

Spot the Difference

Anja Feuerbacher, F. Hoffmann-La Roche Ltd., Basel, Switzerland
Beate Hientzsch, Accovion GmbH, Eschborn, Germany
Gabi Lückel, Accovion GmbH, Eschborn, Germany

ABSTRACT

The delivery of high quality statistical analyses within stringent time limits is a challenge for any statistical programming group. Meeting delivery dates depends on factors such as the availability and quality of data, finalization of the statistical analysis plan (SAP) and table shells, ad hoc requests, last minute changes etc. Above all you still need to provide high quality even against high-pressure deadlines.

Thorough validation and change control of SAS[®] programs and outputs is crucial, but can be time-consuming. Depending on the data volume, as well as the number and complexity of outputs, the repeated execution of all programs can take hours, not to mention the re-validation of hundreds of outputs. At this stage, an efficient process is required to minimize time and personnel expenditure and to increase accuracy in output validation.

This paper demonstrates how

- o archiving of previous data sets and TLGs (**T**ables, **L**istings, **G**raphs)
- o execution of SAS-programs for generation of ADS (**A**nalysis **D**ata **S**ets) and TLGs
- o programmatic re-validation and change control of outputs

can be combined in a single macro call.

INTRODUCTION

For the analysis of a clinical study or the preparation of a meta analysis typically hundreds of outputs, i.e. analysis data sets, tables, listings and graphs need to be produced. Programming and validation always has to be started long before database closure because results need to be available and reliable within a few days after the database has been locked.

To facilitate the process of re-running, some kind of SAS program or script is usually applied to enable re-run with just one command. The Macro RMC (**R**erun programs and **M**ark **C**hanges) presented in this paper combines the automatic re-run of programs with the possibility of saving previous outputs and making comparisons against these with the updated versions. RMC displays an overview of which files have been changed and shows all changes in detail. At a late stage of a project when all data sets and outputs have already been validated and change requests arise, the RMC macro is particularly useful in supporting change control. All changes can easily be reviewed and any unintended changes will be detected. At the same time, it can be shown that no changes occurred if RMC displays that no discrepancies were found.

SYSTEM ENVIRONMENT

The RMC macro was developed at Accovion using SAS on a UNIX operating system and makes use of the interface to the operating system. Hence, the macro combines SAS procedures and data steps with UNIX-specific commands and shell scripts. The UNIX specific elements could be adapted to respective commands for other operating systems, e.g. to make the RMC macro compatible for use on a Windows[®] platform.

PhUSE 2008

Examples for applied *UNIX* commands:

- o `sas test.sas`
executes the SAS program test.sas and creates test.lst and test.log. This command can be combined for the execution of several programs to automatically run all programs in a defined sequence.
- o `grep <options> test.log "XYZ"`
searches for the string "XYZ" within the file test.log and can be used to find keywords like ERROR, WARNING, etc. in SAS log files.
- o `diff <options> test1.lst test2.lst`
compares the files "test1.lst" and "test2.lst" and displays all unequal lines .

The UNIX commands can be run from SAS using an "x" command either alone or in combination with call execute or filename statements. Hence, the usage of the interface to the operating system in conjunction with SAS procedures and data steps allows for a high flexibility in programming.

