

SASMAX – Maximize potential of reusing SAS code

Beate Hientzsch, Accovion GmbH, Eschborn, Germany

ABSTRACT

In statistical programming units of pharmaceutical companies and contract research organizations (CROs), it is common practice to reuse SAS[®] code across projects. For complex standard programs and macros designed, validated and maintained by a standards group, this is typically a well-defined process.

However during daily routine, all kinds of SAS programs initially written for single use are copied, modified and adapted from one project to the next to reduce time for development and validation, potentially leading to a bunch of slightly different versions.

It's a challenging question for each SAS programming unit: How can one ease and standardize the reuse of SAS code, i.e. collect know-how from a group of individuals, set up a transparent architecture, strengthen awareness of what is available and optimize retrieval, but avoid at the same time an overload of corresponding administrative activities?

This paper presents the implementation of the Accovion SAS program library SASMAX[®], the underlying concepts and technologies as well as its Web-based interface designed to support efficient import and retrieval of SAS programs and macros.

INTRODUCTION

Diversity and complexity of tasks that need to be handled by statistical programming units of pharmaceutical companies and CROs are affected by growing standards and requirements specified by the authorities and the pharmaceutical industry. For CROs, diverse requirements issued by different sponsors further increase task variety.

In this highly regulated environment, controlled sharing and reuse of SAS code becomes more and more essential to save time and resources and most notably to ensure a high-quality level of the results and outputs produced.

The idea of sharing SAS code is very popular and can be realized differently. It is common practice to copy SAS programs from one project to the other or to share SAS code by chance. That typically leads to a bunch of slightly different versions and tracing back changes over projects and over long periods becomes challenging.

Accovion came to the conclusion that there was an urgent need to set up a process for the convenient sharing of expertise and know-how between SAS programmers. The process also needed to fit well within the multi-client and multi-project CRO business environment. Primarily the intention was to define a centrally managed platform, however it quickly became obvious that an adequate user interface was also essential. This paper explains how the SAS program library SASMAX supports sharing of SAS code, the application of standard macros through the software development life cycle and how SASMAX contributes to the training of SAS programmers.

PREREQUISITES

SAS PROGRAMMING ENVIRONMENT

Accovion's SAS programming environment is mainly located on a Unix server using a standardized directory structure defining the filing system for programs, macros, original and derived datasets and outputs. To meet all customer requirements, usually 2 productive SAS versions are available in parallel. Currently these are SAS versions 8 and 9. During project initiation, the respective SAS version is fixed and the working environment is set up accordingly. Each customer's project is assigned a separate location on the server and access rights are defined as appropriate.

PhUSE 2007

CONFIDENTIALITY

Confidentiality of customers' data is assured by applying a strict strategy for granting user access rights: SAS programmers only have access to customers' projects they are working on. Therefore it is only possible to view SAS code for projects to which they are assigned.

SAS PROGRAMMERS

The most important factor that needs to be considered when building a standardized programming strategy is the programming staff itself. At Accovion, SAS programmer turnover is low, nevertheless in a group of more than 20 SAS programmers at 2 sites, changes in staff happen and backup via contractors is common. New team members usually have to be a valuable help after a short training period.

Staff need to be able to change quickly from one project to another especially in CRO business. Therefore internal processes need to be optimally standardized and expertise needs to be shared efficiently to meet different customer requirements.

Know-how must not be dependent on the presence of single individuals, but should be shared to avoid major impact on current or future projects in the event of staff turnover.

DESIGN CONCEPT

SASMAX is designed as a project and customer-independent library of SAS programs and consists of different types of SAS programs and macros that can be shared.

The architecture is set up to handle 3 different types of SAS programs/macros:

- ✓ Generic macros
- ✓ Sample programs
- ✓ Tools

The goals and visions that go along with the different types of SAS programs and macros will be discussed in detail in the following sections.

GENERIC MACROS

Generic macros are designed, developed and thoroughly validated for usage as standard macros across different projects. These generic macros are maintained centrally throughout the software development life cycle. The respective version of a generic macro is applied by multiple users without changes.

