

All the Data That Fits, We Print

Janet E. Stuelpner, ASG, Inc., Cary, North Carolina

Elizabeth Kaptanov, Pfizer, Inc., New York, New York

ABSTRACT

How many times do we want to print off a sample of data from a whole SAS (®) database? The first thing you do is to run a PROC CONTENTS and then figure out what the names of all of the data sets are before writing all of the PROC PRINTs to print off some data or even the whole catalog. It is tedious and time consuming. The other thing to consider is that typing in the name of each data set manually is prone to error. This paper will explore several ways in which to capture the names of the members of a SAS catalog and then using macro language, create a program to print off data samples.

INTRODUCTION

The first task is to determine which data is needed to be printed. It is easy to print off one data set at a time, especially if we know the name of the data set. Using a PROC PRINT or PROC REPORT, the printing becomes an easy task. However, what happens if we need to print a sample of data from many data sets. If they are all located in the same SAS catalog, it is easy and we don't need to know each individual data set name. We have many choices in which way we want to capture the names of the data sets. We will restrict ourselves to two very common ones. After that we will produce a macro in which the PROC PRINT is created for each data set in the catalog and then run. The only thing

we need to supply is the library of the data's location and how many observations need to be printed for the sample. The rest is done automatically.

GATHER NAMES

There are several ways in which to capture the names of the members of a SAS catalog. The first method to be explored is the PROC CONTENTS or PROC DATASETS. This will allow us to print off a list of members. This information can be captured in a SAS data set using the NOPRINT option and specifying an output data set. The only variables that are kept are the ones that are needed for this exercise. In this case, all we need is the data set name which is kept in the variable MEMNAME. It can also be helpful to keep the number of observations in each data set (NOBS). The drawback to this method is that you get one record for each occurrence of a variable in each data set. For example, if there are 5 variables, there will be five observations for that data set. There must be some additional processing done to arrive at a list of unique values. Below is an example of the code necessary to achieve the desired data set which contains the necessary information to proceed. First, a procedure to gather the member names and then a DATA step to subset only the first observation which contains the member name. (Note: The DATA step uses BY processing. It is not necessary

to sort the output from the PROC CONTENTS because it is automatically sorted by MEMNAME.)

```
LIBNAME MYLIB 'C:\MYDIR\MYDATA';
```

```
PROC CONTENTS  
  DATA=MYLIB._ALL_  
  NOPRINT  
  OUT=TEST  
  (KEEP=MEMNAME NOBS);  
RUN;
```

```
DATA TEST1;  
  SET TEST;  
  BY MEMNAME;  
  IF FIRST.MEMNAME;  
RUN;
```

Another way to get the same information is to use a PROC SQL and create a query of the dictionary tables. You can create a table (or data set) with all of the required information. The resultant data set will include one record for each member along with the number of observations. The PROC SQL also uses the default variable name of MEMNAME for the name of the SAS data set and NOBS for the number of observations in that data set. In either case, the result is a data set with the names of the members of the SAS database we need to sample.

```
LIBNAME MYLIB 'C:\MYDIR\MYDATA';
```

```
PROC SQL;  
  CREATE TEST1 AS  
  SELECT LIBNAME, MEMNAME, NOBS  
  FROM DICTIONARY.TABLES  
  WHERE LIBNAME='MYLIB';  
QUIT;
```

MACRO TO PRINT

The next step is to write a program that writes a program to create all of the PROC PRINTs needed to print the data. By using various options, it is possible to print out a subset of data or the whole data set. To accomplish this, an external file must be defined. A DATA step is used to create PUT statements that are then written out to an external file. Then this file is read back in and executed using a %INCLUDE. The purpose of this is to bring the code into the current SAS session and execute the code.

MACRO EXAMPLE

The macro defined below is very general. There are many ways in which to enhance the macro and allow for more specific options on the print out. There are many options that can be used to include titles, differences in pagination and spacing. The important thing to remember is that all variables will be printed and the number of observations in the original data set will be specified even if only a subset is output. The default for the macro is to print all observations (OBS=MAX). This can be changed in each invocation of the macro. The number of observations will always be reset to maximum. This will not compromise an interactive session.

```
%macro printit(path=,outfile=,prtobs=MAX);
```

```
*****
```

```
Part I: to gather member names from database
```

```
*****,
```

```
*Define database and output file;
```

```
LIBNAME MYLIB "&path";
```

```
FILENAME OUTFILE "&outfile";
```

OPTIONS NODATE;

*Get member names and number of obs;

PROC SQL;

```
CREATE TABLE TEST1 AS
SELECT LIBNAME, MEMNAME, NOBS
FROM DICTIONARY.TABLES
WHERE LIBNAME='MYLIB';
```

QUIT;

*Print member names and number of obs;

```
PROC PRINT DATA=TEST1;
TITLE1 "DATA SETS FOR &path";
RUN;
```

Part II: Print data from each data set

*****;

```
%macro prt(memname,nobs);
OPTIONS PAGENO=1 PS=60 OBS=&prtobs;
PROC PRINT DATA=MYLIB.&memname;
TITLE1 "DATA SET=&memname NOBS=&nobs";
FOOTNOTE "RUN DATE: &sysdate";
RUN;
%mend prt;
```

```
%macro prtds;
OPTIONS OBS=MAX;
DATA _NULL_;
SET TEST1;
FILE "&outfile";
PUT '%prt('MEMNAME $8.', 'NOBS 8.>');
RUN;
```

```
%INCLUDE "&outfile";
```

```
%mend prtds;
```

```
%prtds;
```

*To reset OBS parameter;

```
OPTIONS OBS=MAX;
```

```
%mend printit;
```

CONCLUSION

Through a simple series of SAS procedures and DATA steps, a program is created, written out to an external file, read in and then executed. Many special features can be incorporated. A print of the member names can be included in the macro. Titles can be very explicit or very simple depending on the needs of the user. A great deal can be accomplished with a basic knowledge.

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Janet Stuelpner
ASG, Inc.
326 Old Norwalk Road
New Canaan, CT 06840

(203) 966-7520 (voice)
(203) 966-8027 (fax)
jstuelpner@worldnet.att.net

Elizabeth Kaptsov
Pfizer, Inc.
235 East 42nd Street
New York, NY 10017

(212) 573-1924 (voice)
(212) 573-2612 (fax)
kaptse@pfizer.com