

ClinXPORT a software to streamline the analysis and reporting of clinical trials with SAS®

Julien Vanwinsberghe, ClinBAY, Genappe, Belgium
Philippe Remusat, ClinBAY, Genappe, Belgium
Francois Vandenhende, ClinBAY, Genappe, Belgium

ABSTRACT

The reporting process in clinical trials can be schematized in three phases: standardization and derivation of source datasets, statistical analysis and calculations, and finally results reporting by creation of the output tables, listings, and graphs. The method proposed here is to follow a simple, clear, structured streamline for these activities, and to automate the coding of the last phase in order to focus SAS programming (and validation) to only data derivations and calculations. The concept is based on the definition of all levels of information (project parameters, output shells, etc) in an Excel workbook, and using named ranges to identify them. Then, a set of SAS macros is used to read these meta-data and automate the outputs production. This approach has been implemented and validated on more than 60 studies for several companies in pharmaceutical industry. The estimated time rate gained in reporting activities is about 25%.

INTRODUCTION

SAS programmers in pharmaceutical industry have the responsibility of generating a variety of reports (usually in form of tables, listings and graphs) to fulfill clinical regulatory requirements. Examples of these functional reports include summary and listings of demography, adverse events, subject disposition, protocol deviations, laboratory results, etc.

Each type of report requires a specific layout that can be defined and reused in form of a template. Furthermore, by establishing clinical reporting templates for each type of output, the SAS programmer can also create, validate, and reuse specific SAS reporting programs for generating an output, by using the information defined in associated template. Only minor adjustments may be required to reuse templates and programs in new projects, and all the study specific information can be defined in the templates for not having to change a validated SAS program when it is reused.

In this paper, we will present a method going even further, which consists in defining all the study specific information in a Microsoft Excel workbook (list of outputs, project information, output templates, etc), and using a set of validated SAS macros and reporting programs to read these informations in order to generate output reports.

The SAS macros automate all the repetitive SAS code (proc report statement is done automatically by reading template information for example), and the reporting programs are validated and reusable without any modification for standard outputs.

This method has been implemented in form of software called ClinXPORT including two main components: an add-in for Microsoft Excel 2007/2010, and a set of 73 SAS macros called BUM (Broad Used Macros).

MANAGING THE REPORTING PROJECT IN TWO ENVIRONMENTS : EXCEL AND SAS

FOLDER ORGANIZATION OF A CLINXPORT REPORTING PROJECT

The recommended structure of folders for a reporting project can be summarized as following:

- ➔ DATA : Data folder containing SAS derived datasets
- ➔ PGM: SAS reporting programs
- ➔ OUT: Output reports
- ➔ UTIL: Excel workbook and meta-data files

PhUSE 2011

Name	Type	Size
data	File folder	
out	File folder	
pgm	File folder	
util	File folder	
ClinXPort_Runme.sas	SAS Program	4 KB

Figure 1. Example of folder organization for a reporting project

This is an example, which allows separating each type of files for a better organization, but any folder structure can be used, on both Microsoft Windows or UNIX environments. ClinXPORT can also be adapted to specific folder environments as version controlled systems or document management systems.

DEFINING REPORTING REQUIREMENTS AND META DATA IN EXCEL

Very often, the shells for tables, figures and listings are created in an excel document and then, the SAS programmer has to recreate the same layout. To do so, he often has to spend time with the options of the SAS proc report statement to get the closest and best layout possible, with all the columns on the same page, an adequate label and format etc... These steps take a long time. It is possible to reduce time for creating the shells in the excel document, for managing the SAS programming in only one place (data, programs, outputs and shells) and for saving time to the programmer by directly reusing the excel worksheet in the programming step.

THE STRUCTURE OF A CLINXPORT EXCEL WORKBOOK

A ClinXPORT workbook includes two main worksheets called "Study" and "Domain" (names can be changed) containing respectively general options and information about the project, and a listing of all outputs to be created. An additional worksheet can be used to define SAS formats in a user-friendly way, and a worksheet is created for each of the output templates.

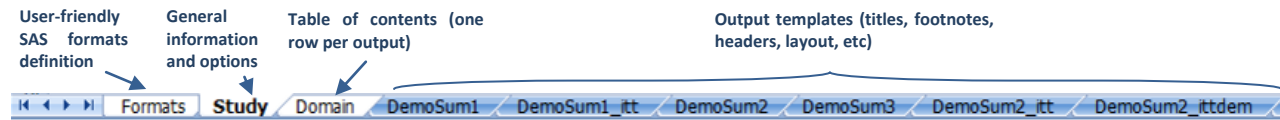


Figure 2. Screenshots of the worksheet names of a ClinXPORT Excel workbook

The "Study" sheet defines general information about the study (names of programmer and statistician, study number, etc), and specific options used during the SAS reporting activity (missing values in reports, footnote alignment, SAS options, etc). Then options will be available as macro variables for all the SAS session to set up the environment and the output processing.

