

Paper PO-087

Architecting a Regulatory Compliant Macro Library using SAS Drug Development

Virginia Redner, Merck Research Laboratories, North Wales, PA
Margaret M. Coughlin, Merck Research Laboratories, North Wales, PA

ABSTRACT

Centralized macro libraries can provide efficient solutions to project programming and analysis needs. They also may provide consistency of statistical methodologies, programming techniques and implementation of industry standards across projects. In the pharmaceutical industry, programming is required to comply with 21 Code of Federal Regulation (CFR) Part 11. This requires traceability of changes to programs or macros. Additionally, validation of programs or macros is needed to ensure they perform the functionality intended.

SAS Drug Development (SDD) provides a regulatory compliant repository. This paper addresses the challenges of architecting a centralized macro library stored in SDD for use in analysis and reporting.

INTRODUCTION

The centralized macro library is designed to be used primarily with late stage clinical studies. Macros contained in the library support the analysis and reporting for safety and efficacy studies, as well as regulatory deliverables. The design of the library is adaptable so that other areas of support can be added as needed.

The focus of the paper will be on the unique challenges of implementing a SAS macro library in the SDD web-based repository versus on a network share. It will also address design and maintenance considerations in order to meet regulatory compliance. In the next few sections of this paper the implementation of a macro library residing in SDD that supports PC-SAS based analysis and reporting will be presented.

SDD PLATFORM

SDD is implemented on a UNIX platform utilizing an Oracle 9i backend. The SDD user interface is web-based. Alternately, SDD may be mapped to appear like a network share drive using Xythos software. The SDD environment provides controlled access, audit trails and automatic version control. These features serve to promote the regulatory compliant implementation of a macro library on the SDD platform. They are either unavailable or less flexible on a network server platform where macro libraries are most frequently implemented. The network server platform in itself does not provide a 21 CFR Part 11 compliant implementation of a macro library.

DESIGN OF MACRO LIBRARY

SDD is the official repository for the global macro library. Our SDD library consists of a production area which is versioned to trace all updates to the macros. Our testing is done on the PC and stored on a network drive until complete. All official macro validation documentation is stored in the production area on SDD to enable global access, should the documentation evidence be required during an audit.

On occasion, clinical trial programmers may need to work remotely, and library macros may need to be submitted to a regulatory agency where they will most likely be run on the PC. When designing a macro library on SDD, it is important to assure the macros are portable and can be run in the PC environment, as well as the Unix based SDD environment. One convention that is helpful in this regard, is to limit the parameter values accepted by library macros to only SAS assigned filename references and SAS data library references, rather than path names. Whereas the format of path names change based on the operating system, SAS file name and library name references are used the same across operating systems, enabling the use of path names via the references. Passing only previously allocated filename references, assigned using the SAS FILENAME statement, and SAS data library name references, assigned using the SAS LIBNAME statement, into the macros, is a step that can be taken toward assuring macros can be run in both the SDD and PC environments.

Another important decision to be made when designing a macro library is whether to store the macros in compiled or uncompiled format. SDD supports either of these approaches. One benefit of storing macros in compiled format is increased performance. When library macros are stored in uncompiled format and the autocall facility is used to access the macros, all the macros in the macro library are compiled each time a SAS session is started. Macros accessed using the %INCLUDE statement are compiled each time the %INCLUDE statement is executed.

Eliminating these compilation steps can increase performance, especially when working with large macro libraries.

Another reason some organizations decide to store their library macros in compiled format is to block users from creating alternate versions of the macros. In order to run the macros, all users must have read access to the macro library. When macros are stored in un-compiled format, there is nothing preventing users from copying the standard macro code into their SAS session, or into another folder, and changing the macro code prior to execution. Storing the macros in compiled format eliminates the ability to access the source code, and therefore prevents users from changing the standard macros.

Enabling access to standard library macro source code is essential to some organizations, where it is considered common practice to copy and alter standard library macros for project or protocol specific needs. In these organizations a better design may be to store the macros in un-compiled format, and apply a process oriented approach to monitor the creation of alternate versions of the standard macros.

DIRECTORY STRUCTURE OVERVIEW

The directory structure is the organizational mechanism for the macro library. SDD provides flexibility in the design of the macro library directories. This flexibility allows us to design a macro library that not only stores macros, but also user documentation, sample data sets and calling programs, and validation documentation. Subdirectories can be created to logically group related macros. This is important to help both new and existing staff quickly locate needed macros. Figure 1 gives an overview of the macro library directory structure.

