

Once Again on Efficiency and Base SAS®

Milorad Stojanovic
RTI International

ABSTRACT

Efficiency is a broad term. Different users of SAS Base software understand efficiency in quite different ways, from running programs faster to creating and maintaining application programs more easily and faster. Modifying existing programs by changing program logic requires more effort than other approaches to improving efficiency.

The author's goal was to improve efficiency of SAS programs with a minimum of interference with existing program code. He discusses two SAS Base features – the **COMPRESS** option and the **SASFILE** statement. His goal was to present guidelines that were unavailable and unknown to him at the time of writing this paper about the types of data sets in terms of size and content for which these SAS Base features will improve efficiency. His goal ultimately was to produce a 'cook book' for these two SAS features. Also he points out when and why the options will not achieve any advantage and in fact will produce less efficient results.

INTRODUCTION

Merriam-Webster (1) defines efficiency as 'Effective operation as measured by a comparison of production with cost (as in energy, time, and money)'. To obtain efficiency of SAS programs using Base SAS the author discusses and examines the option **COMPRESS** and the **SASFILE** statement. In his opinion these two SAS features are prime candidates for making programs more efficient with little to no human interference with program logic. "SAS Version 5 stored all data in a fixed length record format." "SAS Version 6 treats an entire observation as a string of bytes and can compress up to 129 consecutive identical bytes into two or three bytes of data" (5).

A few years ago two papers among others were published on Hard Disk Drive Performance – one by Quantum Company (6), and other The PC Guide (7). Both papers clearly explained a hierarchy in computer systems (by speed or by capacity). The fastest components have as a rule the smallest capacity. The order from fastest to slowest is CPU → RAM → Hard Drive.

COMPRESS option

The first question that could be asked is why we need the SAS **COMPRESS** option if we have good and easy-to-use zipping programs like WinZip and PKZIP to name just two of them. They are very efficient in compressing input files including SAS data sets with a compressing ratio close to 95% (depends on data in the files). The answer is simple and a little bit bitter disappointing. We cannot use a zipped data set directly as input to SAS procedures or to data steps. We need to unzip it first.

Option **COMPRESS=YES** has existed for years in SAS Base. This option can be used as a SAS System option like

Options **COMPRESS=YES**;

Or, it can be a SAS data set option in the data step when new a SAS data set is created.

```
Data New_SAS_Data_Set (COMPRESS=YES other options) ;  
    Set old_SAS_Data_Set ;  
    More SAS statements ;  
Run ;
```

The COMPRESS option is not an invention of SAS 9.1.3, but it has not been used a lot. The reasons for this are the following:

1. Some SAS users have never heard about it.
2. Some have heard of it but were not willing to apply it.
3. Some were too cautious because they did not clearly understand all of the consequences of that option.
4. Some users applied it each and every time.
5. Some users applied it when and where it was useful.

First we should point out when you **must not** use COMPRESS=YES option. If you need to read a SAS data sets directly (with POINT option) **please do not use it – it will not work**. Mainly, SAS Base procedures can read SAS data sets generated with COMPRESS=YES with no problem.

This paper attempted to produce some guidelines supported by numeric results from experiments with SAS data sets with and without the COMPRESS option.

All data sets used to collect numerical results for comparison were generated using a random number generator – namely with the RANUNI function. Each observation has a length of 3,000 positions. Data sets were put into five groups:

1. Very small (~ 300 KB), 100 obs.
2. Small (~ 3 MB), 1,000 obs.
3. Medium (~ 30 MB), 10,000 obs.
4. Large (~ 300 MB), 100,000 obs.
5. Very large (~ 3GB), 1,000,000 obs.

There were five different contents of the data sets:

1. Numbers (80 %) and strings (20%) with a large number of unfilled positions.
2. All numeric variables with a small number of blank positions.
3. Numbers and strings, some with a small number of blank positions and some with a large number of unfilled positions.
4. String variables with almost no unfilled positions.
5. String variables with a lot of unfilled positions in comparison to the length of each variable.

Each test was repeated at least five times. Each test includes creating SAS data sets, sorting it and processing with SAS Base procedures (FREQ, MEANS, etc.). Those represent a typical workload in surveys and also in Clinical Trials.

Results of the tests on the very large data sets (average values in CPU and Real time) are presented in the table by data set contents.

Raw data after using COMPRESS=YES option

Very large SAS data sets - 1,000,000 observations

Data type	Proc type	Time type	Time (sec)	Difference (%)
Large numbers and strings	Data step	Real	17.74	9.27
		CPU	-22.53	-31.07
	Freq	Real	-3.62	-4.27
		CPU	-2.87	-51.58
	Means	Real	8.10	10.36
		CPU	-2.66	-47.01
	Sort	Real	-66.95	-7.08
		CPU	-27.37	-32.98

Data type	Proc type	Time type	Time (sec)	Difference (%)
Mix large and small numbers	Data step	Real	65.43	29.73
		CPU	-16.59	-16.17
	Freq	Real	12.63	12.91
		CPU	-6.38	-114.22
	Means	Real	21.17	28.88
		CPU	-6.20	-112.03
	Sort	Real	166.92	16.22
		CPU	-21.87	-25.67
Small numbers and strings	Data step	Real	74.41	30.55
		CPU	-10.44	-9.53
	Freq	Real	43.38	43.53
		CPU	-9.10	-172.49
	Means	Real	40.08	59.30
		CPU	-9.43	-186.22
	Sort	Real	585.85	34.96
		CPU	-16.03	-19.04
Large numbers	Data step	Real	-41.66	-27.45
		CPU	-19.85	-24.66
	Freq	Real	0.08	0.08
		CPU	-1.49	-28.52
	Means	Real	20.36	19.34
		CPU	-1.23	-22.23
	Sort	Real	65.94	6.32
		CPU	-18.20	-19.17
Small numbers	Data step	Real	27.03	13.50
		CPU	-11.78	-7.87
	Freq	Real	79.93	59.64
		CPU	-10.92	-205.34
	Means	Real	38.40	53.50
		CPU	-11.60	-221.96
	Sort	Real	241.45	24.12
		CPU	-9.39	-10.09