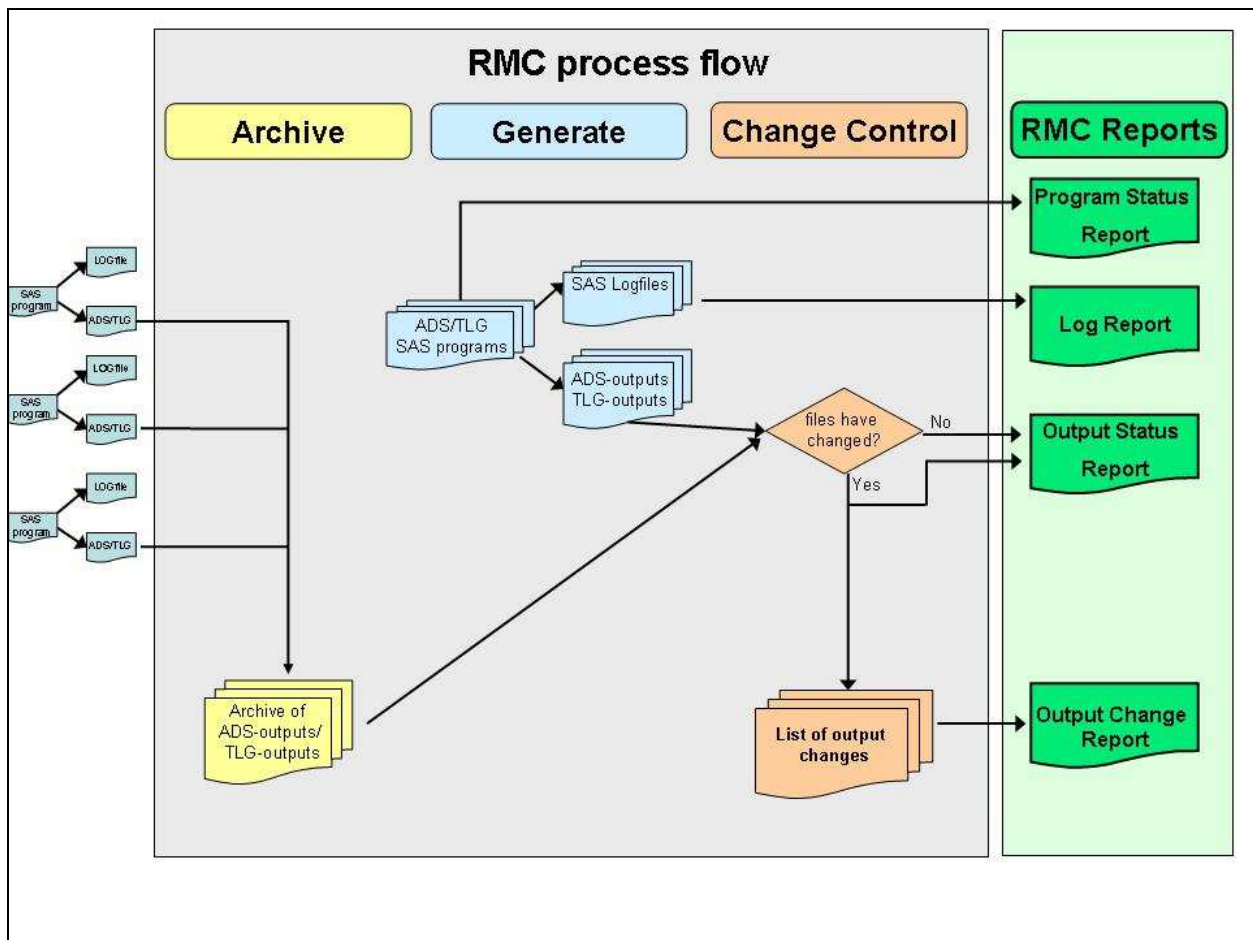
DETAILED DESCRIPTION OF FUNCTIONALITY

The RMC macro is composed of three independent modules

- ARCHIVE outputs: save previous version of ADS / TLGs
- GENERATE outputs: execute SAS programs in batch mode
- CHANGE CONTROL: compare outputs in support of re-validation

These modules provide short and concise status reports that serve as action lists for programmers, showing that either corrective actions or verification are necessary or that no changes occurred. The corresponding reports can be kept as documentation, that everything was checked and approved.

The RMC modules can be used independently of each other but also in various combinations.



PhUSE 2008

RMC supports archiving, generation, and comparison of analysis data sets (ADS) as well as of tables, listings and to some extent for graphical outputs (TLGs). With very few exceptions all modules work in the same way for ADS and TLGs and are therefore described in general. The various options of the RMC macro provide a certain grade of flexibility and will be explained in the following sections for each module separately.

ARCHIVING

Parameter	Functionality	Values
SAVE_ADS	Save ADS by moving analysis data sets to an automatically created archive folder	YES/NO
SAVE_TLG	Save TLGs by moving all tables, listings and graphs to an automatically created archive folder	YES/NO

With the archiving functionality the RMC macro provides the possibility to save previously generated outputs, i.e. ADS/TLGs. Thus, a “previous version” is archived, against which new ADS/TLGs can be compared.

These archive folders are created automatically. A new name is assigned to each archive folder including the current system date and time. This approach allows for a secure and chronological archiving process because it minimizes the chance of unintentionally overwriting existing archives.

