

Documenting SAS® Programs: Increasing Programmer Efficiency

Imelda C. Go, Richland County School District One, Columbia, SC

ABSTRACT

A frequently neglected aspect of computer programming is documentation. Systematic program documentation facilitates the reuse of programs and reduces the drudgery when programmers must study or revise programs written by others. The drudgery is not usually because of inability or lack of skill but because of lack of time and inadequate documentation. In work environments that rely heavily on computing, programs are typically reused and revised long after the original programmer has gone. This paper outlines a strategy for systematic program documentation, specifically in the SAS computing environment. It includes ideas that the author has found helpful in conquering the accumulation of programs and printouts. Your needs should guide the development of a system that best works for you. Modifying and improving the system will help it evolve into an organizational tool well worth the thought and implementation.

INTRODUCTION

The need for sufficient documentation stems from the need to understand what was done in the past in order to proceed with present or future work. Documentation is an important but usually not urgent activity. Its importance and urgency are most evident when we look for it and realize there is not enough of it.

1. Documentation records useful information.

Documentation is a record of work that has been done. The documentation of SAS programs involves written guidelines, summaries, instructions, comments, and notes related to the use and function of SAS programs.

2. Documentation prevents future problems.

Documenting programs is a crisis-prevention activity that anticipates future informational needs. Having the foresight to make vital information readily available prevents future delays. It saves time and effort spent on retracing the course of past work.

3. Documentation indicates professional courtesy.

Have you ever tried to put together a jigsaw puzzle without a picture of the completed puzzle or gone to a foreign place without a map or instructions? When you make a program and leave enough documentation, you are leaving a guide for those who need to revisit your program in the future. These people will be grateful for your thoughtfulness. Just remember that you might have to revisit your own programs. When that happens, retracing the course of your own work is one thing. Retracing the course of other people's work can become an annoyance, especially when there is not enough documentation.

4. Documentation is part of good management.

Documenting programs can facilitate the transition between programmers. It can also promote better coordination among programmers who work together on programming projects. Enforcing minimum documentation standards conserves time and human resources for more important tasks.

5. Documentation increases programmer efficiency.

Documenting programs is an organizational skill. When programmers can easily find the information they need, they are more efficient. The transfer of information that links past and present work requires less time and effort. Therefore, the work is accomplished with less effort. Compare this to retracing the course of past work and trying to answer questions that would not have been asked if there had been enough documentation.

Your answers to the following questions are worth considering before planning an organizational system:

- What type of documentation information do you typically need? (Consider meeting these needs as part of your minimum documentation standards.)
- What standards can facilitate the continuity of work between different programmers?
- What type of standards are you willing to enforce?
- How much documentation is too much and how little documentation is too little? (The answer to this question depends on your situation. Leaving a "big picture" or general overview can be useful. Details have their virtue, but too much can be overwhelming and even unnecessary.)

Remember the following when considering the suggestions in this paper:

• Be flexible

Systems are often dynamic and standards may need to be adjusted and modified over time. Optimizing the functionality of the system is a journey and not an end in itself.

• Use your judgment

The amount of documentation you need is related to your work environment. In some places, computing plays a very central role and in others, it plays an auxiliary role. Consider the holistic nature of your work environment. Standards should be situational. The effort you expend should be commensurate with the importance of the program you are documenting. Common sense dictates that programs are not to be documented equally. Some require no documentation while others require much more.

SUGGESTIONS

The proper reuse, modification, or improvement of a pre-existing program requires knowledge of the program. Documentation should provide information so that the reuse, modification, or improvement can proceed without much delay.

1. Cover the basics

Think of the important specifications and features that should be shared about a program. Commonly referenced information should be strategically located to reduce information retrieval time.

- Where is the program located? What is its file name?
- Who wrote the program? What is the program for?
- When was the program written?
- What are the prerequisite conditions that need to be met before using a program?
- What are the essential elements of a program such as the input and output data sets?
- Where are the data set layouts and the variable definitions?
- What are the assumptions that the program makes about the data?
- What does the program produce? Where are the samples of the output from the program?
- What should people know in order to replicate a result/product from the program?
- Where are the paper and computer files related to the program?

2. Provide a "big picture"

Several programs can be involved in a project. How do these programs relate to each other and how should they be used relative to each other? Are they to be used independently or in sequence? How do these programs relate to the nature of the project? A brief summary (e.g., diagram, figure, flow chart) about how the programs relate to the overall process for completing a project can be very useful. For example, as part of quality control procedures, checking the integrity of data requires analyzing the data using several data-checking programs. Only after the data have been checked for irregularities can the analysis programs be used.

3. Inform others

Does the program use a particular technique or methodology in one of its routines? What should a person trying to modify your program know? Are there any caveats about using or revising the program that you should remind others about?

