Paper 80-27

A New Method to Estimate the Size of a SAS® Data Set

Xingshu Zhu and Shuping Zhang,
Merck Research Laboratories, Merck & Co., Inc., Blue Bell, PA 19422

ABSTRACT

According to the "Guidance for Industry - Providing Regulatory Submissions in Electronic Format - NDAs" (January 1999), the submitted analysis dataset (SAS transport file) must not exceed 25 MB in size and must not contain more than 62999 records. These requirements make it essential to have a convenient and reliable method to estimate the size of a SAS dataset. There are existing formulas available to be used for estimating the size of a SAS dataset, however, these formulas are all dependent on the operating system. Thus, the codes developed using such formulas are also system dependent, making them inconvenient and unreliable under certain circumstances. We have developed an alternative set of SAS codes to determine the size of a data set. These codes can perform the desired task effectively, and more importantly, they do not depend on the operating system.

KEYWORDS

dataset size, proc contents, NDA submission.

INTRODUCTION

In many circumstances, it is required to know the exact size of the dataset that is handled by a program. For example, in the NDA submission, the maximum size of the dataset is restricted to 25 MB. If a data set exceeds this limit, it needs to be divided into smaller subsets according to specific regulations. Therefore, the correct determination of the size of a data set becomes an indispensable step in the NDA submission process.

There are some formulas for calculating the amount of disk or memory space occupied by a data set. However, these formulas are intended for particular operating systems. For example,

for AOS/VS and VMS,

the following formula is used to calculate the size of the data set:

SIZE=CEIL((260 + VAR*136+NOBS*LRECL)/512),

whereas for CMS and VM/PC,

the formula is SIZE=CEIL((98+372+HISTLEN+(NVAR *82)+ (NOBS*(LRECL+4)))/BLKSIZE), where

NVAR is the number of variables,

NOBS is the number of observations,

LRECL is the length of the data record, and

HISTLEN is the length of the history data stored with the data set. Although NVAR and NOBS can be obtained from the output of the PROC CONTENTS of a dataset, the variable HISTLEN cannot be determined in a straightforward method. In addition, since a different formula has to be used for each operating system to calculate the size of a SAS data set, it is inconvenient and prone to error when a program has to be utilized under different operating systems. It is thus desirable to develop methods that are independent of the operation system. In this paper, we describe our approach to develop such a method.

METHODS

The procedure PROC CONTENTS is normally used to extract information intrinsic to a data set, such as the names of the variable, the type and length of the data, the number of variables and the number of records, etc. The procedure PROC CONTENTS also reports certain engine-specific and operating-system-specific details, such as NUMBER OF DATA PAGE and DATA SET PAGE SIZE, etc. If one can manage to obtain the relevant information provided by PROC CONTENTS, then it will be possible to calculate the size of a SAS data set from information intrinsic to the SAS system, therefore independent of the engine and the operating system. We have thoroughly investigated the output of PROC CONTENTS and developed a method to use this procedure for the determination of the size of any type of SAS datasets in different operating systems.

The following codes were developed during the NDA submission process in order to satisfy the requirements of the submission protocols from FDA, which dictate that the SAS datasets must not exceed 25 MB in size and not to contain more than 62999 records.

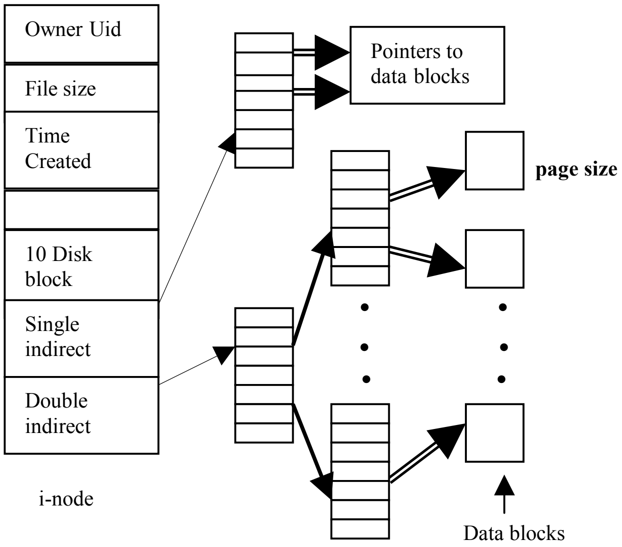
SAS CODES

```
%macro DsSize (
 dsin =    /* dsin is a one- or two-level input SAS dataset. */  );

%* --------------------------------------------------------------------------------*;
%* Output of this macro contains two global macro variables:  *;
%* kb  – the size of the data set &dsin in kb.                      *;
%* mb – the size of the data set &dsin in mb.                       *;
%*---------------------------------------------------------------------------------*;
%global kb mb;



%* --------------------------------------------------------------------------------*;
%* Use PROC CONTENTS to obtain data set information.       *;
%* Save the output into a flat file named C:\_dssize_.lst          *;
%* --------------------------------------------------------------------------------*;
proc printto print="C:\_dssize_.lst" new;

proc contents data=&dsin;

proc printto;

run;


%* --------------------------------------------------------------------------------*;
%* Extract engine/operating-system specific information.         *;
%* --------------------------------------------------------------------------------*;
data _dssize_(keep = pagesize totpages);

      length string $200;

      infile "C:\_dssize_.lst" length=lg;

      retain pagesize totpages;

      input @1 string $varying. lg;

      if index(upcase(string),"DATA SET PAGE SIZE")

      then pagesize= nput(compress(scan(string,2,':')),8.);

      if index(upcase(string),"NUMBER OF DATA SET PAGE")

      then totpages= input(compress(scan(string,2,':')),8.);

      if index(upcase(string),"OBS IN FIRST DATA PAGE")

      then output;

run;
```

```
%* --------------------------------------------------------------------------------*;
%* Calculate size of the input data set &dsin in KB and MB.      *;
%* Output generated macro variables kb and mb.                    *;
%*---------------------------------------------------------------------------------*;
data _null_;
       set _dssize_;
       size_bt=256+pagesize*totpages;
       kb=ceil(size_bt/1024);
       mb=round(size_bt/1048576,0.1);
       call symput("kb",kb);
       call symput("mb",mb);
run;


%*---------------------------------------------------------------------------------*;
%* Clean data set dssize and flat file C:\_dssize_.lst            *;
%*---------------------------------------------------------------------------------*;
proc datasets nolist; delete _dssize_; quit;
%local rc;
filename TARGET "C:\_dssize_.lst";
%let rc=%sysfunc(fdelete(TARGET));
%mend  DsSize;
```

