

## To Macro or Not To Macro?

Amy Gillespie, Merck Research Laboratories, Blue Bell PA  
Shuping Zhang, Merck Research Laboratories, Blue Bell PA  
Hong Qi, Merck Research Laboratories, Blue Bell PA  
Fang Chen, Merck Research Laboratories, Blue Bell PA

### ABSTRACT

As employees and businesses, we are often challenged to complete our work with a “do more with less (or same)” attitude. We are expected to increase productivity by being more efficient in our jobs, yet assuming a constant high level of quality. Process improvement initiatives are vital for success. SAS® programmers have opportunities to make valuable contributions, primarily by building departmental SAS macro libraries to standardize programs for analyses, report generation, and graphing. However, as SAS programmers, we may be questioned as to whether the amount of work needed to build a generalized SAS macro library is time well spent. To investigate this issue we developed metrics on our department’s SAS macro library. This paper will discuss our metrics research efforts, present our results, and demonstrate how these important findings aid in our SAS macro library development.

### INTRODUCTION

Do SAS macros really help us do our job more effectively?

Are we more productive by utilizing SAS macros from a departmental macro library?

What are the tradeoffs of spending large amounts of time and resources to develop a generalized departmental macro versus a smaller, less robust macro for use by one project at a time?

To answer these questions, the effectiveness of our department’s SAS macro library was evaluated. Our goal was to measure the productivity and efficiency of SAS macros used for statistical analyses and reporting by collecting a series of metrics.

This paper will briefly introduce our macro library followed by a discussion of our metrics collection approach, the results we determined, and the future plans for our department’s macro library. Our research indicates that SAS macros can lead to greater productivity and higher quality results.

### MACRO LIBRARY OVERVIEW

In 1998, a committee comprised of statisticians and statistical programmers was formed by management to develop standard statistical tools for analysis and reporting. A primary objective of this committee was to develop and maintain a departmental SAS macro library. As of 2004, the macro library contains 79 macros, including graphical, data manipulation, statistical analysis, utility, document formatting, and text handling SAS macros. The library contains both small simple utility macros as well as large, complex, and robust statistical analysis macros.

The development efforts for the large and complex statistical analysis macros often require a significant time investment by statisticians and statistical programmers. Time is needed to determine macro specifications, implement code, test, prepare user documentation, and ultimately train users. On average, these development efforts require at least 1 year since developers are often balancing project work with macro creation. Since such a large time investment is needed, the resulting macro must be efficient, utilized, and of high quality.

During the past 6 years, several large macros have been added to the library. For example, in 1999 a Rich Text Formatting (RTF) macro was added to the library. This macro enabled users to create RTF (e.g., MS Word ready) tables with a relatively simple SAS macro call and eliminated the need to create tables using data \_null\_, proc report, etc. In 2000, the first large statistical analysis macro was added to the library. This macro completes analyses of non-inferiority (one-sided equivalence) between two proportions and produces a report-ready table for easy insertion into a study document. This equivalence macro

was developed according to project needs and as a result was utilized within at least 3 large drug and vaccine regulatory submissions .

The RTF and Equivalence macros are two examples of successful macro development efforts . Such success was measured by collecting metrics.

## METRICS COLLECTION

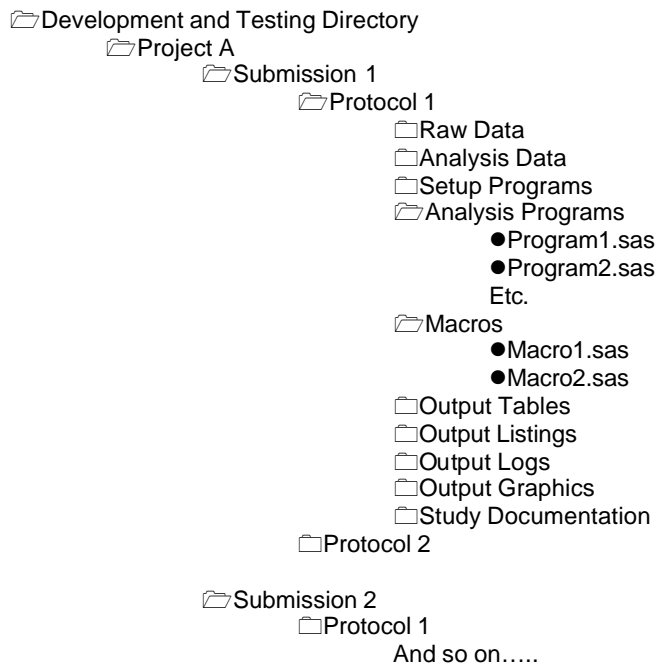
Metrics on the department macro library were collected using both quantitative and subjective measures . Among the metrics collected were:

- Number (%) of projects, clinical study reports, etc. using macros
- Number (%) of tables in a submission generated using macros
- Number (%) of individuals utilizing macros
- “Time savings” achieved through macro use (programming and validation)
- Macro library quality
- Macro library usage

## QUANTITATIVE METRICS COLLECTION

To measure the efficiencies of macros quantitatively, we extracted the frequency counts of macro usage between 1998 and 2004. We counted usage on 2 different programming areas which included a development and testing directory and a regulatory submission directory (e.g., stores all programs/data sent to a regulatory agency). The directories for each of the programming areas follow a hierarchical structure. An example of the structure is included in Figure 1 below.

**Figure 1**  
Sample Directory Structure



In 2004, the development and testing area (pictured above) consists of 57 projects which includes programming in support of several regulatory submissions, clinical study reports, interim analyses, publications, and exploratory ad-hoc requests. Given the hierarchical directory structure and the amount of files and programs available, a SAS macro was required to electronically extract the frequencies of macros used.

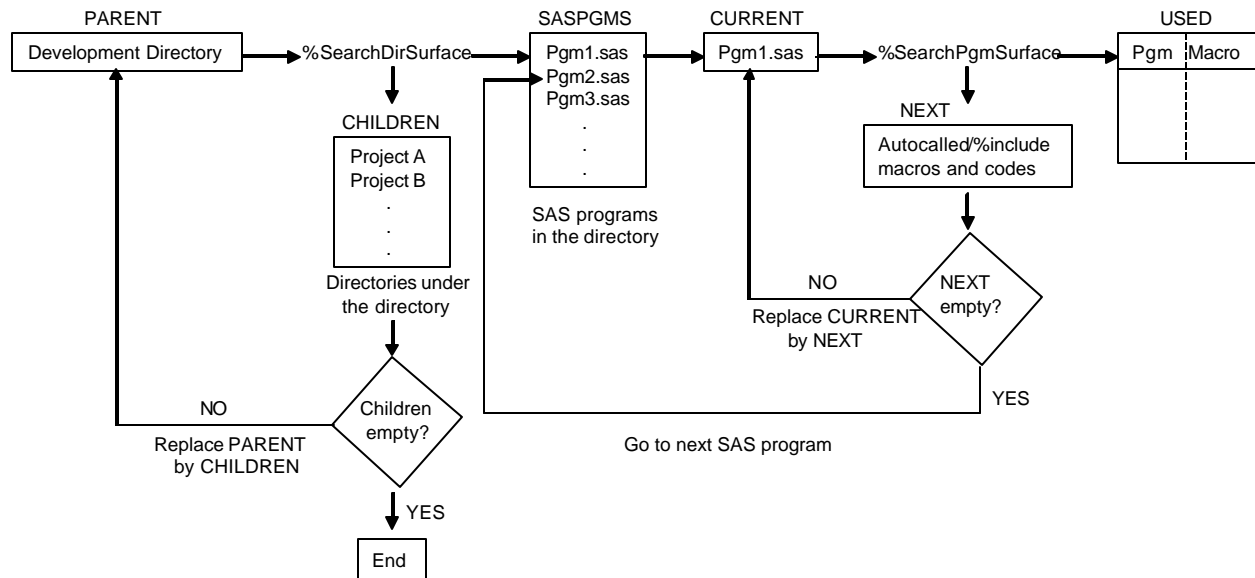
The macro developed is quite sophisticated. The macro loops through the different directory levels shown above , as well as SAS programs and macros, to extract the number of library macros called or included in a SAS program . Generally, the macro implements the following program flow:

- (1) Search program surface for autocall macros and %include macros.