In this step all ADS/TLGs are moved (not copied) to the archive folder. This means a subsequent batch run of all programs can start in a “quasi-clean” environment in order to check whether the required set of programs can be executed without any prerequisites. For example,

- A data set with adverse events of special interest (AE_SP_INT) was provided by the customer as source for the ADAE data set. AE_SP_INT must not be changed and there is no program which creates this data set. In general, such a data set is stored together with source data or at least in a separate folder. If this file was accidentally stored together with the ADS data sets, AE_SP_INT is automatically moved to the archive folder during archiving, so the later execution of the ADAE program fails.
- Another example is the usage of a data set which was created by a program that is no longer executed. When starting the complete run in an unclean environment, it could happen that programs use the obsolete data without the programmer’s knowledge. Instead, the RMC macro allows to quickly determine the programs which fail because of a missing data set.

If the programs also need to be executable at the customer’s site, checking the error-free run in a clean environment is essential.

GENERATION OF ANALYSIS DATA SETS / TABLES, LISTINGS, GRAPHS

Parameter	Functionality	Values
RUN_ADS	Execute programs to generate analysis data sets.	YES / NO
RUN_TLG	Execute programs to generate TLGs.	YES / NO

Another functionality of the RMC macro is the controlled execution of SAS programs in batch mode. This facilitates the transparency of a study analysis: you can quickly determine which programs are used and in which order they need to run. The list of programs is specified in the respective step of the calling programs as shown below.

This module can already be applied during the early development phase. If, for example, an update of raw data is imported overnight, you can define a subsequent job for the batch run of available ADS/TLG programs to integrate the new data in the analysis data sets and update the outputs. Thus, you can update all ADS/TLGs “at the push of a button”, and start with the most current version in the morning and verify that all programs work as expected.

Furthermore, the specification of the correct sequence of programs for proper execution creates a transparent documentation of the program logic for a study and enables easy re-run of the whole set of programs. This is especially important if programmers act on the assumption that certain ADS are already available and can serve as input for further ADS. Hence, the list of programs always displays the current dependencies. If a study programmer drops out, another programmer can fill in and work efficiently without potentially threatening time lines by struggling with small details (e.g. program A must run before program B). The following example shows an extract of such a program:

PhUSE 2008

List of programs to create all ADS:

```
*-----;
* ADS programs ;
*-----;
** Note, that programs have a defined sequence to be called **;

** NOTE: ADSL must run as first program **;
%runpg(prg= der/adsl, sas= YES);
%runpg(prg= der/adcm, sas= YES);
%runpg(prg= der/addm, sas= YES);
...
%runpg(prg= der/admh, sas= YES);
```

List of programs to create all TLGs:

```
*-----;
* TLG programs ;
*-----;
** DEMOGRAPHY **;
%runpg(prg= rep/cond/dm01, sas= YES);
%runpg(prg= rep/cond/dm01b, sas= NO);
%runpg(prg= rep/cond/dm02, sas= YES);
...
** ADVERSE EVENTS **;
%runpg(prg= rep/saft/aes01, sas= YES);
%runpg(prg= rep/saft/aes02, sas= YES);
** AE plots **;
%runpg(prg= rep/saft/aes312, sas= YES);
...
```

The macro RUNPG, which is used inside the RMC macro runs programs in batch mode and automatically creates an aggregated Log Report, i.e. a Unix script searches for pre-specified key words in the SAS log files and all lines with critical or potentially critical key words like "ERROR", "WARNING", "duplicate", "missing" and other special "NOTE"(s), or user defined INFO messages are displayed. The following example demonstrates how this provides a concise spot to programs which must be checked:

Log Report

```
*****/.../der/adsl.log*****
----- ERROR -----
----- FATAL -----
----- WARNING -----
----- INFO: (user defined messages) -----

*****/.../der/addm.log*****
----- ERROR -----
521: ERROR: Ambiguous reference, column weight is in more than one table.
----- FATAL -----
----- WARNING -----
725:WARNING: The query as specified involves ordering by an item that doesn't appear in its
SELECT clause since you are ordering the output of a SELECT
----- NOTE: Variable ... is uninitialized -----
1410:NOTE: Variable age_group is uninitialized.

----- NOTE: Missing values were generated ... -----
624:NOTE: Missing values were generated as a result of performing an operation on missing
values.
----- INFO: (user defined messages) -----
125: INFO: INFO: The warning 'The query as specified involves ordering by an item ...' has
been checked and is harmless - don't worry.
2556:INFO: Age does not meet the inclusion criteria: subjid=0002489 age=17
```

PhUSE 2008

The log file for *adsl.sas* does not show any problems, but the log file of *addm.sas* identifies several messages, where corrective actions or verification are necessary.

