

Regulatory Compliant Validation and Deployment of SAS[®] Programs with the *Colibri* Server

Andreas Mangold, HMS Analytical Software GmbH

ABSTRACT

This paper shows the concepts and demonstrates the use of Colibri, which is a SAS[®] based server for the regulatory compliant management of data analysis programs and analysis results. Colibri is designed for speeding up the validation process by providing a framework for every activity of developers, testers and software administrators in a complex, multi-client environment in large organizations. The framework extends to an end user environment compliant to 21 CFR Part 11, where users who have no programming knowledge apply the validated SAS programs to their data and where they manage and archive the results.

Colibri ("Code Libraries") supports the software lifecycle in the regulated industries with built in tools for change management and version control, configuration management, programming in the team, multi language support and web-deployment without additional coding.

The testing environment reduces the effort for the specification, execution and documentation of tests.

Colibri is completely web based, so users of the validated SAS-programs can work with Colibri across sites and organizations. The Colibri administrator assigns access rights for programs and for analysis results. Audit trails and other traceability tools make pursuable the complete process from raw data and data preparation and cleaning to results, ready for submission.

After an introduction to the challenges in the validation of SAS programs and the management of analysis results, this paper shows the essential concepts of the Colibri solution.

INTRODUCTION

To ensure quality, companies in the industries for pharmaceutical and medical products need complex statistical and mathematical analysis procedures for the development, evaluation and production of their products. Examples are:

- Pharmaceutical: evaluation of clinical and preclinical studies,
- Diagnostics: calibration of test materials,
- All: quality control for the production process.

Analytical software is complex, has to be individually designed and programmed and always has to keep pace with the development of new products. Software engineering is needed.

Two additional factors make it hard to provide analytical software in time:

The American Food and Drug Administration (FDA) and comparable authorities for the admission of medical products in other countries place strong requirements both on the software development process and on the processes for data storage, data manipulation and data analysis. See chapter Regulatory Requirements.

Moreover, analytical software is used in different processes of a company, in different countries, sites and departments and even in partner organizations like clinics, laboratories or suppliers. Users in research work differently from users in evaluation or in production. Some users need simple, standardized ways of data analysis; others need to combine several complex individual steps of analysis. See chapter Multi-Client Requirements.

HMS Analytical Software GmbH faced those challenges in different project situations where we used SAS software for data analysis. The following two chapters show standard answers to the different challenges placed by regulatory and multi-client requirements. In order to make the processes of validation and application of SAS programs fast and easy, HMS developed the integrated web based Colibri platform which incorporates many of those standard answers. See chapters Colibri – Concepts and Colibri – how it works, which demonstrates parts of Colibri software by means of an example. Chapter Software Architecture briefly describes the internals of Colibri.

What Colibri cannot do: Colibri is not a replacement for SAS Enterprise Guide or SAS Display Manager; program logic will still have to be developed before programs can be integrated into Colibri. And: Colibri is not an environment for the management of validation plans and requirement and design documents but it can refer software modules to those documents.

REGULATORY REQUIREMENTS

When doing data analysis with SAS software, how do we live with the regulatory requirements? Following are some solutions suitable for SAS programmers. We have also built some of these solutions into Colibri.

Requirement	Possible solutions
Software validation	<ul style="list-style-type: none"> Apply a software development methodology, where user requirements, software design and testing are well documented and traceable to one another. Use existing document templates from the software development quality guidelines of your organization and adapt them to data analysis tasks. Do not forget to ask your quality commissary for approval. Break your SAS programs into modules which can be tested separately. Test bottom - up. Make enough tests for every item in the software design. Document tests and their association to design items.
Traceability of analysis results to program runs	<ul style="list-style-type: none"> Document parameters in SASLOG Verbose SASLOG (MPRINT etc.) Include reference to user in SASLOG and reports Include reference to raw data in SASLOG Save SASLOG with results
Traceability of analysis results to software used	<ul style="list-style-type: none"> Apply change and configuration control for SAS programs, use an revision control system Print version numbers on reports
Traceability of manual changes to raw data	<ul style="list-style-type: none"> Save audit trails with the changed data Make printouts of audit trails if you have no electronic signatures
Safe electronic records	<ul style="list-style-type: none"> Store data, logs and results in a safe place on the server Plan directory structure Apply checksums (PROC CHECKSUM)
Restricted access	<p>If other users run programs:</p> <ul style="list-style-type: none"> Define responsibility Ensure that users get training check authorization restrict access