- (2) For each autocall macro and %include macro identified above, search code for additional nested macros.
- (3) If additional autocall macros and/or %include macros are identified, continue to search through these macros for more nested macro calls.
- (4) Continue looping through programs and macros until there are no additional autocall or %include macros found.

Steps 1-4 are repeated for every program within a protocol (%SearchPgmSurface). A second loop is also used to search each of the programming directories (%SearchDirSurface). Figure 2 illustrates this complex counting logic

**Figure 2**  
Macro Searching Logic



Once the extraction macros are finished, an analysis dataset is created including the following variables (Refer to Figure 3).

- Pgmdir - location of the program
- Pgmname - name of the program
- Pgmdate - date the program was created
- Macname - name of the macro used by the program
- Count - how many times the macro is used by the program

**Figure 3**  
Metrics Analysis Dataset

Obs	Pgmdir	Pgmname	Pgmdate	Macname	Count
1	S:\test\project1\prot1	Pgm1.sas	23Oct2003	.	.
2	S:\test\project1\prot1	Pgm2.sas	04Dec2003	RTF	1
3	S:\test\project1\prot1\macros	Pgm3.sas	12Apr2002	.	.
4	S:\test\project1\prot1\macros	Pgm4.sas	21Dec1998	.	.
5	S:\test\project1\prot1\macros	Pgm5.sas	14Jul1999	.	.
6	S:\test\project1\prot1\macros	Pgm6.sas	01Oct2003	Center	6
7	S:\test\project1\prot1\macros	Pgm6.sas	01Oct2003	Commatxt	1
8	S:\test\project1\prot1\macros	Pgm6.sas	01Oct2003	Equiv	3
9	S:\test\project1\prot1\macros	Pgm6.sas	01Oct2003	Exact	1
10	S:\test\project1\prot1\macros	Pgm6.sas	01Oct2003	Index	3
11	S:\test\project1\prot1\macros	Pgm6.sas	01Oct2003	Nobs	3
12	S:\test\project1\prot1\macros	Pgm6.sas	01Oct2003	NumCols	8
13	S:\test\project1\prot1\macros	Pgm6.sas	01Oct2003	Parse	23
14	S:\test\project1\prot1\macros	Pgm6.sas	01Oct2003	Qwords	10

This analysis dataset is used to create various metrics analysis tables and graphs. The results are discussed in a following section.

### SUBJECTIVE METRICS COLLECTION

In addition to the quantitative metrics identified above, subjective measures were obtained by using a metrics questionnaire to assess the efficiency, quality and effectiveness of macro use. The questionnaire was distributed to all of the statistical programmers, statisticians, and managers in our department through E-mail with a web link. As an incentive, a money raffle was offered to all those who responded. Information on five categories was collected with close-ended questions, including

- survey participants background, such as job role, department, number of years with company and the name of the projects worked on,
- macro use by macro type (statistical, utility and other), macro name, and type of analysis (efficacy, immunogenicity, safety, exploratory and others),
- time saved in programming and validation with macro in the categories of 0%, 1-25%, 26-50%, 51-75%, 76-100%, and N/A,
- reliability/accuracy of the macro, and
- user friendliness and robustness of macro evaluated in the grades of poor, fair, good, excellent, and N/A.

At the end of the survey, the responses were downloaded from the web to an Excel spreadsheet, and then transferred to SAS to be summarized with tables and graphs.

### METRICS RESEARCH RESULTS

The following is a subset of the tables and figures created for the macro metrics research.