During the development life cycle, SAS programs are usually separated in different areas depending on their development / validation status and to enable version control. In our case these is the development environment, the validation environment and the productive environment. The RUNPG macro autonomously finds and executes the specified programs in their actual environment and the RMC macro provides a status report displaying the information about the development status of each program:

Program Status Report

<u>Status</u>	<u>Program name</u>
Productive	rep/cond/dm01.sas
	rep/cond/dm02.sas
	...
Under development	rep/saft/aes01.sas
	rep/saft/aes312.sas
Under validation	rep/saft/aes02
Program missing	rep/eff/dm01b.sas

CHANGE CONTROL

Parameter	Functionality	Values
COMP_ADS	Compare analysis data sets.	YES / NO
CMPDIR_ADS	Specify a back-up directory for analysis data sets, which is used for comparison instead of the automatically created archive folder.	<directory>
COMP_TLG	Compare table and listing outputs.	YES / NO
CMPDIR_TLG	Specify a back-up directory for table and listing outputs, which is used for comparison instead of the automatically created archive folder.	<directory>
BLOCK	Specify a character sequence to compare a subset of table and listing files.	<subset>

The main focus and intention of the RMC macro is the programmatic support of re-validation and change control of output files, i.e. ADS/TLGs, by automatic comparison. Note that the RMC macro does not re-validate ADS/TLGs, but verifies if changes between old and new versions occur – it cannot tell if the changes are correct or not. However, if you do not expect changes and the comparison module of the RMC macro does not report any difference, all ADS/TLGs can be considered as re-validated. On the other hand, if any differences compared to a previous version are found, the RMC macro supports change control and the user only has to take care about explanation of changes and re-validation.

New ADS/TLGs can be compared against the “previous version”, which was automatically created in the archiving module of the RMC macro, or against versions in a user defined folder, e.g. test_study/Validated/... or test_study/backup_20080218/... .

RE-VALIDATION AND CHANGE CONTROL OF ANALYSIS DATA SETS

The comparison of ADS is done with SAS-based comparison techniques, in particular *SAS proc compare*. The RMC macro reads all single outputs of the compare procedure and extracts the relevant information, i.e. equal or not equal into one concise Output Status Report.

The following example shows a report presenting the number of variables in the old data set (OLDVAR), number of variables in the new data set (NEWVAR), number of variables in common (COMVAR) and, similarly, the number of observations in the old data set (OLDOBS), in the new data set (NEWOBS) and the number of common observations in both data sets (COMOBS). If the output of *SAS proc compare* does not detect any differences in those observations and variables that are available in both data sets and therefore displays the note “NOTE: No unequal values were found. All values compared are exactly equal”, the EQUAL column shows a “Y”. The CHANGE flag summarizes the comparison of number of observations, number of variables and comparison of data values and therefore enables the quick determination of unequal ADS.

PhUSE 2008

Output Status Report for ADS

Data Set Overview									
	FILE	OLDVAR	NEWVAR	COMVAR	OLDOBS	NEWOBS	COMOBS	EQUAL	CHANGE
1	ADAE	69	75	69	208	379	208	Y	*
2	ADEF	26	26	26	456	456	456	Y	
...									

It is especially important during the validation phase to use this module for risk minimization. With a complete run of all ADS programs, it enables you to quickly verify that changes to programs only have the desired effect and do not affect other ADS. Another example of this functionality is described in the Section “Further examples for efficient usage of the RMC macro” below.

When comparing ADS, the RMC macro does not allow for ID variables to be used in the compare procedure. The status output can only give an idea that something has changed. Further exploration of differing data sets must be done outside the macro.