Initially the objective of avoiding redundant copies of these macros' source or catalogues and particularly the potential risk of changes to the source brought Accovion to the initiative of setting up a SAS program repository. Changes to the source of a generic macro, whether accidental or intentional, cancel the macro's validation status. That's why setting up a standardized, secure and transparent process for the use of generic macros was deemed indispensable.

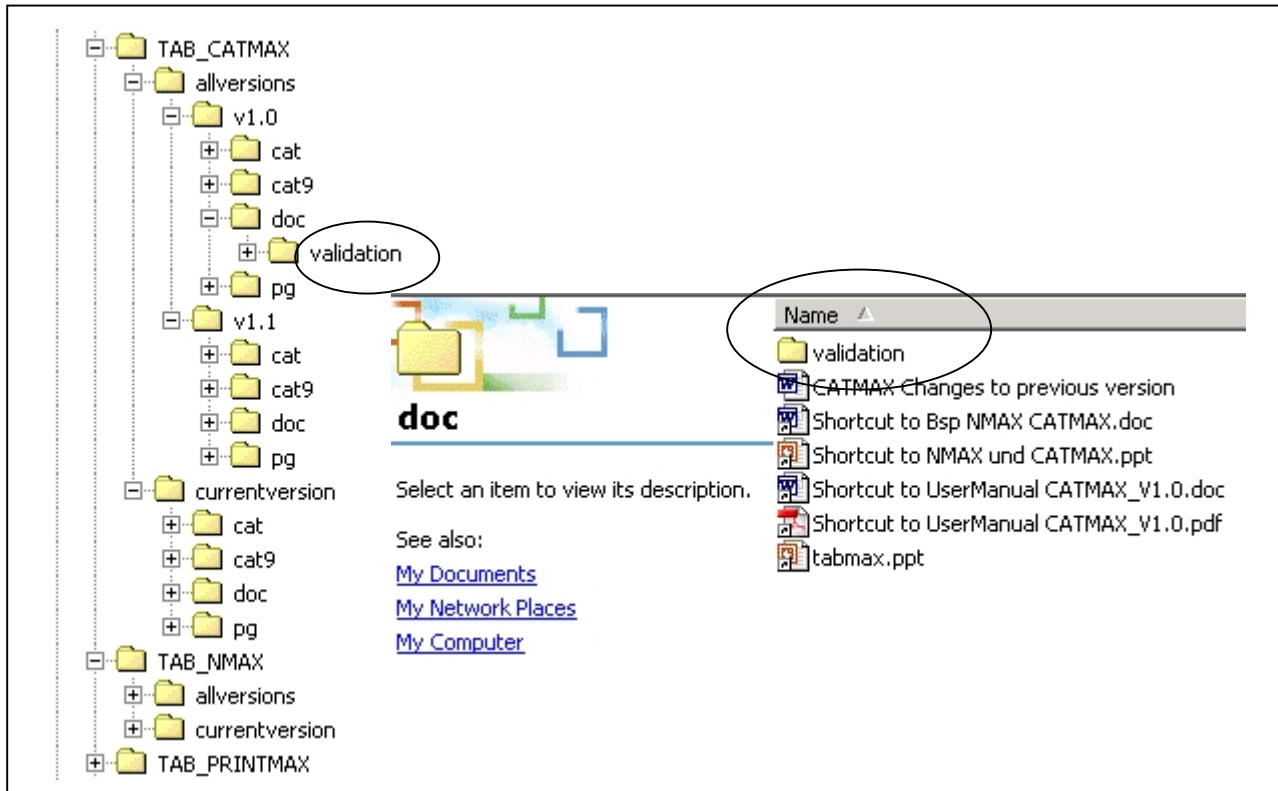
SASMAX generic macros must meet the following essential requirements:

- ✓ All versions need to be accessible, i.e. the current version as well as all former versions. This is necessary to be able to rebuild results achieved with former versions of the macro. Nevertheless each new version of a standard macro should be downwardly compatible.
- ✓ All corresponding documentation, i.e. user manual, examples, validation documentation, needs to be available together with the macro's source and the compiled macro catalog.
- ✓ Generic macros need to be available and validated for all SAS versions currently installed in the productive environment.
- ✓ Generic macros need to be available as plain SAS source for transparency and as macro catalogs for execution.
- ✓ Generic macros must not be copied to the respective study or project, but run from the working environment automatically using the correct version of the macro as well as the correct SAS version.
- ✓ A version stamp should be printed to the log file of the calling program for documentation and transparency.
- ✓ A stamp indicating that an official version of a generic macro was used should also be printed to the log file.

The directory structure for generic macros is set up to meet the requirements above and to allow as much transparency as possible. In order to avoid redundancies, links to files are used instead of copies, e.g. a user manual that did not change from one version to the next is accessible through a link to the original file.

PhUSE 2007

The extract below shows the directory structure for a single generic macro:



For the availability of generic macros from different projects without copying a SAS macro, USEGENMAX was developed by Accovion as part of the SASMAX library. Provided that for each version of a generic macro a compiled SAS macro catalog exists, references to all required SAS macro catalogs can be generated while initializing a SAS session. USEGENMAX concatenates the respective references and assigns the name `_sasmax_`.

Example how to use USEGENMAX:

```
...
%usegenmax (name=macro1);                               * Name of the Macro (mandatory);

%usegenmax (name=macro2, version=currentversion);       * Version (optional),
                                                         version=currentversion is the
                                                         default;

%usegenmax (name=macro3, version=v1.0);
...
```

The name of each generic macro and its version need to be handed over to USEGENMAX, whereas the current SAS version is automatically retrieved via the SAS system variable.

The extract of the USEGENMAX macro below illustrates its functionality:

```
%MACRO usegenmax (name=, version=);
...
* define libname referencing the directory where the required version of macro catalog can be found;
p =
"/SASMAX/generic/" || TRIM(UPCASE("&name")) || "/" || TRIM(UPCASE(&version)) || TRIM("/&catm")
;
CALL SYMPUT("libpath",COMPRESS(p));
LIBNAME __genmac "&libpath" access=readonly; * path to selected catalog of generic macro;
```

PhUSE 2007

```
*****;
* Append libname to _sasmax_;
*****;

%IF (%SYSFUNC(LIBREF(__genmac))) %THEN %DO; * selected version of macro does not exist;
  %PUT %SYSFUNC(sysmsg());
  %PUT WARNING: __GENMAC=&libpath;
  %PUT WARNING: Selected Version of Macro does not exist;
%END;
%ELSE %DO;
  LIBNAME _sasmax_ (_sasmax_ __genmac); * concatenate libref ;
  OPTIONS mstored sasstore=_sasmax_; * set search path for macro catalogs;
%END;
...

```

The libname `_sasmax_` is assigned by logically concatenating the physical SAS data libraries sequentially. Each library reference needs to be defined “read only” to avoid unintentional overwriting.

The SAS system option `MSTORED` searches for stored, compiled macros in the SAS data library referenced by the `SASMSTORE` option.

During the development of `USEGENMAX`, Accovion discovered that concatenating libraries defined as “read only” leads to the following warning message in the log file: “WARNING: The `_SASMAX_`.`SASMCR` catalog is opened for read only”. This warning message can be avoided by starting the definition of the concatenated libname with the current work directory. The current work directory does not need to be defined as “read only” and the warning message is avoided.

Notes in the log file after execution:

```
NOTE: Libref _SASMAX_ was successfully assigned as follows:
Levels:          4
Engine(1):       V8
Physical Name(1): /saswork/... ← Work directory
Engine(2):       V8
Physical Name(2): /SASMAX/generic/macro1/currentversion/cat
Engine(3):       V8
Physical Name(3): /SASMAX/generic/macro2/currentversion/cat ←
Engine(4):       V8
Physical Name(4): /SASMAX/generic/macro3/allversions/V1.0/cat

```

The diagram shows three boxes representing macro catalogues:

- Macro catalogue for MACRO1 / currentversion (arrow points to Physical Name(1))
- Macro catalogue for MACRO2 / currentversion (arrow points to Physical Name(3))
- Macro catalogue for MACRO3 / V1.0 (arrow points to Physical Name(4))

All generic macros introduced according to the procedure described above are now available for execution.