4. Provide commentary

Traditional ways of documenting a program include annotating a hard copy of the program with handwritten comments. You can use SAS comment statements to type your comments into a program. You can also incorporate comments into the TITLE and FOOTNOTE statements. This documents the program internally (program statements) and externally (output) at the same time. However, some of the output might be required for formal presentation or official use and not all types of comments are appropriate.

5. Be intuitive

Using intuitive or meaningful SAS names can eliminate some of the guesswork in determining what a SAS name represents. They are also easier to remember. Accepted acronyms or ordinary words could be a good choice for intuitive SAS names.

Example 1:

How would you name a variable for the average daily sales (ADS)? ADS or SALES are possibilities. Simpler is sometimes better and SALES might be a better choice for those who prefer to remember SALES instead of an acronym. SAL is not a very intuitive choice for a name. It is almost the word SALE and might cause people to instinctively think of SALE instead of SAL.

Example 2:

Consider a variable that has as values a subject's birth date. Some possible SAS names for it are:

BIRTHDAT
BRTHDATE
BDAY
BIRTH

Choosing one of the above options may have its advantages and/or disadvantages depending on your point of view. BDAY and BIRTH are quite easy to remember while BIRTHDAT and BRTHDATE are not as straightforward.

6. Use labels

SAS names are not always intuitive names and they can become codes in themselves. Variable names and SAS data sets could use good labels. Use the LABEL statement to assign labels to variables and the LABEL= data set option to specify a label for a data set. This information is displayed when PROC CONTENTS is used to print the contents of a data set.

7. Be consistent

Standardize or create your own conventions in terms of how things are done. For example, it can be very convenient to give a variable that occurs in different programs the same SAS name in the different programs. Otherwise, you can encounter the same variable that goes by different names depending on which program you find it in.

8. Be explicit

The DATA= option could be consistently used with procedures to provide explicit input data set information. When it is not used, the most recently created data set is used as the input data set for a procedure. When the DATA= option is used, the relationships between procedures and data sets are very clear. This can be advantageous when there is a need to reuse code or a need to revise lengthy code that involves multi-step data set manipulation.

9. Use a tracking code

Create a coding system that helps keep track of your reports and/or programs. An example would be to have a system that identifies the programmer, date, and project. John Smith used the "quotas" program for the Wilson Food Co. in May, 1996. This could be coded as: WFC (JS, 9605, quotas) and included in a TITLE or FOOTNOTE statement. Some time later when you encounter a page of SAS output, you can identify (via the title/footnote code) its exact source. You can include whatever information is relevant to your needs. Other elements could be the SAS file name, the input data set, etc.

10. Use shorthand

Sometimes SAS names are actually real words. When you write comments, the SAS names that are real words could interfere with other words in the comments. In some cases, it is even convenient to have the same name for different SAS components: data sets, ordinary variables, macros, or macro variables. One way to avoid the confusion is to introduce shorthand that indicates what you are referring to. You can attach a meaningful prefix to a SAS name when you are referring to it in your comments.

Example 1:

Use the following prefixes:

- d. when referring to a data set,
- gmrv. when referring to a global macro variable,
- mv. when referring to a macro variable,
- m. when referring to a macro, and
- v. when referring to an ordinary variable.

You can refer to prefixed SAS names and at the same time indicate the context in which you are using them. This also distinguishes between real words and SAS names that are real words when both are used in your comments. This prefix system eases the use of a different letter case for the SAS names especially if you prefer to type in strictly capital or lowercase letters. You achieve brevity and conciseness without the loss of meaning.

Example 2:

SAS comment without shorthand:

```
/* Macro DRUG uses the global macro
variable DRUG. The global macro
variable DRUG is the drug of choice
for drug addicts in the study. */
```

SAS comment with shorthand:

```
/* M.drug uses gmrv.drug. The
gmrv.drug is the drug of choice for
drug addicts in the study. */
```

11. Leave a snapshot

Leaving a "picture" of the input data format can be useful. A data "ruler" and "template" can be used as tools to make quick revisions in the INPUT statement when the data layout has not been finalized. Intended changes in the data layout can be reflected on the template. This can then be used as a guide for revising the INPUT statement in terms of which columns contain the data and what type of data are involved.

Example:

Using a "ruler" and "template" can give you an idea of what the input data should look like at a glance. The SAS code below shows a ruler for 20 spaces. The line after the ruler is a data template that describes the data required by the INPUT statement. The # stands for numeric data and the \$ stands for character data. From the ruler and the template, you can immediately see that the values of numeric variable x are in columns 1 to 4 and the values of character variable y are in columns 6 to 8.

```
data example;
input x 4. +1 y $ 3.;
/* The following line is the ruler.
12345678901234567890
#### $$$
*/
cards;
1234 ABC
;
```

12. Use efficient programming

When you have less code, there is less to document and explain. However, efficiency in the brevity or number of statements could sacrifice straightforwardness and readability. Sometimes it is easier to use more programming statements than to use less. Using macros tends to reduce the number of statements but decrease the readability of a program. You might prefer to find a reasonable balance between the readability of statements and efficiency.