MULTI-CLIENT REQUIREMENTS

As soon as you create SAS programs not only for a specific purpose, but your programs will be used by many users multiple times in defined processes, standardization and harmonization become important.

Requirement	Possible solutions
Data analysis for different tasks and different products for different departments	<ul style="list-style-type: none"> store programs on a central server and give selective access to programs have metadata around programs to document usage, prerequisites and result types organize reuse of software through modularization and documentation support multiple languages Provide an access control framework for programs and data Use thin client software to reduce installation
Data analysis for partner organizations (clinics, laboratories, suppliers)	<ul style="list-style-type: none"> Use web based user interfaces with https connection
Users in research work differently from users in evaluation or in production	<ul style="list-style-type: none"> Give templates to users performing highly standardized tasks Give others the possibility to combine several complex individual steps of analysis

COLIBRI – CONCEPTS

Colibri is a server based software system for validation and deployment of SAS software modules and for management of analysis runs and analysis results.

ROLES

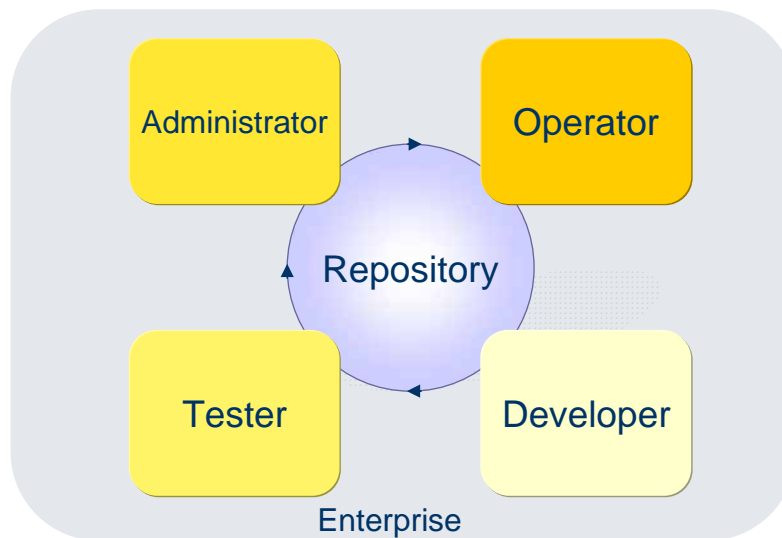


Figure 1: Roles in Colibri

Colibri assigns users to four different roles and makes available a web based user interface to each of them. Here is the workflow:

- The developer writes SAS programs using SAS Enterprise Guide or SAS Display Manager, does some testing and integrates the programs into Colibri where they are managed as “Colibri functions”. User interface for testers and operators is automatically created by the system.
- The tester uses the testing environment and executes all the necessary tests for each SAS program and for an ensemble of programs, documents the outcome, releases for use or sends back to developer.
- The administrator manages users, groups and grants access rights. No installation on personal computers is necessary.
- The operator uses a web based interface to run the SAS-Programs in an environment called “Colibri workspaces”, which can be permanently and securely stored on the server.

The process is kept together by means of a software repository which tracks all the Colibri objects and the associated metadata. Every user has to login so authorization and access rights can be checked.