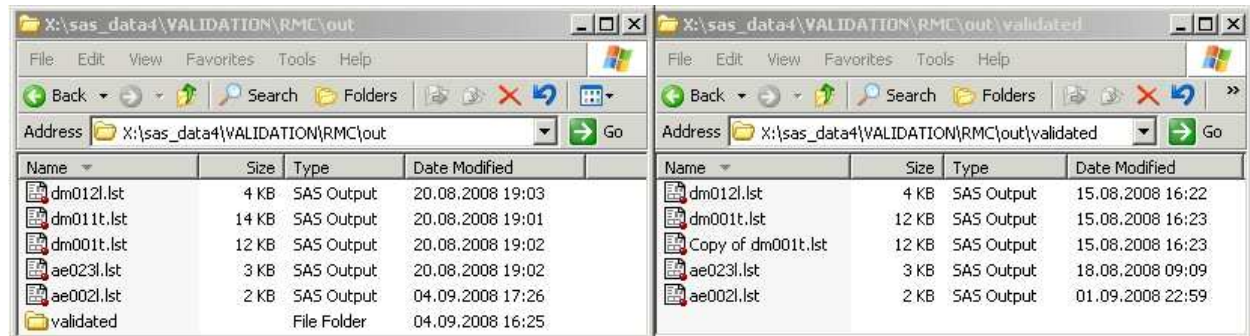
RE-VALIDATION AND CHANGE CONTROL OF TABLES AND LISTINGS

The comparison of tables and listings shows an overview of their existence and status first. Graphs cannot be compared in their usual .eps or .cgm format. A workaround is to store the underlying data as data sets or print the data to a table or listing. The old and new versions of these data sets or tables/listings can then be compared by using the change control module of the RMC macro.

The Output Status Report of tables and listings comprises a list of all files where no differences were found as well as a list of new, missing and changed files.

	A file is flagged as...		
	“new”	“missing”	“equal/unequal”
File found in the archive folder?	no	yes	yes
File found in the output folder?	yes	no	yes

The following screenshot shows the contents of an output folder and a user-defined archive folder named “validated”:



The application of the RMC comparison module for TLGs produces an Output Status Report showing first equal files and then the list of changed, new and missing files:

PhUSE 2008

Output Status Report for TLGs

Comparison of outputs against outputs in \$SAS.../out/validated

File Overview - Equal Files

Filename	Status	Equal
dm001t		Y
ae023l		Y

File Overview - Changed, New, and Missing Files

Filename	Status	Equal
Copy of dm001t	MISSING	
dm011t	NEW	
dm012l		N
ae002l		N

The following screenshot shows the old and the new version of the listing ae002l.lst, which are reported as unequal. Spot the difference!

ae002l.lst - WordPad

Listing of all adverse events (includes pre-, on-, and post-treatment) - Safety population

TREATMENT: Placebo

Patient	Age/ Gender/ Race	Adverse event: Preferred term name/ Reported term	Period[b]/ Last day on study med	Onset: Date/ Day [c]	Date of resolution/ Duration (days)	Rel.to study med [d]	Inten- sity [e]	Study med [f]/ Med [g]/ Other [h]	Outcome [i]
0007	8/F/ White	Mouth cyst/ ORAL CYST	ON/ 84	19MAR2007/ 8	19MAR2007/ 1	N	Mod	None/ Yes/	Rec, no seq No
0009	10/F/ White	Asthma/ ASTHMA EXACERBATION	ON/ 68	23APR2007/ 52	04MAY2007/ 13	N	Mild	PER/ Yes/	Rec, no seq No
		Cough/ COUGH	ON/ 84	18FEB2007/ 63	19FEB2007/ 2	N	Mild	None/ No/	Rec, no seq

Accovion GmbH: 1SEP2008 / 22:59 / ae002l.lst / 1ae02.sas Page 50 of 230

For Help, press F1

ae002l.lst - WordPad

Listing of all adverse events (includes pre-, on-, and post-treatment) - Safety population

TREATMENT: Placebo

Patient	Age/ Gender/ Race	Adverse event: Preferred term name/ Reported term	Period[b]/ Last day on study med	Onset: Date/ Day [c]	Date of resolution/ Duration (days)	Rel.to study med [d]	Inten- sity [e]	Study med [f]/ Med [g]/ Other [h]	Outcome [i]
0007	8/F/ White	Mouth cyst/ ORAL CYST	ON/ 84	19MAR2007/ 8	19MAR2007/ 1	N	Mod	None/ Yes/	Rec, no seq No
0009	10/F/ White	Asthma/ ASTHMA EXACERBATION	ON/ 68	28APR2007/ 52	04MAY2007/ 13	N	Mild	PER/ Yes/	Rec, no seq No
		Cough/ COUGH	ON/ 84	18FEB2007/ 63	19FEB2007/ 2	N	Mild	None/ No/	Rec, no seq

Accovion GmbH: 4SEP2008 / 17:26 / ae002l.lst / 1ae02.sas Page 50 of 230

For Help, press F1

Working with visual comparison of various outputs would be time-consuming and there is a potential high risk that differences would not be detected.

The RMC macro compares the two files by using the Unix command *diff*. The results are redirected into a file and read into a SAS data set for further processing. This particularly includes the deletion of unimportant lines, i.e. irrelevant changes, which is done with the macro `DELETE_LINES`.

