



AUTOMATION OF CHECKS ON A PROJECT WITH THE SUBSEQUENT CONVENIENT REPORT OF THEM

12-October-2018, Warsaw
Arseny Khaplanov

PAREXEL[®]
YOUR JOURNEY. OUR MISSION.[®]

AGENDA



- **Introduction**
- **Three subsets of checks**
 - Scan log files
 - Saving compares
 - Collecting timestamps
- **Final analysis Report**

INTRODUCTION

WHY DO WE NEED SUCH SYSTEM?



THREE LEVELS OF PRODUCING ANALYSIS REPORT

SDTM

- **Source Datasets**
- **SDTM specification**
- **SDTM programs**
 - Log files
- **QC SDTM programs**
 - Results of compare

ADAM

- **SDTM datasets**
- **Auxiliary data**
- **ADaM specification**
- **ADaM programs**
 - Log files
- **QC ADaM programs**
 - Results of compare

OUTPUTS

- **ADaM datasets**
- **?SDTM datasets**
- **Outputs description**
- **TLFs programs**
 - Log files
- **QC TLFs programs**
 - Results of compare

THREE SUBSETS OF CHECKS



STATUS OF THE PROJECT

We should have status tracker to know answers for the following questions:

- What?
- Who?
- What program?
- Done (completed)?

Type of output	Output ID	Main programmer	Main program	Status	QC programmer	QC program	QC Status
SDTM	DS	Alex	Ds.sas	Complete	Smit	Ds.sas	In progress
SDTM	DM	Smit	dm.sas	Complete	John	Dm.sas	In progress
ADaM	ADSL	John	Adsl.sas	Complete	Alex	Adsl.sas	
Table	demography	Olga	Demo.sas	Complete	Peter	Demo.sas	Complete

THREE SUBSETS OF CHECKS

- **Scan of Log files for each level**

- To be sure that all programs ran properly

- **Saving compares**

- To have one or several datasets with results of all compares

- **Collecting timestamps of everything**

- Source datasets had been created earlier than aim datasets was created
- All programs were submitted in the correct order
- All above checks were done after the submission of the corresponding programs
- Any additional checks which system specific (e.g. checks of subversion control)
- e.t.c.

SCAN LOG FILES

- Create some library of issues with as many types of messages in log files as we know
- Designating every unknown message as “UNKNOWN”
- Save results of all also scanned files in a project specific dataset

Path	Program name	Date of the file	Time of the file	Result
.../prog/sdtm/	ds.sas	11-10-2018	12:00:50	No issues
.../prog/sdtm/	dm.sas	11-10-2018	12:00:55	Warning
.../prog/adam/	adsl.sas	12-10-2018	00:01:17	Warning
.../prog/tlf/	demography.sas	13-10-2018	05:23:18	Error

DOUBLE PROGRAMMING AND PROC COMPARE

DOUBLE PROGRAMMING AND PROC COMPARE



- For each output or dataset we need to have two datasets: main and control.
- For example, table.dm
c_table.dm,
- `proc compare base=table.dm
compare=c_table.dm;
run;`

DOUBLE PROGRAMMING AND PROC COMPARE



How to save result of compare?

1. Indirect way:

- Save results of proc compare in the file;
- Submit macro which do the following:
 - Goes thought all such files and parse them;
 - Saves results somewhere.

DOUBLE PROGRAMMING AND PROC COMPARE



How to save result of compare?

1. Indirect way:

- Save results of proc compare in the file;
- Submit macro which do the following:
 - Goes thought all such files and parse them;
 - Saves results somewhere.

2. Direct way:

- Use macrovariable sysinfo:

Proc compare base=... compare=...;

Run;

%PUT SYSINFO=&sysinfo.;

SYSINFO is number [0,65535].

SYSINFO MEANING

SYSINFO=0011100000101010.

Bit	Description	Bit	Description
1	Data set labels differ	9	Base data set has BY group not in comparison
2	Data set types differ	10	Comparison data set has BY group not in base
3	Variable has different informat	11	Base data set has variable not in comparison
4	Variable has different format	12	Comparison data set has variable not in base
5	Variable has different length	13	A value comparison was unequal
6	Variable has different label	14	Conflicting variable types
7	Base data set has observation not in comparison	15	BY variables do not match
8	Comparison data set has observation not in base	16	Fatal error: comparison not done

SAVING COMPARES

After each “proc compare” QC programmer needs add call of macro `%parse_compare(sysinfo,<type>,<name_of_output/dataset>)`.

Macro parses sysinfo, and saves the results in the dataset.

Example of the dataset:

Type of output	Output name	Date of the compare	Time of the compare	Result
S	DS	12-10-2018	15:17:50	0000000000001101
S	DM	12-10-2018	15:17:55	0000000000000001
A	ADSL	13-10-2018	18:15:17	0001001000000001
T	DEMOGRAPY	13-10-2018	19:00:18	1000000000000000

COLLECTING TIMESTAMPS OF EVERYTHING



Timestamps of the following:

- Every program on all levels
- Every source and every result dataset
 - Source dataset
 - SDTM, ADaM dataset
 - Dataset of the output
- All outputs
- All log files
- All compares
- All documentation which is used in the production
- Any additional company- and sponsor- specific checks

FINAL ANALYSIS AND REPORT



PRE-RUN OF THE MACRO

- **Save the address of the status file somewhere;**
- **Know or save paths of all directories which is used;**
- **Be sure that status was created from the standard template.**

PRE-RUN OF THE MACRO

- **Save the address of the status file somewhere;**
- **Know or save paths of all directories which is used;**
- **Be sure that status was created from the standard template.**

If we know everything above, we can submit the macro:

*/.../.../<project>/projectStatus
status_is_downloading
status_is_analyzed
File Status_Report_<project>.pdf has been created.*

FINAL REPORT

- So macro collect as much information as possible. Below are examples of issues, which macro is able to detect.
- It report all issues by programmer. It will help programmer to find their own issue and resolve them.

<Olga>:

- Output/dataset
- Program
- List of issues
- Report issues only if task has been completed by default.

Primary Checks	Programmer Checks	QC-er Checks
There is no programmer for this output! There is no QC-er for this output!	Programmer did not prepare output for QC There are some warnings in LOG There is no Scan LOG information	QC-er has not prepare QC while ADS was prepared QC-er has not prepare QC while output was prepared QC-er has completed QC process even output had been not prepared
	Output was not found Output was not recreated after last updating ADS There is no program name in status Programm was not found Output was not recreated after last change of program	There is no any information about QC status from parse_compare macro There is no QC program name QC program was not found QC-er ran his program before output creation QC program has not be rerun after modification of QC program
	There are some errors in LOG	QC was not completed: <List of differences from parse_compare>

FINAL REPORT

An example of the report based on the status in the beginning:

Alex		
DS	ds.sas	There are some warnings in LOG Output was not found
ADSL	adsl.sas	QC-er has not prepare QC while output was prepared
Smit		
DM	dm.sas	Output was not recreated after last change of program Programm was not found
DS	ds.sas	QC-er has not prepare QC while ADS was prepared
John		
DM	dm.sas	QC-er has not prepare QC while ADS was prepared
ADSL	adsl.sas	Output was not recreated after last updating SDTM
Peter		
demography	demo.sas	QC-er ran his program before output creation QC was not completed: Different Labels; Base data set has observation not in comparison;

THANK YOU

*Arseny Khaplanov
Principal Biostatistician
PAREXEL International
Arseny.Khaplanov@PAREXEL.com*