Is Your SAS Library a Disk Hog?
Here's How to Put it on a Diet.

Ross Bettinger, Modern Analytics, San Diego, CA

Abstract

We have developed a set of SAS macros to deal with big SAS libraries that hog large amounts of disk space. We may put them on a diet by squeezing the unnecessary bytes out of them without losing any accuracy of numeric variables or the contents of character variables. Using these macros may result in significant reductions in disk space. In one case, a SAS library containing 177 datasets which required 44.9 GB was reduced to 21.3 GB, a reduction in disk space of 52.6%.

Introduction

In the normal course of use, SAS libraries change dynamically over time, shrinking and expanding in size to accommodate users' datasets. Some projects don't require much disk space, while others demand humongous amounts for long periods of time. While hardware gets cheaper every day, budget-constrained IT system administrators cannot dash out to the nearest supplier and upgrade server storage to satisfy space requirements as quickly as SAS libraries can increase in size. An especially difficult situation arises when a project of long duration and high space allocation is completed but the datasets in the project's SAS libraries are not archived and moved off-line. We have developed a set of SAS macros to deal with big SAS libraries that hog large amounts of disk space. We may put them on a diet by squeezing the unnecessary bytes out of them without losing any accuracy of numeric variables or the contents of character variables.

The `%SQZ_LIBRARY` macros comprise a set of SAS macros that may be used by system administrators to remove excess space from SAS datasets in a SAS library. These macros automatically process all SAS datasets in a SAS library or a subset of them as indicated by a system administrator. They determine the minimum number of bytes required to contain numeric and/or character variables without losing significant digits or dropping characters.

Using the %SQZ_LIBRARY Macros

There are three macros that comprise the %SQZ_LIBRARY set of macros: %SQZ_LIBRARY, %SQZ_DATASETS, and %SQUEEZE. The %SQZ_LIBRARY macro controls the execution of the other two macros. The invocation of the executive macro, %SQZ_LIBRARY, is given below.

```
%SQZ_LIBRARY( LIBNAME    /* name of SAS library to squeeze       */
            , EXCLUDE=   /* List of datasets to EXclude          */
            , INCLUDE=   /* List of datasets to INclude          */
            , LIST=N     /* Flag for listing of dataset contents */
            , TRIALRUN=Y /* Flag for mode of operation           */
            )
```

The parameters to the %SQZ_LIBRARY macro are listed in Table 1.

Parameter	Type	Default Value	Contents
LIBNAME	Mandatory		Name of SAS library
EXCLUDE	Optional	NULL	Names of SAS datasets in library to *ex*clude from squeeze process
INCLUDE	Optional	NULL	Names of SAS datasets in library to *in*clude in squeeze process
LIST	Optional	N	Flag controlling display of contents of datasets: Y = list, N = do not list
TRIALRUN	Optional	Y	Flag controlling mode of operation: Y = execute %SQZ_LIBRARY in trial status without actually squeezing any datasets, N = perform squeeze operation

Table 1: %SQZ_LIBRARY Parameters

The %SQZ_LIBRARY parameters are discussed below.

- LIBNAME is the name of the SAS library containing datasets that are to be squeezed
- EXCLUDE represents an optional list of SAS datasets in &LIBNAME that are to be *excluded* from the squeeze process. For example, a dataset that is damaged may be excluded, or a dataset that is password protected may be excluded. If &EXCLUDE has been specified, &INCLUDE may not be specified.
- INCLUDE represents an optional list of SAS datasets that are to be *included* in the squeeze process. For example, if you want to squeeze only a limited number of datasets in a SAS library and not every dataset in the library, you would include only those datasets to be squozen [sic] in the include list. If &INCLUDE has been specified, &EXCLUDE may not be specified.
- LIST is a flag that is set to Y if you want to see a PROC CONTENTS listing for every dataset that is processed. You should set LIST=N to see a summary listing only.
- TRIALRUN is a flag that is set to Y if you want to run %SQZ_LIBRARY in test mode to determine if any datasets are damaged or password protected. You must set TRIALRUN=N to indicate that you want %SQZ_LIBRARY to run in "live" mode and perform compression of the datasets in &LIBNAME.

