A SAS® Macro for Comparing the Size of Compressed and Uncompressed SAS Data Sets

Xingshu Zhu and Shuping Zhang, 
Merck & Co Inc., Upper Gwynedd, PA. 19454-2505

Abstract
Knowing the size of the SAS data sets that you are working with is very important for SAS programmers in pharmaceutical companies. First, this information is critical because the size of transport file is one of the requirements for FDA submission [1], and SAS data set and its transport file are very closely related to each other. Secondly, information on data-set size is useful in planning for storage, especially when dealing with large numbers of data sets. Thus, we have developed a SAS macro to perform the task of size calculations for both compressed and uncompressed data sets and to output all of this information into a Word file.

Keywords: Compress, dataset size.

Introduction
A COMPRESS option is often used by a SAS programmer to reduce the size of a SAS dataset. The COMPRESS option can be performed either with the data statement option COMPRESS=YES or with an OPTIONS statement. Although in most situations compression saves space, it is possible in a few cases for compression to actually increase space requirements. The effect of compression on data-set size depends on how many character variables are present in the data set and how many of these variables contain blanks and repeating values. When a SAS data set is composed of mostly numeric variables with only a few character variables, compression will not achieve its space-saving purpose. Therefore, it is important to be able to compare the size of a compressed SAS dataset with an uncompressed SAS data set to be sure that compression has actually saved space.

One way to make this comparison is through information obtained from the program LOG file at the end of the data step in which a compressed data set is created. However, when a programmer is simply given a data set, it does not come with this information. The other way is to use sashelp.vtable. However, the sashelp.vtable does not provide the information about data set size, also the information provided in PCOMPRESS does not accurately reflect the true value on amount of space which might be saved if a data set is compressed. Therefore, we have developed a simple macro that can determine whether a data set is compressed or uncompressed and whether compression saves space. This macro then outputs information on the data-set size, the compressibility (whether the data set is compressed or uncompressed), and the percentage of space that is saved or lost through compression.

Macro %CompSize
 This new SAS macro, %CompSize, was developed to compute the size of a SAS data set, regardless of whether the data set is compressed or uncompressed. The macro, which works both for temporary and permanent data sets, has only one parameter, name of the input data set. This new macro is composed following three components as described below.
1) Determines whether the input data set, \&dsin, is compressed.

The PROC CONTENTS procedure can be used to determine whether an input data set has already been compressed. In the attribute portion of a SAS data set from PROC CONTENTS output, the compressed or uncompressed information of the data set is designated by the item “Compressed,” as shown in the last row of the first column of the table below. Two data sets are depicted in this table: the compressed one is listed in the second column, and the uncompressed one is listed in the third column. A compressed data set is indicated as Compressed: CHAR, while an uncompressed data set is indicated as Compressed: NO.

<table>
<thead>
<tr>
<th>Item in Attribute portion of PROC CONTENTS output</th>
<th>SAS dataset &amp;dsin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Was compressed</td>
</tr>
<tr>
<td>Observations:</td>
<td>52</td>
</tr>
<tr>
<td>Variables:</td>
<td>34</td>
</tr>
<tr>
<td>Observation Length:</td>
<td>8208</td>
</tr>
<tr>
<td>Deleted Observations:</td>
<td>0</td>
</tr>
<tr>
<td>Compressed:</td>
<td>CHAR</td>
</tr>
</tbody>
</table>

The determination of CHAR or NO is read through the PROC CONTENTS procedure and stored in a macro variable named ‘compress’ as shown below.

```sas
ods output Contents.DataSet.Attributes=attr;
   proc contents data=&dsin; run;
ods output close;

data _null_; 
   set attr;
   where upcase(Label2) eq 'COMPRESSED';
   call symput('compress',compress(upcase(cValue2))); 
run;
```

2) Outputs a temporary data set in opposite compressibility from the input data set.

Macro %CompSize uses the data-set option “COMPRESS=” to output a temporary data set, TEMP, which has the opposite compressibility of the input data set. For example, if the input data set is compressed, then an uncompressed data set is output and vice versa. The purpose of this step is to ultimately be able to compare the size of the temporary data set with the size of the input data set.

```sas
data temp (COMPRESS=%if &compress eq CHAR %then NO; %else YES;);
   set &inds;
run;
```

3) Calculates the size in KB for both the input data set and the temporary data set.

To determine whether compression saves space for a particular data set, it is first necessary to calculate the size in KB for the input data set, as well as for the temporary data set with opposite compressibility. As shown below, the ODS option makes it possible to perform this calculation by simplifying the code described in the proceedings from SUGI 27 [2].