**Table 1**  
Programming Effort

Category	Tables		Programs	
	n	%	n	Effort Ratio
Total	574		156	3.68
With macros	501	87.2	122	4.11
Without macros	73	12.7	34	2.15
% is calculated as # of tables in category / total number of tables *100				
Effort Ratio = # of tables / # of programs				

Table 1 indicates that 156 programs were developed to create 574 tables. This programming was in support of 3 regulatory submissions (3 regulatory submissions were chosen as a representative sample). Of these 156 programs, 122 utilized a department macro. Therefore, each program, with a macro call, created 4 tables on average. However, for those programs which did not utilize a macro, only 2 tables were created per program. This result may indicate that programs utilizing macros are more efficient because code can be used over and over. Programming time and resources are being saved by using a macro. Additionally, department macros may be used for more repetitive and routine analyses, while stand-alone programs (those without macros) are needed for more special and unique analyses.

**Table 2**  
Frequently Called Macros for 3 Submissions

		CSR Tables Filed (N=574)	
Macro Type	Name	n	%
Statistical	%Exact_CI	322	56.0
	%Equiv_PP	247	43.0
Utility	%Numcols	303	52.7
	%Varattr	247	43.0
Other	%RTFTable	465	81.0
	%RTFSymb1	307	53.4
	%Parse	306	53.3
	%QWords	286	49.8
	%NObs	249	43.3
	%IndexC	247	43.0
	%Setwords	247	43.0
	%Uniwords	247	43.0
%Words	247	43.0	
% is calculated as n/N *100			

Table 2 displays a list of frequently called macros for 3 regulatory submissions. Of the 574 tables created, approximately 50% of the tables were created using a macro to create exact confidence intervals (%Exact\_CI), as well as a macro for equivalence analyses (%Equiv\_PP). Additionally, 81% of the tables were created by an RTF formatting macro (%RTFTable). These numbers indicate that considerable programming efficiencies were achieved since the same macros are routinely used to generate the tables sent to the regulatory agency.