This way of embedding generic macros has proved to be a major enhancement. There is no longer the risk of unintentional changes to validated standard systems. The application of generic macros became easier and transparency, with respect to the version used, increased significantly.

Examples:

- ✓ SAS macro `NMAX` counts column totals for table outputs.
- ✓ SAS macro `CATMAX` creates frequency outputs of categorical variables for table outputs.

SAMPLE PROGRAMS

A sample program is any study or project-specific program that could be reused in another study with minor modifications. Sample programs serve as examples e.g. for the application of generic macros and need thus to be copied, adapted and revalidated as appropriate for the current project.

There are various kinds of sample programs that are worth publishing. Therefore, subcategories like derived datasets, safety tables/listings/graphs, appendices, etc. are introduced for structuring.

Each sample program submitted to `SASMAX` has to have the program header completed to enable easy understanding. This standardized program header contains a short description of the intended purpose of the program, information on required input and produced output, interaction with other programs, macro parameters if

PhUSE 2007

appropriate, etc. It is highly recommended to submit output examples together with a sample program. No customer or project-specific information must be published in SASMAX. All programs and output examples shared need to be blinded accordingly.

Examples:

- ✓ SAS program COND_DM creates a demographic table following Accovion's sample table layout.
- ✓ SAS program SAE_COMPARE creates a listing supporting serious adverse event reconciliation.

TOOLS

Tools handle recurring programming issues. They are normally far less complex than generic macros and don't require the same level of validation. A short description has to be provided for each tool to enable easy reapplication by other programmers. Tools are copied into the specific project for reuse as appropriate. Revalidation has to be considered.

Examples:

- ✓ SAS macros SAVEOPT, RESTOPT save SAS Options at the beginning of a macro and restore them after the end of the macro.
- ✓ SAS macro GET_LMDT identifies the last day of a given month.

ACCESS RIGHTS

Read access to SASMAX is granted to all SAS users but only a few administrators have full access to the SASMAX directories.

ADMINISTRATION AND MAINTANANCE

SASMAX is a living component within the SAS programming environment. Therefore, maintenance and supervision need to be done by a few administrators. The SASMAX administrators are responsible for the maintenance of the program library SASMAX and all associated elements (e.g. Web interface, SASMAX User Manual and Administrator Manual). They are not only liable for technical maintenance, but also for the supervision of the contents of SASMAX. SASMAX administrators need to check if new entries to SASMAX are

- ✓ Reasonable and worth publishing
- ✓ Not yet available
- ✓ Accurate, i.e. adhere to Accovion's Good Programming Practice and Accovion's Validation SOP
- ✓ Complete, i.e. all required documentation is available

In case of any ambiguity or missing documents, administrators will contact the author of the new entry. As soon as all requirements are fulfilled, the administrators include and activate the new entry in SASMAX. This means setting up new directories according to the SASMAX standards, copying the SAS program and the documentation into the correct location, creating links as appropriate and compiling macro catalogs for new or updated generic macros.

WEB-BASED INTERFACE

One major question while setting up the concept of the SAS program repository was how the contents could easily be made available for all SAS programmers. Programmers need to have the possibility to get a quick overview of what is available and they need to have the chance to submit their own code without major effort., i.e. the design has to be as user-friendly as possible. Initially, a simple table of contents was considered, e.g. an Excel file. But while advancing the concept of SASMAX, it became obvious that acceptance is basically dependent of the ease of use. Therefore, a user-friendly Web-based interface was designed offering several advantages

- ✓ User-friendly design
- ✓ Menu-driven
- ✓ Self-explanatory
- ✓ Point-and-click
- ✓ Filter by category and subcategory
- ✓ Search by string
- ✓ Search by date
- ✓ Link to source code and documentation
- ✓ Ease of submitting new entries via adjusting entry masks

Maintenance administrators as well as regular users need a comfortable way of accessing SASMAX. The following sections show the architecture of the Web-based interface, its functionalities with respect to retrieval and publishing of SAS code as well as for maintenance and administration.