COLIBRI FUNCTIONS

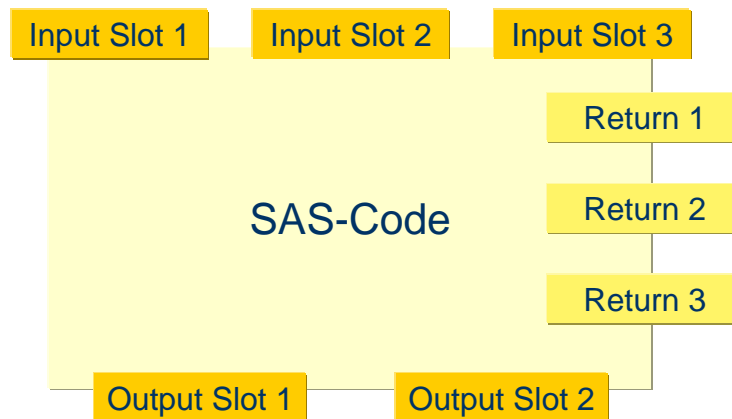


Figure 2: structure of a Colibri function

A Colibri function is a SAS program, usually a SAS macro. When the developer integrates a program into Colibri, he assigns it a name and a description and a set of named parameters, called input slots, and a set of named results, called output slots. Data types are assigned to the slots. He also assigns a name and a description to every possible return condition of the program. This makes it possible to inform the operator of incorrect usage or special error conditions when he runs the function later.

WORKSPACES

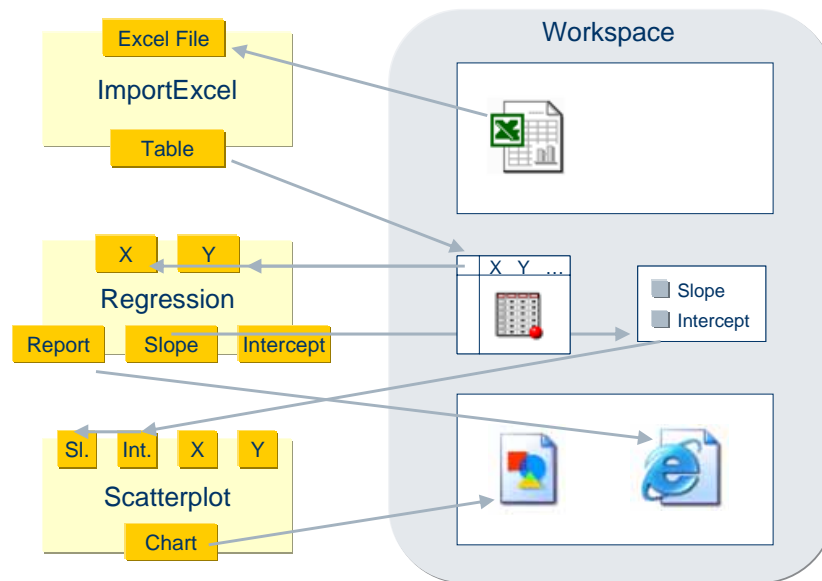


Figure 3: Function calls and data elements in a workspace

A Workspace is much like what accumulates during a normal SAS session, but it is always reproducible and it can be stored permanently and safely. Figure 3 shows a simple example: three Colibri functions can be used together to import data, estimate regression parameters and to draw a scatter plot with a regression line. The raw data (can of course also be read from a database), the call sequence, the parameters, the intermediate results (table and regression properties) and the reports are all stored in the workspace. The workspace can be reopened later and extended or it can be made read only when finalized.

COLIBRI – HOW IT WORKS

DEVELOPER

SAS programs can easily be integrated into Colibri. The programmer uses a set of standard macros to assign input slots, output slots and return codes (see Figure 4). Upon integration into Colibri, these metadata are stored in the repository and a standard parameter page is being built.

Some more concepts structure and document the development process:

- Colibri functions are structured into packages, and packages assemble to applications. The operators work with applications. The developers can organize software reuse with packages.
- The integrated revision control system provides check-in check-out functionality and thus facilitates teamwork. It also ensures that all versions stay documented.
- As soon as the developer releases his Colibri function for test, he can no longer change it unless the tester assigns it back to him when there is an error. Then he has to create a new build and fix it.
- The developer can change the layout of the parameter page by means of an interactive layout editor.