PhUSE 2008

Typically, we are not interested in lines with the creation date of the files nor lines without differences. In our case, the creation date is expected in the same line as "Accovion" or the customer's name; therefore the RMC macro can drop those lines. It is helpful that the *diff* command does not display equal lines and marks differences with the symbols "<" and ">". That means all lines without these symbols are metadata from the DIFF command and can be deleted. If no lines remain for a pair of files, the outputs are considered as equal. Otherwise, lines starting with "<" or ">" indicate the changed, new or missing lines of the compared files. We optimize the readability of the Output Change Report by replacing all leading "<" symbols by "O" (line in old file), and all leading ">" symbols by "N" (line in new file). The easiness of post-processing the *diff* command's output supports a well arranged display of the Output Change Report.

```

%MACRO DELETE_LINES;
  if index(compress(upcase(diff)), "ACCOVION") gt 0 then delete;
  if index(compress(diff), "&company") gt 0 then delete;
  ...
  if substr(diff,1,1) not in ("<",">") then delete;
%MEND DELETE_LINES

```

The following Output Change Report shows the details of differences identified by the RMC macro after the comparison of the two versions of listing ae002l.lst shown above.

Output Change Report

Comparison of outputs against outputs in /sas_data4/VALIDATION/RMC/out/validated											
<u>Changed Files - Details</u>											
----- Filename=ae002l -----											
<u>Old/</u>											
<u>New</u>	<u>Differences</u>										
O	0009	10/F/	Asthma/	ON/	23APR2007/	04MAY2007/	N	Mild	PER/	Rec,	
N	0009	10/F/	Asthma/	ON/	28APR2007/	04MAY2007/	N	Mild	PER/	Rec,	

FURTHER EXAMPLES FOR EFFICIENT USAGE OF THE RMC MACRO

The following examples demonstrate the efficient usage of the RMC macro. It is impossible to re-validate hundreds of ADS and TLGs again and again. The RMC macro supports avoiding the re-validation by creating overview reports. Only relevant details of changes are listed. Equal ADS/TLGs are displayed in a very short and clear way, and if they had already been validated, it is safe to consider them as re-validated. Programmers can then concentrate on the identified and presented changes and verify whether they meet the expectations.

Example 1

ADS programming for the study is nearly finished. Now the statistician wants the flag indicating if a patient belongs to the safety population added to all ADS. This information can be retrieved from the ADDM data set. In this example the RMC macro helps you to find two unexpected effects: The ADCM data set did not change and was obviously not extended by the new variable (OLDVAR=NEWVAR and CHANGE is not flagged (a)) and someone altered the ADDM data set (CHANGE is flagged (b)) although it did not need to be changed, because there is an additional variable (NEWOLBS=OLDOBS+1). All other data sets were updated as expected (NEWVAR=OLDVAR+1). The number of observations and the existing variables did not change (OLDOBS=NEWOBS, EQUAL=Y).

Output Status Report

<u>Data Set Overview</u>									
	<u>FILE</u>	<u>OLDVAR</u>	<u>NEWVAR</u>	<u>COMVAR</u>	<u>OLDOBS</u>	<u>NEWOBS</u>	<u>COMOBS</u>	<u>EQUAL</u>	<u>CHANGE</u>
1	ADAE	69	70	69	208	208	208	Y	*
2	ADCM	36	36	36	307	307	307	Y	← (a)
4	ADDM	69	70	69	484	484	484	Y	* ← (b)
...									
18	ADEX	47	48	47	2048	2048	2048	Y	*
19	ADMH	36	37	36	430	430	430	Y	*

PhUSE 2008

Example 2

Data set ADDM was already validated, but to be compatible with other studies, the algorithm of age calculation was changed and adjusted across studies. Change control of the old and modified data set with the RMC macro came up with the following Output Status Report: Neither the variables nor the observations changed, but values changed (EQUAL=N).

Output Status Report

Data Set Overview									
FILE	OLDVAR	NEWVAR	COMVAR	OLDOBS	NEWOBS	COMOBS	EQUAL	CHANGE	
1	ADDM	69	69	69	484	484	484	N	*

Output Change Report

```

Comparison of data sets against data sets in /sas_data4/VALIDATION/RMC/dds

The COMPARE Procedure
Comparison of OLDDDS.ADDM with DDS.ADDM
(Method=EXACT)

Data Set Summary
Data set          Created          Modified  NVar   NObs
OLDDDS.ADDM      11SEP08:13:20:51 11SEP08:13:20:51   20   440
DDS.ADDM         11SEP08:13:20:51 11SEP08:13:20:51   20   440
...
Number of Variables in Common: 20.
...
Number of Observations in Common: 440.
...
Number of Variables Compared with All Observations Equal: 19.
Number of Variables Compared with Some Observations Unequal: 1.
Total Number of Values which Compare Unequal: 421.
...
Variables with Unequal Values
Variable  Type  Len  Ndif  MaxDif
AGE       NUM   8    421   0.695

Value Comparison Results for Variables

Obs      ||      Base      Compare
          ||      AGE       AGE       Diff.     % Diff
-----  ||-----
          ||
          ||
          1 ||      58.0000   57.7522   -0.2478   -0.4272
...