PhUSE 2007

The screenshot below shows the appearance of the SASMAX interface:

The screenshot shows the SASMAX interface with the following search filters:

- Show category: all
- Show subcategory: all
- From: 10/2005
- Until: 09/2007

Buttons: Start Search, New entry, Show all categories, Search: [input] Go, search in all categories (checkbox).

Program	Responsible	Docs	Category	Subcategory
get_lmdt.sas	Steffen Müller		SAS Tools	ACC

Macro to create date with last day of a month when only month and year available.

Administration

RETRIEVAL OF SAS CODE

The SASMAX Web interface is available through the Accovion Intranet and can be accessed by all SAS programmers.

For each SAS program published in SASMAX, a minimum of additional information is presented on the main screen:

- ✓ Program name
- ✓ Corresponding main and subcategory
- ✓ Responsible programmer
- ✓ Short description of the program
- ✓ Saving date (by moving the pointer over the program name)
- ✓ For generic macros only: latest version
- ✓ For sample programs only: dependencies, specifics
- ✓ Link to program's source
- ✓ Link to additional documentation like user manuals, output examples, etc.

Without pre-election, all SAS programs published in SASMAX will appear. Filtering by main and subcategory limits the list accordingly. Users can also decide to run a search by publication date or to initiate a keyword search.

After the respective SAS program or macro is identified, it can be used as described in the design concept above.

The screenshot below shows filtering by main and subcategory:

The screenshot shows the SASMAX interface with the following search filters:

- Show category: Sample Programs
- Show subcategory: DDS
- From: DDS
- Until: TLG-Study Conduct

Buttons: Start Search, Show all registered programs, Show new entries, Search: [input] Go, search in all categories (checkbox), Help

Program	Responsible	Docs	Category	Subcategory
---------	-------------	------	----------	-------------

PhUSE 2007

SUBMITTING NEW SAS CODE

SAS programmers are encouraged to submit their own code for publishing in SASMAX. Submitting a SAS program or macro is supported by the Web interface. The button „New entry“ guides the user to the dynamic entry mask. Depending on the type of program, i.e. generic, sample or tool, all input fields that need to be completed appear. For each category the name of the program, the name of the programmer and a short description are required. A browser supports the upload of the SAS source and the corresponding documentation. The whole input will only be accepted if all required information is complete. As soon as the new entry is submitted, an email message including the submitted documents is sent to the administrators of SASMAX.

The example below shows how a sample program can be submitted to SASMAX:

Programdetails	
Category:	Sample Programs
Subcategory:	TLG-Efficacy
Programname:	testprogram.sas
Responsible(s):	Steffen Müller <small>seperate names with semicolons</small>

Programdescriptions (Sample Programs)	
Program creates a:	dataset
Outputtitle:	 e.g.: "Subject accounting"
Dependencies:	Uses CATMAX Macro e.g.: "Uses TABMAX Macro"
Specifics:	 e.g.: "crossover study" or "phase I"

Upload:	
Documents:	<input type="text"/> Browse...
<small>(Code, Output, User-Manual, ...) (only *.sas / *.zip)</small>	

Submit

PUBLISHING OF SAS CODE

After the administrator is informed about a new entry to SASMAX, he or she checks the entry for accuracy. If there are no objections, the administrator sets up new directories as appropriate, saves the source code and the associated documents and compiles macro catalogs for generic macros. A password-protected area in the Web-based interface, the so-called administration area, offers supporting functions for the administrators.

