Organizing and Building a Centralized SAS® Macro Library

Snow Fu, Merck Research Laboratories, West Point, PA
James Wu, Merck Research Laboratories, West Point, PA

ABSTRACT

In many pharmaceutical companies, writing and sharing SAS macros has become common practice in the statistical programming area. To properly manage this proliferation of SAS macros, it seems more and more necessary for a pharmaceutical company to build and maintain its own centrally controlled SAS macro library. These libraries may contain many types of macros, including statistical, utility, text-handling and data-manipulation macros. In this paper, we will discuss some suggestions for effectively organizing and building a centralized SAS macro library.

KEYWORDS

SAS macro library, approval process, approval team, standard directory structure, user documentation template, validation

INTRODUCTION

A centrally controlled SAS macro library stores macros that serve as standard tools for supporting statistical planning and analysis and the presentation of results. A macro library can provide a form of quality control for the macros that it maintains. By centralizing these macros into one easily accessible place, a library can also minimize rework and repeated validation efforts by programmers and statisticians. Thus, a SAS macro library can become an important resource in facilitating the preparation of regulatory electronic submissions.

To maintain the central control of a macro library, an approval team can be formed, whose main responsibility would be to approve and add new macros to the library and to maintain the existing ones. Only the team members should have read and write access to the library while other users should only be able to read.

There could be many new macros ready to be added to the library each year. Once a macro has passed key validation tests, which may include a developer’s test and a validator’s test as required by common programming SOPs, it can be passed on to the approval team for final approval.

In this paper, we will provide an overview of the type of approval team and macro approval process that could be used in a SAS macro library. We will also offer some suggestions for developing a library’s macro directory structure and user documentation template.

GENERAL APPROVAL PROCESS

We propose that the process is most efficient and effective if the approval team is led by a chair and consists of a couple of members representing different therapeutic areas (if applicable). The team can meet regularly to discuss issues related to library maintenance and more importantly, decide which macros are ready for approval. It can then assign the task of approving particular macros to its individual members.

The first step toward approval is the code review, which focuses on whether the macro coding satisfies guidelines and standards. Some issues that may be considered during this review are whether the header is complete; whether the comments are clear, appropriate and easy to read; and whether the macro description is accurate.

The next step is the documentation review, which considers the completeness of both the user documentation and more importantly, the validation documents. One key question that arises during this type of review is whether the macro has been properly validated. Another question is whether the validation files are correctly organized in the appropriate folders. Since the macros in the library will be used quite extensively across all functions, it is important to try to control their quality by assuring that proper validation was performed on each of them prior to their inclusion in the library.

Based on our experiences, we can offer some suggestions on how to facilitate the approval team’s efforts to ensure that a macro has been properly validated. These suggestions apply to the process of developing and validating a macro, which begins well before the involvement of the approval team.
Before the development efforts are initiated for each macro, we suggest that a **programming requirement specifications (PRS)** document be prepared. The PRS document would clearly state the requirements of the macro, which at a minimum may include macro functionality, data source and assumptions, logic assumptions and output specifications. The purpose of this document is like that of a cookbook – to guide the developer/tester through the entire development/testing process. This document can be a collaborative effort between programmers and statisticians (if applicable).

With a well-defined PRS in one hand, the macro developer can start the programming work with accuracy. Once the code for the macro program is complete, **developer testing** could follow. Developer testing helps to ensure that the program satisfies the requirements outlined in the PRS. The testing activities may include testing the program with different data scenarios and preparing documentation to formally record that testing has occurred.

After the developer’s test is complete, we recommend that the validation process continue with **independent programmer testing**. The goal here is to assure that the program accurately and reproducibly provides the information outlined in the PRS. Any programmer or statistician other than the developer could be a good candidate for this task. The main activities for an independent programmer review may include verifying that the program code is easy to understand and follow, and examining the results of developer testing to determine whether it has been performed on complex and important algorithms. If any major problems are detected during the independent programming review, the reviewer should convey these findings to the macro developer so that the program can be revised accordingly. It is also highly recommended that the reviewer include a document that records the testing process.

If the macro submitted for approval is a statistical macro, then **statistical peer review testing** can also be performed. The ideal person for a statistical review is an independent statistician. The goal of a statistical review is to ensure that the output is statistically sound and accurate. The suggested activities for statistical peer review may include confirming that the statistical procedure used is the most appropriate choice for the specific task defined in the PRS, and verifying that the results are correct. If necessary, the statistical reviewer can convey validation findings to the developer and assist the developer in making necessary modifications to the macro. Once again, formal documentation of the testing is also suggested.

If the development and validation process recommended above is properly followed, the approval team’s job will be made much easier. After all, the primary responsibilities of the approval team are to ensure that all required validation is performed and that the validation documents are completed and stored in an appropriate folder.

Finally, as the last step in the approval process, we advise that a member of the approval team perform his or her own small-scale **approver’s validation**. The major tasks involved in this validation would be quite similar to what takes place during the independent programming review.

Once the macro has been approved and added to the library, the approval team will continue to work on improving the library. Occasionally, feedback from end-users will trigger some small revisions or enhancements of already approved macros. The best practice that we can suggest is to set up frequent training sessions on some of the most commonly used macros and encourage end-users to submit comments or inquiries through a centralized communication channel. The feedback would then be analyzed and the approval team would work with designated personnel to make sure that all the issues are addressed promptly.