```

PhUSE 2008

Example 3a

The program labs01t.sas creates one summary table for each of the three analysis populations and two subgroups for each of the 50 laboratory parameters of a study. All tables have already been validated. The slightest change to this program would lead to a large amount of time consuming work: All outputs are re-produced, that means the file date is newer than the validation date. All 300 tables would need to be validated again! The usage of the comparison module of the RMC macro reduces the work load to the necessary minimum.

The label "Albumin (mg/dL)" was corrected to "Albumin (g/dL)". The RMC reports show that only this label was changed:

Output Status Report

<u>File Overview - Equal Files</u>		
	<u>File</u>	
<u>Filename</u>	<u>Status</u>	<u>Equal</u>
labs01t_ALP1I		Y
labs01t_ALP1P		Y
...		

<u>File Overview - Changed, New, and Missing Files</u>		
	<u>File</u>	
<u>Filename</u>	<u>Status</u>	<u>Equal</u>
labs01t_ALB1I		N
labs01t_ALB1P		N

Output Change Report

----- Filename = labs01t_ALB1I -----	
<u>Old/</u>	
<u>New</u>	<u>Differences</u>
O	Albumin (mg/dL)
N	Albumin (g/dL)
...	(same details for the other files)

Example 3b

By accident the label of Alkaline phosphatase was changed instead of Albumin. Supported by the report of the RMC macro you can immediately see that the change has unexpected effects:

Output Status Report

<u>File Overview - Equal Files</u>		
	<u>File</u>	
<u>Filename</u>	<u>Status</u>	<u>Equal</u>
labs01t_ALB1I		Y
labs01t_ALB1P		Y
...		

<u>File Overview - Changed, New, and Missing Files</u>		
	<u>File</u>	
<u>Filename</u>	<u>Status</u>	<u>Equal</u>
labs01t_ALP1I		N
labs01t_ALP1P		N

Output Change Report

----- Filename = labs01t_ALP1I -----	
<u>Old/</u>	
<u>New</u>	<u>Differences</u>
O	Alkaline Phosphatase (U/L)
N	Albumin (g/dL)
...	(same details for the other files)

PhUSE 2008

Example 4

For the final run of all TLGs all modules of the RMC macro can be used: The archiving module saves the last validated outputs and creates a clean environment. With the generation module a controlled batch run can be started and it can easily be verified that all programs are productive and the log files do not show any problems. The comparison module produces status outputs where all ADS and TLGs are reported to be equal.

Program Status Report indicating the validation status of each program

<u>Status</u>	<u>Program name</u>
Productive	der/adae.sas
	der/adcm.sas
	...
	rep/cond/dm01t.sas
	rep/cond/dm02t.sas
	rep/saft/ae0231.sas
	...

Log-Report summarizing relevant findings in all log-files

```

*****/.../der/adae.log*****
----- ERROR -----
----- FATAL -----
----- WARNING -----
----- NOTE: ... observations with duplicate key values were deleted -----
125: NOTE: 20 observations with duplicate key values were deleted.
----- INFO: (user defined messages) -----
120: INFO: The following NOTE about deletion of observations with duplicate key values has been
checked and is ok.

*****/.../der/dm01t.log*****
----- ERROR -----
----- FATAL -----
----- WARNING -----
----- INFO: (user defined messages) -----
...

```

Output Status Report for ADS summarizing change status of each data set

<u>Data Set Overview</u>								
<u>FILE</u>	<u>OLDVAR</u>	<u>NEWVAR</u>	<u>COMVAR</u>	<u>OLDOBS</u>	<u>NEWOBS</u>	<u>COMOBS</u>	<u>EQUAL</u>	<u>CHANGE</u>
1	ADAE	69	69	69	379	379	379	Y
2	ADCM	36	36	36	385	385	385	Y
...								
19	ADMH	36	36	36	430	430	430	Y

Output Status Report for TLGs summarizing change status of each TLG

Comparison of outputs against outputs in \$SAS.../out/validated

File Overview - Equal Files

<u>Filename</u>	<u>File</u>	<u>Status</u>	<u>Equal</u>
dm01t			Y
dm02t			Y
...			
ae0231			Y

Output Change Report is empty (no changes found)

PhUSE 2008

The various RMC reports show that no changes to the validated previous outputs occurred. The final run was performed successfully and can be considered as re-validated. The reports created by the RMC macro can be filed to the validation documentation.