SDD

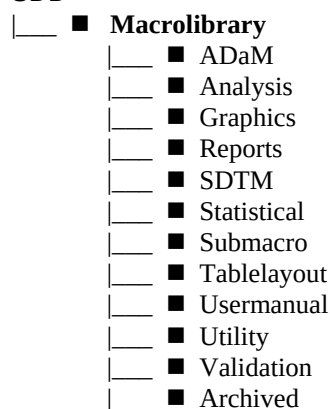


Figure 1 - Overview of macro library directory structure

The description of each directory is explained as follows:

Directory name	Description
ADaM	Create CDISC ADaM datasets
Analysis	Create business logic: data derivations to prepare data before a table layout macro is called.
Graphics	Graphics macros
Reports	Create safety and efficacy reports used across all studies
SDTM	Create CDISC SDTM datasets
Statistical	Standard statistical analysis macros
Submacro	Macros called by a few main macros
Tablelayout	Create standard table layouts used across studies
Usermanual	User documentation
Utility	Macros and utilities used globally across studies, such as output format macros, startup macros
Validation	Sub-directories containing validation documentation for global macros
Archived	Archived (retired) macros and their associated validation and user documentation kept in accordance with record retention policies

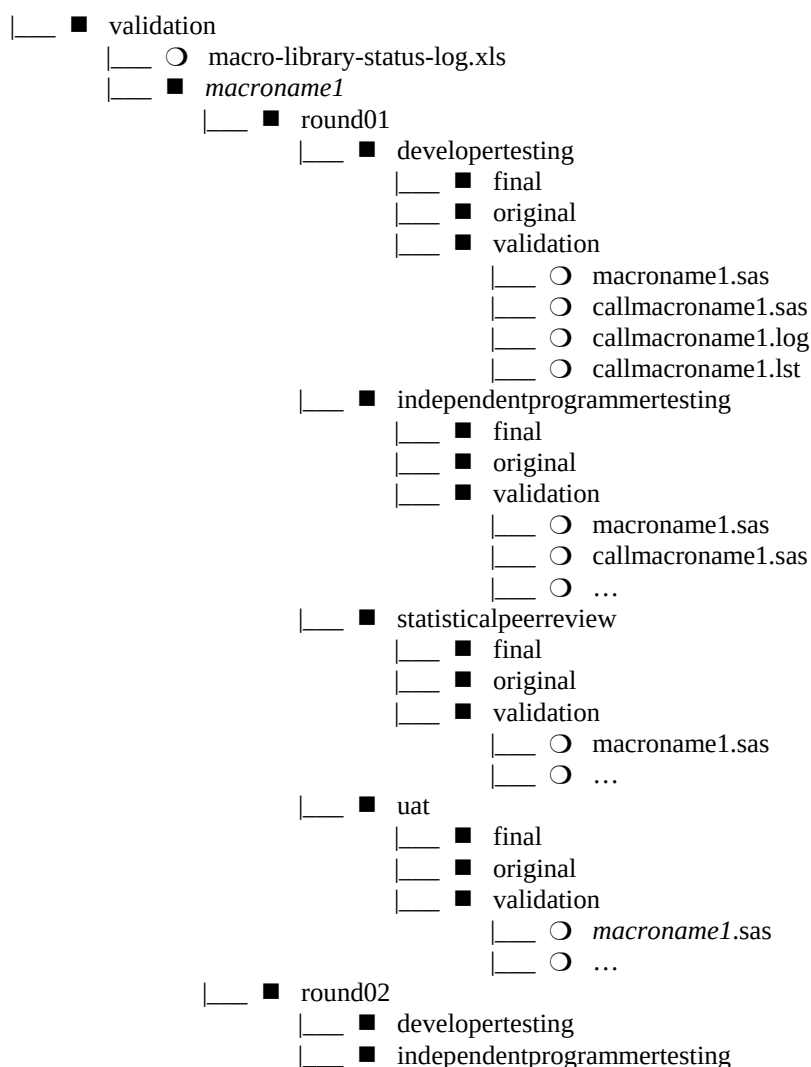
Figure 2 - Directory descriptions

The next section will provide details on validation of centralized macros.

VALIDATION OVERVIEW

As mentioned earlier, the validation for our global macros is done on the PC and stored on a network drive until validation is complete. Once a macro is fully validated the validation documentation is moved to the versioned production SDD environment for final storage.

Having approximately 200 users of our global macro library requires that our macros follow an intense validation process. The process involves up to three separate levels of validation; developer validation, an independent peer review, and for statistical macros, a statistical peer review, followed by a user acceptance test. For each level of validation all testing programs, testing data, output and SAS logs from testing are saved as proof of validation. Following is an example of the validation directory structure.