Examples of %SQZ_LIBRARY use are given in Appendix A.

Discussion of %SQZ_LIBRARY Macro Operations

The %SQZ_LIBRARY macro is the executive macro that controls the processing of datasets in the library indicated by &LIBNAME. It performs the following operations:

1. Processes &EXCLUDE and &INCLUDE lists. Ensures that they are mutually exclusive. If they are not mutually exclusive, issues error message and terminates.
2. Creates list of all datasets in &LIBNAME and applies list of dataset names to be excluded or included to set of all datasets. Final list represents those datasets in &LIBNAME to be squozen.
3. Writes an invocation of %SQZ_DATASET to file SQZ_LIBRARY.txt for each dataset to be squozen.
4. Uses the %include statement to include file SQZ_LIBRARY.txt containing %SQZ_DATASET invocations for the datasets to be squozen by the %SQUEEZE macro.
5. Computes performance statistics for each dataset and summary performance statistics for all datasets squozen as a summary report.
6. Writes the time required for %SQZ_LIBRARY execution to the SAS log.

The %SQZ_DATASET macro is invoked once for each dataset that is to be processed. It performs the following operations:

1. It locks access to the dataset to be squozen so that no other process can read from it, write to it, or delete it. If there is an error in applying the lock, the macro terminates with an error message.
2. It opens the dataset to verify that it is not damaged. If the attempt to open it fails, the macro terminates with an error message.
3. It collects pre-%SQUEEZE statistics to be used in performance evaluation of the squeeze process.
4. It passes the dataset name to the %SQUEEZE macro. The %SQUEEZE macro creates the dataset S_Q_U_O_Z_E_N, which contains the compressed contents of the original dataset.
5. The original dataset is deleted and S_Q_U_O_Z_E_N is renamed with the name of the original dataset. If any error has occurred in the squeeze process, the original dataset is not affected.
6. It collects post-%SQUEEZE statistics to be used in performance evaluation of the squeeze process.
7. It unlocks the dataset.
8. It uses PROC CONTENTS to produce listings before and after squeezing, if &LIST=Y.

The %SQUEEZE macro is invoked once for each dataset that is to be processed. It finds the minimum number of bytes required to store numeric variables without any loss of significant digits (accuracy), and the minimum number of bytes required to store character variables without

dropping any characters off the right end of a string (remember: the contents of a character variable are left-justified). The %SQUEEZE macro is well-documented and is available on the SAS website at http://support.sas.com[1]. We originally wrote it in 2004 and generalized it in 2007.

Discussion of Summary Report

The summary report, produced by %SQZ_LIBRARY, is a compilation of the before- and after-squeezing sizes of each dataset that have been squozen and of the SAS library *in toto*. There is also a percent change in size of each dataset and of the library in aggregate computed from the statistics gathered in the course of operation.

Figure 1 contains a brief excerpt of the report for a library of 177 datasets. Negative percentages for reduction in size are due to the fact that the OPTIONS COMPRESS=YES statement was used. See Appendix A for an example of this usage.

Results of Squeezing Library RANDD
By Dataset

Dataset Name	# of Obs	Size Before Squeezing	Size After Squeezing	Percent Reduction in Size
RANDD.DATASET_1	50	73,728	57,344	22.22
RANDD.DATASET_2	50,000	3,907,584	3,612,672	7.55
RANDD.DATASET_3	50	32,768	32,768	0.00
RANDD.DATASET_4	50	24,576	32,768	-33.33
RANDD.DATASET_5	10	16,384	16,384	0.00
RANDD.DATASET_6	37	16,384	24,576	-50.00
RANDD.DATASET_7	6,300	3,645,440	2,834,432	22.25
RANDD.DATASET_8	6,096	3,547,136	2,785,280	21.48
RANDD.DATASET_9	661,276	95,903,744	67,731,456	29.38
RANDD.DATASET_A	156,881	734,535,680	378,871,808	48.42
RANDD.DATASET_B	222,163	21,684,224	11,550,720	46.73
RANDD.DATASET_C	39,955	5,480,448	3,571,712	34.83
RANDD.DATASET_D	23,029	3,170,304	2,203,648	30.49
RANDD.DATASET_E	10,302	3,284,992	2,056,192	37.41