**Figure 4**  
Percent of Projects (N=57) using a Macro

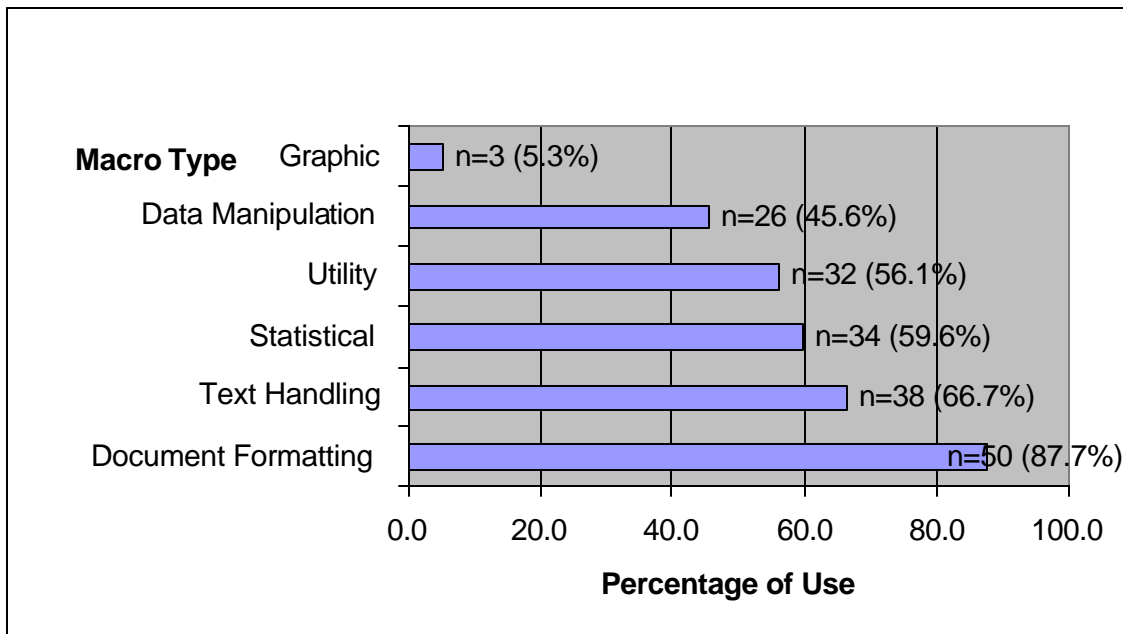


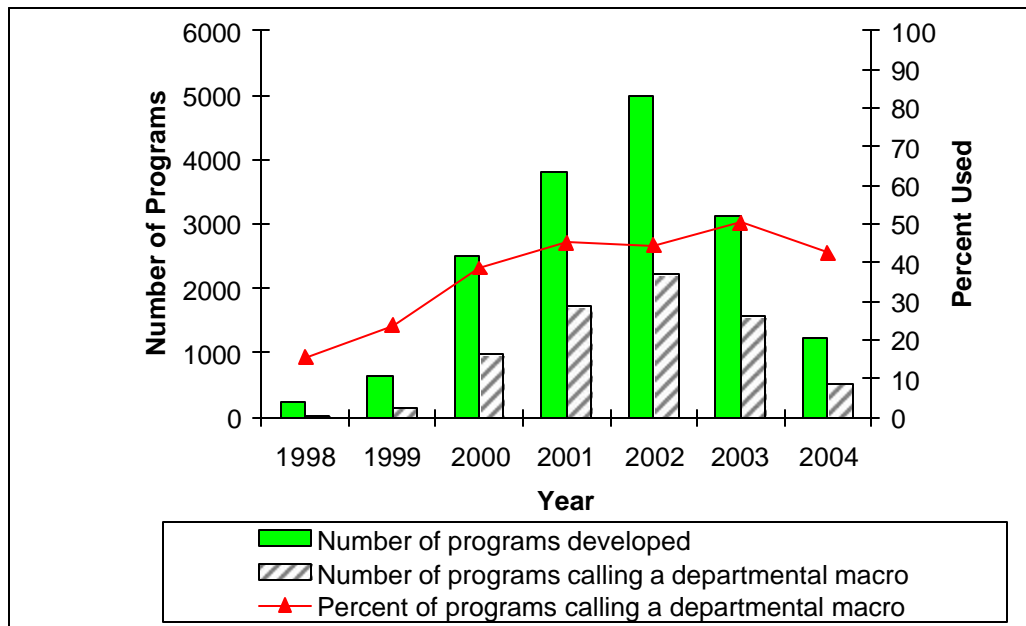
Figure 4 shows metrics on the development and testing directory. Of the 57 projects, 50 (87.7%) of them used a document formatting macro. However, only 5.3% of the projects used a graphic macro. This may indicate that users do not need graphic macros, or the available graphic macros are not meeting user's needs. On the other hand, document formatting macros are critical for programming and are extensively used. These are important metrics for future development work.

**Table 3**  
Volume of Macro Calls

Macro Type	Number of Macros Being Called (N=176,750)	
	n	%
Text Handling	124,297	70.3
Document Formatting	25,097	14.1
Utility	19,734	11.1
Statistical	5,304	3.0
Data Manipulation	2,265	1.2
Graphic	53	<1
% is calculated as $n/N \times 100$		

Table 3 shows that there were more than 176,000 macro calls on the development and testing directory! This number includes nested macros according to a hierarchical searching process.

**Figure 5**  
Percent of SAS programs using Macros over Time



The percent of SAS programs using macros have increased over the years. However, the overall number of SAS programs developed has decreased in 2003 and 2004. This trend may be attributed to programs created for completing larger, more complex tasks and macros developed for generating multiple tables per program. As a result, the quantity of programs developed is smaller, yet the complexity and productivity per program is higher.