**MACRO DIRECTORY STRUCTURE**

The success and maintainability of the macro library also depends on the overall structure of the macro file directories, the naming conventions used within the macro and the location of all relevant files (such as programs, log files, output files and others). Because of all the documentation that is important to the approval process, it is very helpful to create a logical directory structure that makes the necessary information easily accessible. It is also helpful to develop a structure that is reusable for each macro, like the proposed one displayed in Figure 1.
The description of each sub-directory is explained in the following table:

<table>
<thead>
<tr>
<th>Sub-Directory/File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apvtest</td>
<td>Contains the approver’s test files.</td>
</tr>
<tr>
<td>Devtest</td>
<td>Contains the developer’s test files.</td>
</tr>
<tr>
<td>Pgmtest</td>
<td>Contains the independent programmer’s review files.</td>
</tr>
<tr>
<td>Peerrev</td>
<td>Contains the statistician’s review files (if applicable).</td>
</tr>
<tr>
<td>Document</td>
<td>Contains the macro user manual, training documentation and program request specification.</td>
</tr>
<tr>
<td>Macrolib</td>
<td>Contains the source code of the newly developed macro and the nested macro files.</td>
</tr>
<tr>
<td>Development-log.doc</td>
<td>The development-log file will serve as a road map to record the major milestones in the macro’s development process (e.g. revision or enhancement). It is the developer’s responsibility to prepare and update the log file if necessary.</td>
</tr>
</tbody>
</table>

**Table 1 - Sub-directory description**

The more detailed sub-directories within each development and review sub-directory are described in Table 2.

<table>
<thead>
<tr>
<th>Sub-Directory / File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Contains the input data used in the test program(s).</td>
</tr>
<tr>
<td>Pgm</td>
<td>Contains the test program(s).</td>
</tr>
<tr>
<td>Out_listings</td>
<td>Contains the .lst file(s) generated by the test program(s).</td>
</tr>
<tr>
<td>Out_logs</td>
<td>Contains the .log file(s) generated by the test program(s).</td>
</tr>
<tr>
<td>Out_tables</td>
<td>Contains the Word table(s) generated by the test program(s) (if applicable).</td>
</tr>
<tr>
<td>Out_graphics</td>
<td>Contains the graph(s) generated by the test program(s) (if applicable).</td>
</tr>
<tr>
<td>Supporting forms</td>
<td>Contains all sorts of validation forms as required by SOPs.</td>
</tr>
</tbody>
</table>

**Table 2 - Detailed sub-directories within each development and review sub-directory**
We feel that the suggested file structure would facilitate macro development, approval and maintenance. It is easy to move, archive and locate information on a specific macro. When the same structure is used consistently for each macro in the library, it becomes fairly easy to locate information across all macros.

USER DOCUMENTATION TEMPLATE

For many of the statistical and complicated macros, a well-written, uniform user document becomes a must for end-users. To establish a user document that is both comprehensive and uniform for all macros, we suggest that a standard user documentation template be created. Adopting such a template could eliminate the time each macro developer spends trying to determine the best format for presenting macro information.

The template could be a standard Word document that begins with an overview section, which provides a brief summary of the macro functionality. The type of analysis and/or output generated by calling the macro could be covered in a section on basic features following the overview. A good portion of the user document template would be dedicated to the macro’s syntax and usage. For a statistical macro, the statistical method would also need to be explained in a separate section. In addition, for all macros, the template might designate a place in the user document for a few detailed examples to help end-users understand the macros more easily. Finally, the template could provide for contact information, statistical references and appendices.

The suggestions that we have offered are basic points to be considered in the creation of a user documentation template. Depending on the specific requirements of the individual company, there are many other possibilities that can be explored in designing the template. However it is ultimately designed, we recommend that the template be stored in a commonly accessed area of the macro library, with reading and writing privileges available to all users in order to assist future macro developers.

CONCLUSION

This paper presents some ways of organizing and building a centralized SAS macro library to facilitate the work of SAS programmers. A macro library can be an important time-saving resource for programmers, who can avoid the inefficiency of recreating existing macros by accessing them from the library. The approval process that we proposed for accepting macros into the library serves as a form of quality control because it is aimed at ensuring that programming SOPs for validation are followed.

This approval process is able to proceed more efficiently through the help of a standard directory structure, which creates a logical and organized system for storing the documents needed for a macro’s approval. This structure can also benefit the macro developer and validator by providing them with a system to organize their files. In addition, a user documentation template can assist a macro developer or development team to quickly create a standard, user-friendly and comprehensive user document.

Our goal is to explore ideas for enhancing the way we interact with SAS. The suggestions that we have provided may inspire our readers to look for business solutions that best address the needs of their companies.

ACKNOWLEDGMENTS

The authors acknowledge Donna Usavage, Ellen Asam, Margaret Coughlin, Amy Gillespie and Allan Glaser for their strong support in the publication of this paper.

CONTACT INFORMATION

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

Snow Fu
Merck Research Laboratories
P.O. Box 4, BL 3-2
West Point, PA 19486
(484) 344 3656 (phone)
(484) 344 2931 (fax)
E-mail: Snow_Fu@merck.com

Jian (James) Wu
Merck Research Laboratories
P.O. Box 4, BL 3-2
West Point, PA 19486
(484) 344 2841 (phone)
(484) 344 2931 (fax)
E-mail: James_Wu2@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.