Results of Squeezing Library RANDD
Cumulative Over All Datasets
Squeeze Process Required 5140 Seconds

Library Size Before Squeezing (Bytes)	Library Size After Squeezing (Bytes)	Percent Reduction in Library Size
44,897,779,712	21,284,429,824	52.59

Figure 1: Summary Report

[1] http://support.sas.com/kb/24/804.html

Special Considerations

The %SQZ_LIBRARY macro writes code to the SQZ_LIBRARY.txt file, which is included in the runstream. If you are running the macro on your own computer, e.g., desktop PC or laptop, there ought not to be a permissions issue regarding creating the file. If you are running %SQZ_LIBRARY on a server for which you do not have administrative privileges to your account, you may have to request the services of a system administrator to either create the file for you with read/write permissions or grant you administrative privileges to write the file yourself.

Further savings of disk space may be achieved through the use of the COMPRESS system option. If you put the OPTIONS COMPRESS=YES statement before the %SQZ_LIBRARY macro invocation, you will request that the squozen datasets be further compressed. The discussion of the %SQUEEZE macro on the support.sas.com website includes the effect of using system compression. You may have noticed that small datasets actually increase in size when compressed. This phenomenon is observed in Figure 1 for several of the datasets. However, the size increase (kilobytes) is orders of magnitude smaller than the reduction in disk space achieved for large datasets (gigabytes).

Conclusion

The %SQZ_LIBRARY macros may be used on an ad hoc basis or as a regularly-scheduled job to perform lossless compression on a SAS library of datasets. Large reductions in disk space may be achieved by using these macros, and system throughput will improve due to reduced disk space requirements and faster I/O throughput.

Appendix A: Examples of %SQZ_LIBRARY Use

Examples of `%SQZ_LIBRARY` use are presented below.

1. Trial run to test macro invocation syntax, check for damaged or password-protected datasets

```
%SQZ_LIBRARY( LIB, LIST=N, TRIALRUN=Y )
```

2. "Live" run to perform `%SQUEEZE` compression on `LIBRARY`

```
%SQZ_LIBRARY( LIB, LIST=N, TRIALRUN=N )
```

3. "Live" run to perform `%SQUEEZE` compression on `LIBRARY`, system compression

```
OPTIONS COMPRESS=YES ;
%SQZ_LIBRARY( LIB, LIST=N, TRIALRUN=N )
```

4. Limit processing to a selected set of datasets

```
OPTIONS COMPRESS=YES ;
%let INC_LIST = DATASET_A DATASET_B DATASET_C DATASET_D ;
%SQZ_LIBRARY( LIB,INCLUDE=&INC_LIST, LIST=N, TRIALRUN=N )
```

5. Exclude selected set of datasets from `%SQUEEZE` compression

```
OPTIONS COMPRESS=YES ;
%let EXC_LIST = DATASET_A DATASET_B DATASET_C DATASET_D ;
%SQZ_LIBRARY( LIB,EXCLUDE=&EXC_LIST, LIST=N, TRIALRUN=N )
```

References

"Sample *24804:* %SQUEEZE-ing Before Compressing Data, Redux,"
http://support.sas.com/kb/24/804.html

Contact Information

Ross Bettinger
Modern Analytics
1010 Turquoise St, Suite 250
San Diego, CA 92109

Tel: (858) 488-0771
Fax: (858) 488-0775

RBettinger@ModernAnalytics.com

WWW.ModernAnalytics.com