Administrators

- ✓ Release new entries
- ✓ Lock or unlock entries
- ✓ Delete entries
- ✓ Alter entries via entry masks as provided for the submission of new SAS code
- ✓ Enable links

PhUSE 2007

The example below shows the functionality of the administrator interface:

Show category:	<input type="text" value="all"/>	Show all registered programs Show new entries	Search:	<input type="text"/>	Go
Show subcategory:	<input type="text" value="all"/>		<input type="checkbox"/> search in all categories		
From:	<input type="text" value="10/2005"/>				
Until:	<input type="text" value="09/2007"/>				
<input type="button" value="Start Search"/>					
Program	Responsible	Docs	Category	Subcategory	
get_lmdt.sas	Steffen Müller		SAS Tools	ACC	
Macro to create date with last day of a month when only month and year available.					

Additionally administrators are authorized to add new administrator accounts. Nevertheless the workload for the administrators is acceptable and additionally supported by an administrator manual.

AWARENESS, TRAINING AND COOPERATION

One of the most challenging questions during the setup of SASMAX was how the program library could be promoted for daily use. SASMAX can only be a benefit if SAS programmers are convinced that this way of sharing SAS code is reasonable and advantageous. Programmers should be encouraged to publish their own code when the code could benefit others, or to reuse without prejudice code written by other programmers. During the initiation phase it became evident that regular user updates of newly published items to SASMAX are needed. Accovion programmers reached the consensus to inform each other periodically in the monthly routine. This approach has proven to be favorable, because SAS programmers become attentive to new developments and the programmer of the new source has the opportunity to explain the program and answer questions.

FUTURE ENHANCEMENTS

Neither the content nor the design of SASMAX is static. New challenges such as compatibility between different operating systems will lead to upgrades of SASMAX. Generic macros will not only be maintained and validated for several SAS versions, but also for different operating systems, i.e. Unix and Windows platforms.

Introducing standardized processes with regard to SASMAX during project initiation and project close out are worth considering. In any case during project setup, one of the key project players should review SASMAX for SAS programs that could potentially be reused. The same holds true when a project is finalized: Programs written during the project should always be assessed for possible reuse and submitted to SASMAX as appropriate.

CONCLUSION

How SAS code is shared is mainly dependent on the company profile, the projects that need to be handled, the working environment and the SAS programming staff. The strategy presented in this paper is only one of various possibilities how reasonable reuse of SAS source can be implemented.

Accovion implemented a SAS source-sharing process that fit the needs of a CRO business and complied with the requirements introduced by different sponsors. Spontaneous switching of staff to an urgent project is facilitated through the availability of standardized SAS code. SASMAX supports the maintenance of generic macros through the software development life cycle. Awareness of the required accuracy when utilizing generic macros to assure transparency and to meet validation requirements increased noticeably.

More than 2 years of experience with Accovion's SAS program library SASMAX shows that programmers at both sites, regardless of the number of years with SAS programming experience in the pharmaceutical environment, participate in sharing and publishing SAS code. Each programmer's name can be found in SASMAX in the role of the publisher. It has also become popular to review other programmer's source because the team discovered that there are still features, tricks or algorithms they were not aware of even after several years of SAS experience. Thus, SASMAX makes an important contribution to the community of SAS programmers at Accovion.

REFERENCES

- SAS OnlineDoc
(<http://support.sas.com>)

ACKNOWLEDGMENTS

I would like to thank Gaye Lucas, Eddie Arnold and Steffen Müller from Accovion for their valuable help and input to this paper, Monika Kawohl, Steffi Sturm, Michael Ludwig, Dirk Spruck and Paul Gensler from Accovion for promoting the SASMAX project initiation and implementation, Konstanze Schmidt from Accovion for design and implementation of the Web-based interface and all SAS programmers at Accovion for continuous enhancements to SASMAX.

CONTACT INFORMATION

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

Beate Hientzsch
Accovion GmbH
Helfmann-Park 10
65760 Eschborn, Germany
Work Phone: +49 6196 7709-406
Fax: +49 6196 7709-115
Email: Beate.Hientzsch@accovion.com
Web: www.accovion.com

Brand and product names are trademarks of their respective companies.