TESTER

The tester creates test scenarios in order to test the software on three levels:

- Unit tests: test a single Colibri function,
- Package tests: test the ensemble acting of Colibri functions in a package,
- Application tests: acceptance tests.

Test scenarios are basically workspaces with added comments for expected behavior, observed behavior and evaluation. After the software modules have been sufficiently tested, a test report can be printed so all the tests are properly documented. When all tests have been marked as “passed”, the software module is being released for use and can never be changed.

When SAS programs have to be changed, this is greatly simplified: a new build is created, all test scenarios are copied and retesting is effortless.

```

%cimInputSlot (X, System.001.NumVector, xvar, LabelVar=xlabel);
%cimInputSlot (Y, System.001.NumVector, yvar, SameGroupAs=X, LabelVar=ylabel);
%cimInputSlot (Mit_Interceptor, Int0N, int);
%cimInputSlot (Signifikanzniveau, SNiveau, p);

/* ... */
ODS HTML FILE="%cim_saswork_path/report.html" STYLE=EGDefault;
TITLE "Regression [%ymame] = a + [%xname] * b";
FOOTNOTE1 justify=left height=1 "X: [%xname] %xlabel";
FOOTNOTE2 justify=left height=1 "Y: [%ymame] %ylabel";

/* Regression ausführen */
%PROC REG DATA=%dataset OUTEST=est TABLEOUT;
VAR %yvar &xvar;
N: MODEL %yvar = %xvar %IF NOT %int %THEN / NOINT;;
RUN;
QUIT;
%IF %cimReturn (ErrorProcReg, %syserr NE 0, %cim_Error, %syserr)
  %THEN %GOTO EXIT;

ODS HTML CLOSE;
/* ... */

/* prüfe auf Signifikanz */
%LOCAL sign; %LET sign=%SysEValF(%pvalue <= %p);
%IF %cimReturn (NotSignificant, %sign, %cim_Warning, %syserr, %pvalue, %p)
  %THEN %GOTO EXIT;

%IF %cimReturn (NormalExit, 1, %cim_OK, 0)
  %THEN %GOTO EXIT;

%EXIT;
ODS HTML CLOSE;

%cimOutputSlot (Predicted, System.001.NumVector, output.predicted, SameGroupAs=X);
%cimOutputSlot (Residual, System.001.NumVector, output.residual, SameGroupAs=X);
%cimOutputSlot (Report, System.001.HTMLFile, report.html);
%cimOutputSlot (Intercept, Icept, %intercept);
%cimOutputSlot (Slope, Slope, %slope, SameGroupAs=Intercept);
%cimOutputSlot (Equation, System.001.CharProperty, %formula, SameGroupAs=Intercept);
%cimOutputSlot (PValue, System.001.NumProperty, %pvalue, SameGroupAs=Intercept);

```

Figure 4: Source code for Colibri function with Colibri macros added¹

ADMINISTRATOR

The Colibri administrator can sit at his desk while doing his administration tasks. No installation on personal computers is necessary. Every user of Colibri uses according to his role one of four user interfaces based on standard web browsers.