COMPRESS=YES improves processing

Content Type	Data Set Size				
	Very large	Large	Medium	Small	Very small
Small numbers & small strings	YES	YES	YES	NO	?
Mix large & small numbers and strings	YES	YES	YES	NO	NO
Large numbers & large strings	YES*	YES*	YES*	NO	?
Small numbers	YES	YES	YES*	NO	?
Large numbers	YES*	YES*	YES*	?	?

YES* = yes with reserve

? = data inconclusive for making decision

Space savings on hard drive (%)

Content Type	Data Set Size				
	Very large	Large	Medium	Small	Very small
Small numbers & small strings	58.31	58.29	58.21	57.14	47.83
Mix large & small numbers and strings	28.49	28.52	28.51	28.08	21.74
Large numbers & large strings	-0.06	-0.06	-0.1	0	0
Small numbers	54.52	54.5	54.42	53.2	43.48
Large numbers	-0.06	-0.07	-0.15	-0.49	-4.35

Data in table above correspond well with timings from the table with raw data on processing times.

SASFILE statement

Another way to improve efficiency of SAS programs is to use SASFILE statement where it is appropriate.

At least three conditions should be satisfied.

First, you should have several invocations of the same SAS data set by different procedures in a sequence. If you don't use the SASFILE statement observations will be read into RAM when each procedure starts execution.

Second, you can't use the SASFILE statement in generating a new SAS data set. The SASFILE statement is going well along with Read and Update actions only.

Third, you should have enough RAM to accommodate your SAS data set.

Some preliminary research showed significant improvement in data processing times in comparison with not using the SASFILE statement.

Raw data from processing using SASFILE statement

Volume	Time type	SASFILE	Time 1 (sec)	Time 2 (sec)	Time 3 (sec)	Time 4 (sec)	Time 5 (sec)
5000	CPU	No	0.03	0.02	0.03	0.01	0.03
		Yes	0.01	0.01	0.01	0.01	0.01
	Real	No	0.03	0.03	0.03	0.01	0.03
		Yes	0.01	0.01	0.03	0.01	0.01
50000	CPU	No	0.08	0.08	0.09	0.10	0.09
		Yes	0.07	0.07	0.07	0.07	0.07
	Real	No	0.09	0.09	0.09	0.11	0.09
		Yes	0.09	0.09	0.09	0.07	0.07
500000	CPU	No	0.77	0.77	0.74	0.77	0.77
		Yes	0.73	0.75	0.75	0.73	0.72
	Real	No	0.78	0.78	0.79	0.78	0.78
		Yes	0.74	0.75	0.75	0.74	0.75
5000000	CPU	No	8.15	8.48	8.16	8.31	8.08
		Yes	8.78	9.21	9.39	9.56	9.27
	Real	No	31.51	29.21	18.93	20.35	20.25
		Yes	50.65	58.26	54.59	64.56	53.29

The next table presents when it is useful to use SASFILE statement and when you should not expect savings in processing time. Results should be taken cautiously and depends on the RAM size of particular PC.

Comparing processing times for different volume of input data sets.

Volume	Time type	Time (sec)	Difference (%)
5000	Real	0.01	46.15
	CPU	0.01	58.33
50000	Real	0.01	12.77
	CPU	0.02	20.45
500000	Real	0.04	4.60
	CPU	0.03	3.66
5000000	Real	-32.22	-133.97
	CPU	-1.01	-12.21

Please note significant increase in processing time when number of observations is 5,000,000 and SAS data set requires 400 MB (more than was available at PC used for measuring).

CONCLUSION

Usage of the COMPRESS option and the SASFILE statement are a great help in improving efficiency of SAS programs. ROI is huge because we need to add just COMPRESS=YES in SAS programs. Results for very small and small data sets were inconclusive for making decision. The author recommends against using the COMPRESS option in those cases.

SASFILE statement is very useful if you have PC with more than 1 GB of RAM. In that case you should use it as much as possible.

LIMITATIONS

This paper is written from the Windows/PC/network perspective. The author does not have enough knowledge and experience to suggest the same approach in other environments (UNIX, MVS, VAX, etc.).

REFERENCES

1. Merriam-Webster's Collegiate Dictionary 10th edition, (1999), page 368.
2. Karp, Andrew, "Indexing and Compressing SAS® Data Sets: How, Why, and Why Not", NESUG 2000, Philadelphia, PA, Sep 24-26, 2000
3. Thompson, Casey, 'PC Performance and the SAS® System', TS-684
4. SAS Institute (2001), 'A Practical Approach to Solving Performance Problems with the SAS® System', Customer Technology Center
5. Understanding SAS Data Compression, <http://nicus.com/newsletter/nsl93sum1.htm>
6. Storage basics, Quantum Corp., http://www.quantum.com/src/storage_basics/
7. Charles M. Kozierok, The PC Guide, <http://www.PCGuide.com>

ACKNOWLEDGEMENTS

I would like to thank Laura Burns from Education Studies Division (RTI) for all her help and support in producing this paper.

CONTACT INFORMATION

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

Milorad Stojanovic
RTI International
3040 Cornwallis Rd
RTP NC 27709
Work Phone: 919-541-7376

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.