As shown above, our validation structure contains separate folders for each level of validation. Each of these validation level folders is replicated for each round of validation. The macro code being tested is stored in the folders called "original" prior to each level of validation, and the version of the macro at the end of each round of validation is saved in the folders called "final". To better track a macro in development, a spreadsheet (macro-library-status-log.xls) is used to clearly define the various rounds associated with a given macro release. Using SDD as a

repository for our validation documentation assures traceability of all of these documents, providing evidence for each level of validation.

ADVANTAGES OF USING SDD

A major advantage of implementing a macro library in SDD is that it provides a centralized repository. In today's global working environment, this can be a significant benefit over the use of network servers. Performance considerations of global macro library access to network servers often results in distributed implementation of macro libraries across multiple local servers. Multiple copies of the same macro library across local servers must be kept in sync, causing considerably more maintenance than the centralized SDD repository solution.

The single SDD repository also provides centralized control and standardization of the macro library. All global developers access the exact same macros for their analysis and reporting. A single repository provides greater control and ease of macro library maintenance. Moreover, a single centralized global macro library facilitates standardized macro development. As we have seen in the previous section, the validation process is facilitated by SDD and promotes standardization. Standardized macros assist in making analysis and reporting consistent within and between clinical trials.

Another advantage of using SDD as your macro library repository is the availability of version control of objects or files. Automatic version control of macros in SDD allows previous versions of the same macro to be stored and accessed from the same directory. Version control facilitates reproducibility of an analysis since all previous versions of a macro are always available in the SDD repository.

Changes to macros are automatically reported in the SDD audit trail, which records the macro name and location, as well as date of change and userid of the individual making a change. SDD provides a searchable audit trail that users granted the policy to access will find valuable. This audit trail information on macro changes may be advantageous in that it can aid staff in better understanding the changes by providing identification of individuals who may be able to provide greater information and in pinpointing the exact date and time of the change. In a network server environment, audit trails are not easily searchable by staff. Most importantly, SDD audit trails meet regulatory compliance requirements.

SDD provides sophisticated access controls that can be applied to directories or individual files. Not only can access to specific directories and files be limited to individual users or user groups, but the ability to even see the directory or file can be controlled in SDD. This provides great flexibility to the macro library to be able to shut off files or directories from view. For example, if a macro is retired, there is no need to remove it from the macro library. Moreover, due to versioning it cannot be removed. The permissions on this retired macro can be set so that specific users or user groups cannot see or access the macro. Permissions can be set to allow staff involved in macro maintenance to continue to see and access the macro. Network servers do not allow this level of control to be easily implemented.

DISADVANTAGES OF USING SDD

The version control feature of SDD provides several disadvantages that need to be taken under consideration when implementing a macro library in SDD. Versioned macros provide access to all previous versions of a macro. No access control capability is available to limit user access to only the current version. This increases the risk of inadvertent use of an old version of a macro when the most current version is warranted. A versioned macro cannot be renamed. Should renaming of a macro be justified, a new copy of the macro needs to be created with another name. Unfortunately, the link to the old macro name is lost. Confusion could result over which macro to utilize since versioning maintains both old and new macro names. Thus, versioning serves to direct how macros can be managed in the SDD environment.

CONCLUSION

There are important considerations regarding the design of a global macro library that should be addressed when architecting a regulatory compliant macro library using SDD. Implementation of a macro library in SDD provides some advantages and some disadvantages over implementation on a network share due to SDD's technical implementation and regulatory compliance features.

SDD may benefit your macro library implementation if you require a regulatory compliant macro library repository, with version control and audit trails, and/ or if you need to get staff in multiple locations onto one platform.

REFERENCES

Handelsman, David. January 2006. *SAS Drug Development Vision and Roadmap*. Cary, NC: SAS Institute, Inc.

Handelsman, David. 2006. "SAS Drug Development: Bringing Innovation to Life Sciences". *PharmaSUG 2006 Proceedings*

James, Margaret; Maass, Carolyn; Redner, Virginia. 2006. "Developing and Managing a SAS Macro Library". *PharmaSUG 2006 Proceedings*

Kenny, Susan. 2005. "Strategies for Implementing SDTM and ADaM Standards" . *PharmaSUG 2005 Proceedings*

CONTACT INFORMATION

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

Virginia Redner
Merck Research Laboratories
P.O. Box 1000, UG1CD-44
North Wales, PA 19454-1099
Phone: (267) 305-7634
Fax: (267) 305-6474
Email: virginia_redner@merck.com

Margaret M. Coughlin, MS, MPH
Merck Research Laboratories
P.O. Box 1000, UG1D-88
North Wales, PA 19454-1099
Phone: (267) 305-6813
Fax: (267) 305-6474
Email: margaret_coughlin@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.