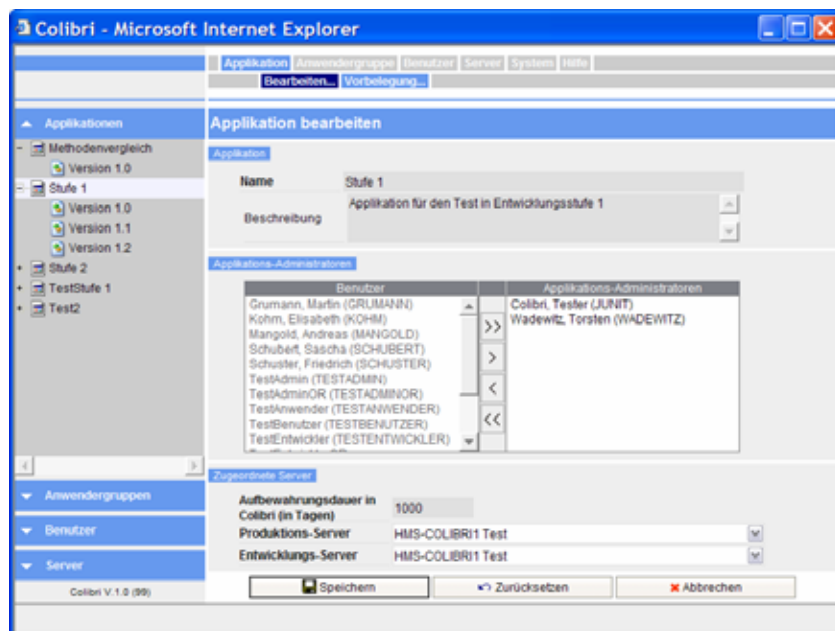


Figure 5: Assign users to groups

¹ The author wishes to apologize for not having translated some terms on screen shots. Colibri comes with full Multilanguage support. Demonstrations on PhUSE will be completely in English.

The administrator manages users and groups and assigns users to roles and groups to applications. As soon as a user with role operator is associated with an application, he can create workspaces and use the Colibri functions (SAS programs) contained in the packages of the application.

OPERATOR

Colibri Operators are the consumers of data analysis via Colibri functions. They create new or open existing workspaces (see Figure 3), upload data or select data from a database, apply manual data correction (with audit trail), apply Colibri functions (see Figure 6), inspect results (see Figure 7) and store the workspace in a workspace library. Results can be printed and downloaded. Functions and data in workspaces can be rearranged, but the traceability of results to data and functions is always maintained.

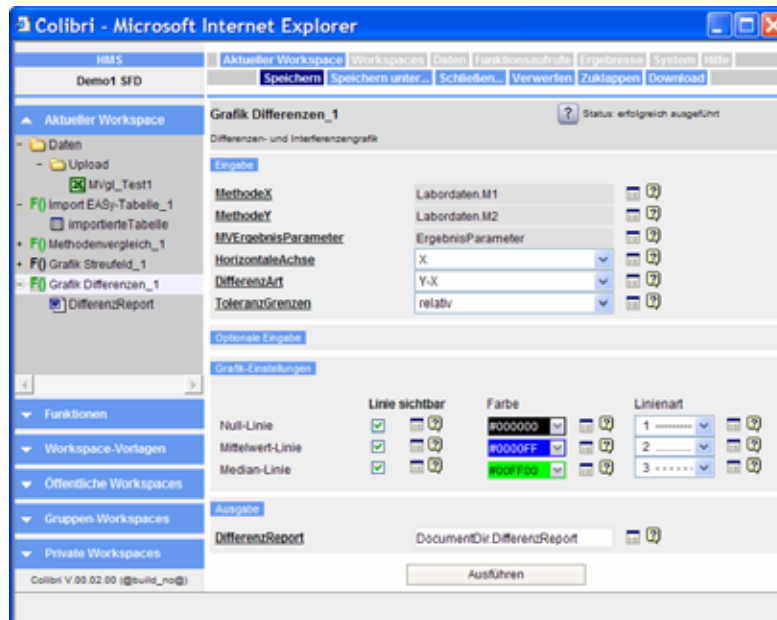


Figure 6: workspace contents on the left side, parameter page on the right (layout arranged manually)