NOTE

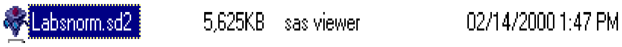
From the equation size_bt = 256 + pagesize*totpages, it can be seen that the size of a SAS data set is calculated through the variables *pagesize* and *totpages,* which are obtained from the output of the procedure PROC CONTENS and stored in the flat file C:_dssize_lst. The number 256 is added to the file size calculation due to the fashion with which data is stored in the computer. According to the theory of operating systems, associated with each file is a table called the ***I-node*** as shown below:



The file's I-node uses 256-byte disk space to store the information about the file, such as user ID, file type, file size, time of last modification, and more importantly, the location of the data blocks. Therefore, the size of a dataset is always 256 bytes larger than the disk space occupied by the data blocks. For a file up to 10 disk blocks (1 block=1 kB) in size, all the disk addresses of the data are kept in the I-node area. When a file is more than 10 disk blocks in size, a free disk block is acquired and the single indirect pointer is set to point to it. This block is used to contain disk block pointers, and the single indirect block can hold 256 disk addresses. For files over 256 blocks, the double indirect pointer is used to point to a disk block and so on. With double indirect block, this scheme is sufficient for files up to 64 MB.

EXAMPLE

As an example, we use an actual SAS data set "labsnorm.sd2" to test the macro DsSize in the Windows NT environment. The actual size of the file "labsnorm.sd2", as determined by WINDOWS, is:



Labsnorm.sd2 5,625KB sas viewer 02/14/2000 1:47 PM

The output from running the macro DsSize using this data set should contain the size in both KB and MB:

```
*---------------------------------------------------------------------------------*;
* Test macro %DsSize on SAS data set labsnorm.sd2           *;
*---------------------------------------------------------------------------------*;

libname datadir "C:\data_analysis";
%DsSize(dsin=datadir.labsnorm);
%put estimate size of labsnorm.sd2 using macro DsSize;
%put size in kb = &kb;
%put size in mb = &mb;
```

The following results are printed in the SAS logs screen:

```
Estimate size of labsnorm.sd2 using macro DsSize,
%put size in kb = &kb;
size in kb =         5625
%put size in mb = &mb;
size in mb =          5.49
```

It can be seen that these results correctly report the size of the dataset. When these codes are used in different operating systems, such as UNIX, the results are all the same since the dataset size is extracted from the intrinsic information of the SAS file.

CONCLUSION

How to accurately estimate the size of a data set is a very important issue to deal with prior to NDA submission. If the size of a dataset is too large, it has to be split into two or more parts according to the specific requirements of NDA. Although there are some equations available to calculate the size of a data set, they are not convenient to use because these equations are dependent on the operating system and are not very accurate under certain circumstances. Here, we have presented an alternative method to calculate the size of the dataset precisely and, more significantly, it utilizes the intrinsic features of the SAS system, thus does not depend on the specific operating system under which the programs runs.

REFERENCES

SAS Macro Language Reference First Edition
Copyright 1997 by SAS Institute., Cary, NC, USA

SAS Language Reference Version 6 First Edition
Copyright 1990 by SAS Institute Inc., Gary, NC, USA

SAS User's Guide: Basics Version 2 Edition
Copyright 1985 by SAS Institute Inc., Gary, NC, USA

Operating Systems-Design and Implementation
Andrew S. Tanenbaum ©1987 by Prentice-Hall, Inc.
A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

ACKNOWLEDGMENTS

The authors would like to thank Eric Zhang, Hong Zhang, Ellen Asam, Donna Usavage and Allan Glaser for their suggestions and assistance during the development of the SAS codes, as well as during the writing of this paper.

CONTACT INFORMATION

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

Author Name: Xingshu Zhu
Company: Merck &Co., Inc.
Address: UNA-102, 785 Jolly road,Blue Bell, PA 19422
Work phone: 484 344 3572
Fax: 484 344 7105
E-mail: xingshu_zhu@merck.com