**Figure 6**  
Programming Time Savings for Statistical Macros

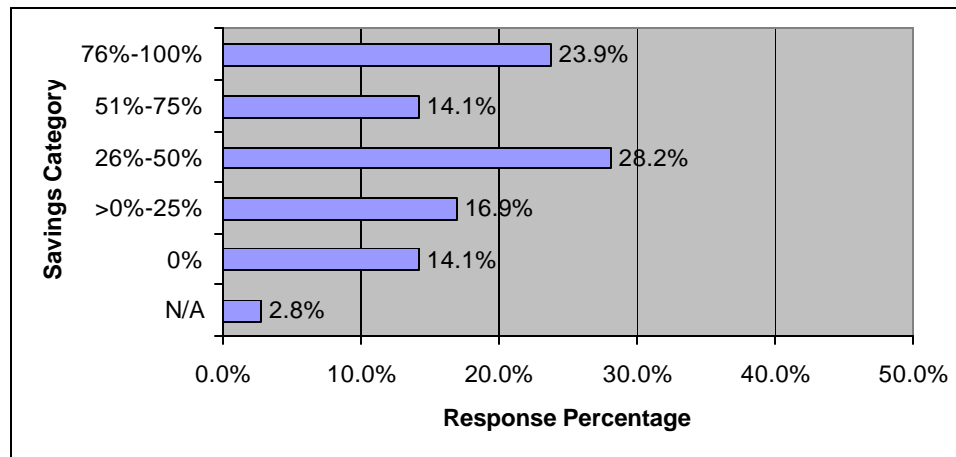


Figure 6 is an example of the survey metrics. This particular survey question was:

Estimate how much development and programming time was saved by using a statistical analysis department macro.  
(Example: by using a macro, you were able to complete a normal 10-hour programming job in 5 hours; therefore you saved 50% of your normal programming time by using a macro).

- 0% - no programming time saved using a macro
- 100% - no additional programming time needed because the macro was used
- N/A - never used a macro

The results in Figure 6 indicate that statistical analysis macros save programming time. Of the survey respondents, approximately 83% conclude that statistical macros save some portion of programming time. Although the initial programming of a macro can take a lot of effort, the ability to reuse code ultimately results in savings and productivity. The payback period for development time is relatively short since savings per project are high.

Similar results were found for validation time savings .

## SUMMARY

Our research identified an increasing trend in the number of programs utilizing macros since the department macro library was established in 1998. Among the macros developed between 1998 and 2004, the most widely used macro and the most widely used statistical macros were recognized along with the frequency of macro use. The needs and usage of macros were shown to vary by different job functionality, for example, large macros may serve programmers well by achieving a high degree of programming efficiency, but may prohibit the use by statisticians due to the longer start-up time; statistical macros are more frequently used in one department, while utility macros are more frequently used in another department.

The programming efficiencies are being achieved as a result of the repeated use of macros for similar and routine analyses and tabulations. The survey results also indicated a great amount of time saved in programming and validation, supporting the findings from the objective data collection. Concurrently, due to the use of validated macros, the programming qualities are being improved with respect to reliability/accuracy, user friendliness and robustness as revealed in the survey result. Assuming that the productivity is a function of efficiency and quality, i.e.,  $P = f(E, Q)$ , where P is the productivity, E is the efficiency and Q is the quality, the improvements in programming productivity are being reflected by the improvements in programming efficiency and quality. It is understood that programming productivity in macro use is also influenced by other factors, such as macro development efforts and time spent. Our library macros have been developed using the available time between project responsibilities. Some small macros were directly derived from the project or routine work which did not involve more than one developer. Large macros were more likely to engage a group of statisticians and programmers and take longer time to develop. However, from the cost-effectiveness point of view, our analyses showed resources well spent in the macro development. A comment from a survey responder summed up macro development as "one unrecognized benefit of the efficiencies gained by macro development effort is the fact that department personnel now have the opportunity to focus on the more scientific issues of clinical trials."

The exploration of metrics of SAS macro usage is an innovative research. Our analyses provided insights into the advantages of developing and utilizing SAS macros. It provided evidence that the departmental macro library meets a

business need to do more with less (or same). In addition to the metrics collected, it helped identify the direction of future macro development. Most importantly, it will also have great impact on SAS programming practice in pharmaceutical industry in the future.

## **ACKNOWLEDGMENTS**

The authors thank the members of the Statistical Tools for Analysis and Reporting Metrics subcommittee for their ideas and knowledge in developing metrics for the Merck Blue Bell Biostatistics department.

## **CONTACT INFORMATION**

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

Amy Gillespie, MS  
Manager, Scientific Programming  
Merck Research Laboratories  
P.O. Box 4, BL3-7  
West Point, PA 19486-0004  
Phone: (484) 344-3383  
E-mail: [amy\\_gillespie@merck.com](mailto:amy_gillespie@merck.com)

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.