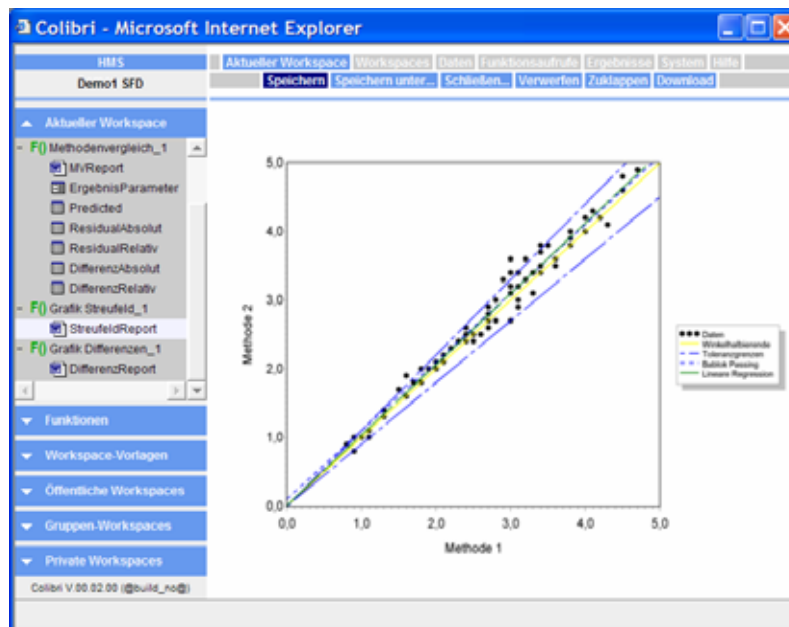


Figure 7: workspace contents on the left side, results on the right

Operators, like the other user roles, need only a web browser and optionally PDF- and XLS-Plugins in case they want to view those result types from within Colibri.

A workspace browser helps to find archived workspaces which can be opened and changed again as long as they are not frozen.

SOFTWARE ARCHITECTURE

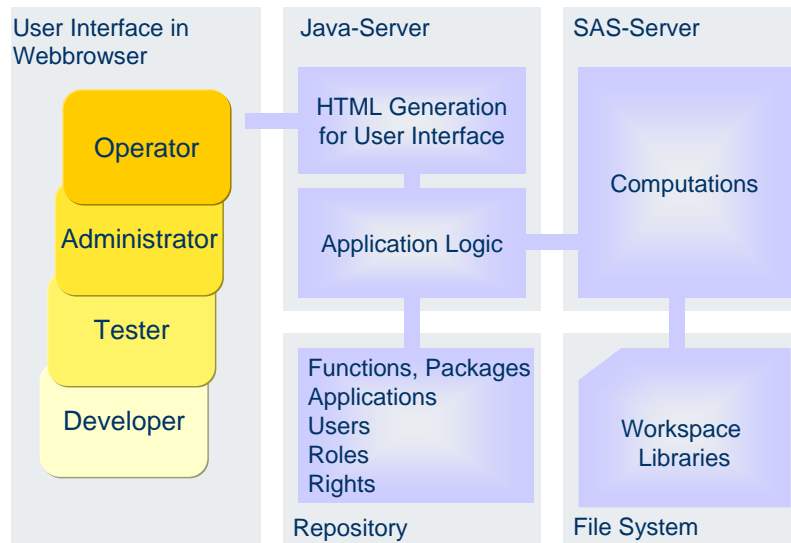


Figure 8: software architecture of Colibri framework

Colibri consists of the following components:

- Standard web browser (popular browsers supported) for the four user interfaces,
- Application logic and user interface generation on java container (e.g. Tomcat, IBM Websphere),
- SAS9 workspace server,
- Repository stored in a relational database (e.g. Oracle),
- Workspace libraries stored on the file system.

CONCLUSION

Colibri enables organizations having complex data analysis tasks to meet the regulations of the authorities regarding software development and data storage and to organize analysis software in a multi-client environment.

Colibri takes much work from the developer: he can concentrate on the realization of the specification and does not have to deal with aspects like user interface, access control or storage of results.

The testing environment reduces the effort for specification, execution and documentation of systematic tests.

Colibri alleviates the work of the administrator. Web based deployment of SAS applications is easy with Colibri.

Operators can work with their workspaces from every location worldwide over the company's intranet or the internet.

Colibri has been developed by and is sold by HMS Analytical Software GmbH, Heidelberg, Germany on the basis of the SAS® software.

CONTACT INFORMATION

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

Andreas Mangold
HMS Analytical Software GmbH
Rohrbacherstr. 26
69115 Heidelberg
Work Phone: +49 (6221) 6051 0
Fax: +49 (6221) 6051 99
Email: andreas.mangold@hms-heidelberg.de
Web: <http://www.hms-heidelberg.de/colibri/english>

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.