13. Consider your client

You might have to adjust your programming style to suit your client or the person who might eventually maintain your program.

Example:

Macros are very useful and if you decide to use macros, determine how much you need to document them. If the person who will maintain your program resists the use of macros, it might even be better to develop the program using macros and then turn over the final program without any macros.

14. Be organized

Professional librarians can teach us a few things about archiving and information storage/retrieval.

Some ways to arrange printouts include the:

- Categorical order (putting like things together)
- Sequential order (by the order in which output is generated)
- Logical order (by the order that makes sense according to your purposes)
- Chronological order (by date and/or time)
- Practical or functional order (by the order in which output is used more efficiently relative to other work tasks)
- Combination of different types of orders (by the order that works for you)

When you have determined the best way to arrange papers and documents, consider the following techniques to help you retrieve and locate specific items more easily than flipping through an endless pile of paper:

- Number pages (It does help when individual pages are numbered.)
- Serialization (For example, go beyond numbering individual pages and number groups of printouts, have a volume for each month, etc.)
- Alphabetization
- Documents are dated
- Creating tables of contents and indexes
- Using dividers and tabs
- Color coding
- Clear labeling

15. Reduce paper

In trying to reduce the volume of paper you have to maintain, consider the storage implications of the printouts generated by your programs. Some possibilities for paper reduction are:

- Printing more text on a page (reduce the font size to get more characters per inch, reduce the margins, increase the page size, increase the line size)
- Non-traditional printing such as duplex (two SAS output pages to a physical page) or quadruplex printing (four SAS output pages to a physical page)
- Electronic instead of paper copies of output
- Periodic elimination of documents that no longer need to be kept or permanent storage of documents that no longer have to be immediately available
- Suppressing unwanted output

Example:

You may only need an output data set from a procedure. Use the NOPRINT option to eliminate the default output.

- Limiting the number of observations being printed

Example:

You may need to see a few records in a data set and have a sample of what its variables look like. Use the FIRSTOBS= and/or OBS= option to limit the records to be used by PROC PRINT instead of printing the entire data set.

- Being selective by knowing the nature of different SAS procedures, their features, and capabilities

Example 1:

You can use PROC UNIVARIATE and PROC MEANS to obtain certain statistics for numeric variables. Let us say your only interest is in the mean, minimum, maximum, and number of observations. Instead of using the default output for PROC UNIVARIATE which uses one page per variable, use:

```
proc means mean min max n;
```

You will get the desired statistics with one row of statistics per variable. Less paper was used and you got what you needed. However, PROC UNIVARIATE provides more types of statistics than PROC MEANS.

Example 2:

The default output for two-way to n-way cross-tabulations in PROC FREQ can be very lengthy and include percentages that you may not need. Use the LIST option with the TABLES statement to have output in list format rather than in crosstabulations.

Example 3:

PROC TABULATE can incorporate information from procedures such as PROC FREQ and PROC MEANS into a single table.

- Using customized output instead of default output from SAS procedures

Example 1:

Use the DATA _NULL_ step instead of PROC PRINT to print the observations in a data set. This gives you maximum control for customizing a report or printout.

Example 2:

You can use output data sets from procedures, manipulate them, and use a DATA _NULL_ step to customize the way they are printed.

Example 3:

You can manipulate SAS data sets and use a DATA _NULL_ step to produce output compatible with a word processor's features. Data to be used in tables can be written in comma-delimited form. These lines can be imported into a word processor and easily converted into a table.

Try not to keep everything. We are in the information age and we must learn to keep what is important and throw away the rest. When other people need to revisit your files, it is a great courtesy on your part to let them just deal with the necessary papers.

CONCLUSION

The suggestions in this paper were written in general terms and without reference to a particular type of program. The use of the suggestions is to be considered in the context and purpose of the programs you are documenting and the level of documentation your work environment needs for a particular program.

Systematic program documentation is an organizational tool that can reduce superfluous and mundane work. This ultimately creates time for more important tasks other than trying to answer questions that would have had readily available answers had there been enough documentation in the first place.

REFERENCES

SAS Institute Inc. (1990), *SAS Language Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS Procedures Guide, Version 6, Third Edition*, Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

I would like to thank David R. Hall from Florida International University's Office of Planning and Institutional Research (Miami, Florida). I wrote the tracking code suggestion based on his idea. He uses a tracking code as a standard in his office. His office generates an enormous number of reports and that code has saved them much time and effort in matching the output to one of the hundreds of programs they maintain.

TRADEMARK NOTICE

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

Comments, criticisms, and suggestions are always welcome. Please contact:

Imelda C. Go
Office of Research and Evaluation
Richland County School District One
1616 Richland St.
Columbia, SC 29201
Tel.: (803) 733-6079
Fax: (803) 929-3873
icgo@juno.com