The first step is to calculate the size of the input SAS dataset, &inds, and to store the size in the macro variable kb

```sas
ods output Contents.DataSet.Attributes=attr;
   proc contents data=&dsin; run;
ods output close;

data _null_; 
   set attr;
   where upcase(Label2) eq 'COMPRESSED';
   call symput('compress',compress(upcase(cValue2))); 
run;
```

```sas
ods output Contents.DataSet.Attributes=attr;
   proc contents data=&dsin; run;
ods output close;

data _null_; 
   set attr;
   where upcase(Label2) eq 'COMPRESSED';
   call symput('compress',compress(upcase(cValue2))); 
run;
```
The second step is to calculate the size of the "TEMP" data set and to store the size in the macro variable kb_temp:

```
ods output Datasets.Members=comp;
   proc datasets library=WORK memtype=DATA; quit;
ods output close;

data _null_
   set comp;
   where upcase(Memname) eq "TEMP";
   call symput('kb_temp',ceil(File_Size/1024));
run;
```

Examples

In order to demonstrate specifically how macro %CompSize works, as well as to display the results of its output, the macro will be applied here to two different types of SAS data sets. One type of data set contains mostly character variables, while the other type contains mostly numeric variables. In this example, each type of data set also includes a compressed and an uncompressed data set. The data sets containing mostly character variables are designated as c_comp and c_uncomp for compressed and uncompressed, respectively. The data sets containing mostly numeric variables are designated as n_comp and n_uncomp, also for compressed and uncompressed, respectively. The four data sets are listed below.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>File Size</th>
<th>Creation Date</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>c_comp.sas7bdat</td>
<td>65 KB</td>
<td>5/30/2006 10:54 AM</td>
<td>NORTHERAMER</td>
</tr>
<tr>
<td>c_uncomp.sas7bdat</td>
<td>637 KB</td>
<td>5/30/2006 10:54 AM</td>
<td>NORTHERAMER</td>
</tr>
<tr>
<td>n_comp.sas7bdat</td>
<td>25 KB</td>
<td>5/30/2006 10:54 AM</td>
<td>NORTHERAMER</td>
</tr>
<tr>
<td>n_uncomp.sas7bdat</td>
<td>13 KB</td>
<td>5/30/2006 10:54 AM</td>
<td>NORTHERAMER</td>
</tr>
</tbody>
</table>

Notation: c = character comp = compressed
           n = numeric    uncomp = uncompressed

The following is the output created by calling macro %CompSize on the permanent input data set c_comp.sas7bdat with libname data, %CompSize(data.c_comp):

```
Input Data Set: datadir.c_comp

<table>
<thead>
<tr>
<th>Compressibility</th>
<th>Size in KB if Compressed</th>
<th>Size in KB if Uncompressed</th>
<th>% of Space Saved if Compressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressed</td>
<td>65</td>
<td>637</td>
<td>89.80%</td>
</tr>
</tbody>
</table>
```
The output reports for the four data sets are summarized in the table below.

<table>
<thead>
<tr>
<th>Input Data Set</th>
<th>Compressibility</th>
<th>Size in KB</th>
<th>% of space saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>c_comp.sas7bdat</td>
<td>Compressed</td>
<td>65</td>
<td>89.80 %</td>
</tr>
<tr>
<td>c_uncomp.sas7bdat</td>
<td>Uncompressed</td>
<td>65</td>
<td>89.80 %</td>
</tr>
<tr>
<td>n_comp.sas7bdat</td>
<td>Compressed</td>
<td>25</td>
<td>-92.31 %</td>
</tr>
<tr>
<td>n_uncomp.sas7bdat</td>
<td>Uncompressed</td>
<td>25</td>
<td>-92.31 %</td>
</tr>
</tbody>
</table>

When the data set containing mostly character variables in the first two rows of the table is compressed, it frees up 89.80% of the space that it occupies when it is uncompressed. However, as to the data set containing mostly numeric variables, the COMPRESS option not only fails to save space, but it actually “wastes” space. In this case, the compressed data set occupies 92.31% more space than it does in its uncompressed form. Therefore, it is very important for a SAS programmer to correctly calculate the size of a SAS data set and, at the same time, to identify when the COMPRESS option should be used on a SAS data set.

**Conclusion**

Macro %CompSize was developed to automatically calculate the size of both compressed and uncompressed SAS data sets. This macro, which has only one parameter (input data set), is very easy to use. The output of the macro is contained in a Word document that indicates whether the input SAS data set has been compressed or not. It also provides the size of the data set in both its compressed and uncompressed forms. This information is very useful because it allows programmers to determine whether it is worthwhile to apply the COMPRESS option in the data step to reduce the size of a SAS data set.

**References**


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.

**Contact Information**

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

Author Name: Xingshu Zhu  
Company: Merck &Co., Inc.  
Address: UG1D-88, Upper Gwynedd, PA. 19454-2505  
Work phone: 267 305 2689  
Email: xingshu_zhu@merck.com

Author Name: Shuping Zhang  
Company: Merck &Co., Inc.  
Address: UG1D-88, Upper Gwynedd, PA. 19454-2505  
Work phone: 267 305 6933  
Email: Shuping_zhang@merck.com