FUTURE PROSPECTS

In addition to the already achieved improvements, the RMC macro could be further enhanced.

The RMC macro was designed for a UNIX environment. In its current version, it is not directly executable in a Windows environment because of the applied UNIX scripts and commands. With the addition of equivalent Windows commands it could be transferred to a version which is both UNIX and Windows compatible.

As explained above, ADS files are compared by application of SAS proc compare but without the possibility of specifying ID variables. It is not intended to enhance the RMC macro in this respect. The consideration of too many individual problems regarding the single ADS would bloat the macro unnecessarily. For further exploration of changes, various tools can be used for both ADS and TLGs as discussed by Jasmin Fredette and Brian Fairfield-Carter ^[1].

Currently, change control is implemented for tables and listings which are produced as standard SAS outputs. However, the increased request for .rft outputs might be worth extending the macro by introducing the functionality of automatically comparing .rft files, as discussed by Jasmin Fredette and Brian Fairfield-Carter ^[1]. The possibility of overlooking particular types of discrepancies need to be further explored.

After the development phase of the macro, the idea came up to create the RMC Reports in different formats, i.e. as standard SAS outputs as well as .html outputs. Html outputs provide the advantage of hyper-linking, i.e. data set names and output names in the Output Status Reports could be hyperlinked to the Log Report and Output Change Report for quick navigation through large documents. During a test phase this was already realized for a small number of programs and outputs with few changes. However, it was not feasible for the typically large number of programs and outputs with many differences. Even with the simplest .html formatting, the RMC Reports grew to sizes that were not manageable in a browser. Future trends of .html formatting should be watched to further investigate this interesting topic.

CONCLUSION

RMC supports the process of analyzing clinical study data from the very beginning of program development until the final delivery of all study results, i.e. all final statistical results and TLGs for use in study reports or integrated summaries. Whilst change control for SAS programs is already state of the art, RMC mainly focuses on output change control. Additionally, RMC supports program development by allowing for an automatic re-run of all available programs "at the push of a button". Summary Reports produced by RMC show the status of each program and highlight all noticeable findings in the Log files.

From the beginning of the validation phase RMC enables archiving of previous outputs and automatic change control against updated versions. RMC provides concise reports on the status of the outputs as well as an overview of all changes. This facilitates the process of verifying that program changes work correctly and to identify unintended changes. Depending on the findings, the programmer can decide on the adequate approach for the required corrective actions. Since the results of RMC processing presented in the Summary Reports across programs and outputs show traceable evidence of change/no change, they can be filed together with the validation documentation. Especially at a late stage of a project, when all programs and outputs are validated and considered as final but changes become necessary, reliable change control of outputs is required to minimize the risk of unintended and unrecognized changes.

Avoiding the risk of accidental changes is essential therefore **Spot the difference:**

Re-compare Outputs after each run!!!

Minimize the risk!!!

Clarify changes!!!

PhUSE 2008

REFERENCES

[1] Jasmin Fredette and Brian Fairfield-Carter (PharmaSUG 2008): Using *Automated File Comparisons to Increase Efficiency and Accuracy in SAS Code Development and Validation*

ACKNOWLEDGMENTS

We would like to thank Martin Hall from Roche, Christoph Helwig from Accovion for their valuable help and input to this paper, Christian Müller from Roche for supporting this paper, and Elke Schüler and Judith Neudek from Accovion for initiating and promoting the development of the RMC macro.

CONTACT INFORMATION

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

Anja Feuerbacher
F. Hoffmann-La Roche Ltd.
PDIB (PDI)

Bldg. 670/R. 316
Malzgasse 30
4070 Basel, Switzerland

Work Phone: +41 61 68-80216
Email: anja.feuerbacher@roche.com

Beate Hientzsch
Accovion GmbH
Helfmann-Park 10

65760 Eschborn, Germany

Work Phone: +49 6196 7709-406
Email: beate.hientzsch@accovion.com

Brand and product names are trademarks of their respective companies.