256		Data Set Label
MEMNAME	Char	32		Library Member Name
MEMTYPE	Char	8		Library Member Type
MODATE	Num	8	DATETIME16.	Last Modified Date
NAME	Char	32		Variable Name
NOBS	Num	8		Observations in Data Set
NODUPKEY	Char	3		Sort Option: No Duplicate Keys
NODUPREC	Char	3		Sort Option: No Duplicate Records
NPOS	Num	8		Position in Buffer
POINTOBS	Char	3		Point to Observations
PROTECT	Char	3		Password Protection (Read Write Alter)

Alphabetic List of Variables and Attributes from a Proc Contents Output Dataset (continued)

Variable	Type	Len	Format	Label
REUSE	Char	3		Reuse Space
SORTED	Num	8		Sorted and/or Validated
SORTEDBY	Num	8		Position of Variable in Sortedby Clause
TYPE	Num	8		Variable Type
TYPEMEM	Char	8		Special Data Set Type (From TYPE=)
VARNUM	Num	8		Variable Number

When creating SDTM datasets there is a requirement to place variables in a specific order as defined by the CDISC Specification Guide. One way to do this is to place a RETAIN statement after the DATA statement in the final data step of the program. Below is an example using an Adverse Event (AE) dataset.

```
DATA AE;

RETAIN STUDYID USUBJID AESEQ AESTDT AEENDT AESEV AESER AETERM PETERM SOCTERM;

SET AE;

RUN;
```

To verify that the variable order in the dataset matches the variable order in the CDISC Specification Guide, the programmer can refer to the Proc Contents output to see the variable order. Specifically, the programmer would want to look at the NAME and VARNUM variables.

If a dataset is small, it may be a relatively easy task to check the variable order manually. However, if a programmer is dealing with a large dataset containing many variables, or if he/she has to check many smaller datasets, the programmer may consider doing these checks programmatically. The following page displays the CHKORD macro, which contains some code to complete this task. This code assumes that there is a development dataset as well as a QC dataset. The QC programmer can add this code to his/her QC program to compare the variable order between the two datasets and to print out results with mismatches flagged.

```
*****************************************************************;
***       Check the Variable Order in a SDTM  Dataset   ***;
*****************************************************************;
%macro CHKORD(lbd=,lbq=,ds=);
%* lbd=development libname reference, lbq=qc libname reference, ds=dataset name ;


%* Contents of the Development Version of the Dataset;
proc contents data=&lbd..&ds out=cont_dev(keep=name varnum rename=(varnum=devnum)) noprint;
run;
```

```
proc sort data=cont_dev;
by name;
run;

%*Contents of the QC Version of the Dataset;
proc contents data=&lbq..qc_&ds out=cont_qc(keep=name varnum rename=(varnum=qcnum)) noprint;
run;

proc sort data=cont_qc;
 by name;
run;

%*Compare the Development and QC Versions and Flag any Mismatches;
data comp;
 merge cont_dev
            cont_qc;
 by name;
 if devnum ne qcnum then mismatch=1;
run;

proc print;
 title "Check Variable Positions for &ds";
where mismatch=1;
run;

%mend CHKORD;
%CHKORD(lbd=SDTM001,lbq=SDTMQ001,ds=AE)
```

If a programmer has a set of SDTMs that have already been programmed, passed QC review and passed a senior-level review, he/she may want to compare the variable names and labels from his/her current set of SDTMs to those in the final set of SDTMs. This would be another way to confirm that the variable names and labels conform to what is listed in the SDTM Specification Guide. The macro CHKPRV, which is below, contains code to perform this task.

```
*************************************************************************************************************************;
* Compare Variable Names and Labels in the Current Protocol's Dataset to those from the Previous Protocol *;
*************************************************************************************************************************;

%macro CHKPRV(lbc=,lbp=,ds=);
%* lbc=current development libname reference,
     lbp=previous development libname reference,
     ds=dataset name ;

%*Contents of the Development Dataset from the Current Protocol;
proc contents data=&lbc..&ds out=cont_cur(keep=name label);
run;

proc sort data=cont_cur;
 by name;
run;

%*Contents of the Development Dataset from the Previous Protocol;
proc contents data=&lbp..&ds out=cont_prev(keep=name label);
run;

proc sort data=cont_prev;
 by name;
run;
```

```
%*Compare the Variable Names and Labels from the Previous and Current Versions of the Development File;

proc compare base=cont_prev compare=cont_cur listall maxprint=(15 10000);

 id name;

 var label;

 title1 "&ds: Compare the Variable Names and Labels from Previous Protocol &lbp to those in Current Protocol &lbc";

 title2 "Base=Protocol &lbp , Compare=Protocol &lbc";

run;


%mend CHKPRV;

%CHKPRV(lbc=SDTM002,lbp=SDTM001,ds=AE)
```

CONCLUSION

In conclusion, the Proc Contents output dataset can be a useful programming tool for anyone who needs to perform QC checks on SDTM datasets or other types of data. These techniques can help the user save time and allow him/her to check data in a programmatic way as opposed to a manual way, which will increase the quality of the finished product.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Jennifer Srivastava
Enterprise: Quintiles
Address: 5927 S Miami Blvd
City, State ZIP: Morrisville, NC 27560
Work Phone: (919) 998-7035
Fax: (919) 998-2382
E-mail: Jennifer.srivastava@quintiles.com