sponsor <Sponsor name>							
A	B	C	D	E	F	G	H
Sponsor	Project Code	Study Number	Study Title	Statistician	Programmer	Treatment Label	Missing value in reports
<Sponsor name>	<Project code>	Demo study	<Study title>	<Statistician name>	<Programmer name>	Treatment: 150 mg AAA-XXXXYY	--
footnote alignment	topline	bottomline	pagexofy	domain sheet name	Debug	SAS options	Format sheet path
left	no	no	Page xx of xx	Domain	YES	options nospool center ls=120 nobylne nodate nonumber ps=44 NOFMERR fntsearch=(sformat) mprint nosymbolgen nomlogic NOSYNTAXCHECK NODMSSYNCHK;	&projectpath.\util\for mats.sas
Treatment label	Treatment variable name	Treatment order name	Save metadata in project directory	User defined macro folder	Auto-open outputs	Create template datasets	Use bodytitle in ITT
Treatment	TREATVAR	TREATORD	YES		NO	NO	YES

Figure 3. Screenshot of the study worksheet

The "Domain" sheet lists all the outputs to be generated in the reporting project. Each row corresponds to an output report to produce, and about 20 columns give details about the output (title, number, type, orientation, format, file name), the input data (dataset name, data filter, status), the reporting program (program name and status), the paths (data, program, report paths), the template name (with an hyperlink to navigate to the template definition) and more (specific value to pass to the program, etc).

PhUSE 2011

Comments	Use	Domain Name	Template Name	Output Type (Table, Listing, Figure)	Output number	Table title	Population	Output File Name	Output Format	Output orientation
In-text tables										
Subject Disposition										
	X	Demo	DemoSum2_tt	Table	1-1	Subject disposition - n (%) of patients	All subjects	t1_01	rtf	L
Demograph, PD and PK endpoints										
	X	Demo	DemoSum1_ttt	Table	2-1	Demographic summary	Safety analysis set	t2_01	rtf	L
Adverse events										
	X	AE	AEsum1_ttt	Table	3-1	Adverse events overall and frequently affected system or	Safety analysis set	t3_01	rtf	L
Post-Text Tables										
Demographic Data										
	X	Demo	DemoSum2	Table	4-1.1	Subject disposition	All subjects	t4_1_01	txt	
	X	Demo	DemoSum3	Table	4-2.1	Number (percent) of subjects in the analysis sets	All subjects	t4_2_01	txt	
	X	Demo	DemoSum1	Table	4-3.1	Summary of demographic information	Safety analysis set	t4_3_01	txt	
Safety Data										
	X	Lab	LabSum1	Table	5-1.1	Summary of blood chemistry: SOD POT CHLOR BIC CALC PHOS	Safety analysis set	t5_1_01	txt	
	X	Lab	LabSum1	Table	5-2.1	Summary of hematology: EOS LYH BAS MON	Safety analysis set	t5_2_01	txt	

Figure 4. Screenshot of the Domain worksheet

When clicking on a template name, the corresponding template sheet is selected. As it can be seen on following screenshot, one can define the title, footnote, header text, SAS variable names corresponding to the columns, proc report options (for automating the proc report statement), column widths, and column alignment. The macro variables colored in red correspond to meta-data defined in the study (&studynumber for example) or domain (&outpttitle for example) sheets, which will automatically be replaced by their values during the SAS session (see figure 16 for an example of output created from this template).

Center/ Country Subject	Initials	Predominant Race	Ethnicity	Sex	Date of screening	Date of birth	Age (yr) + (cm) ++ (kg) ++ (kg/m2)
TRT1BL1 USA	001/5105 DRH	Caucasian	Other	Female	ddMmYyyy	ddMmYyyy	10 182 96.3 29.1
	001/5105 DRH	Caucasian	Other	Female	ddMmYyyy	ddMmYyyy	10 182 96.3 29.1
	001/5105 DRH	Caucasian	Other	Female	ddMmYyyy	ddMmYyyy	10 182 96.3 29.1

SAS variable names

Country	CS	SUBJIN	RACE	ETHN	SEX	DATE	SOBIRTH	AGE	HGT	WGT	BMI
group flow order=	data	group order=	data	format=RACE_	format=ETH_	Flow	format=SEX1_	format=3	format=6.1	format=4.1	

Proc report options

Footnote

Program: &programpath.\&programname.sas %sysfunc(date(),date9.):%sysfunc(time(),time5.) &programmode.

Figure 5. Screenshot of a template worksheet (DemoList1)

THE CLINXPORT ADD-IN AS A TOOLBAR IN MICROSOFT EXCEL 2007/2010

After installing the ClinXPORT software, a new tab called "ClinBAY" appears in the Microsoft Excel application. This tab provides several functionalities for defining meta-data, exporting it for SAS, executing SAS for generating the outputs, opening outputs and programs, etc.

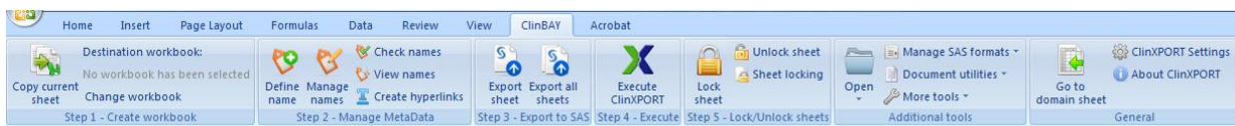


Figure 6. Screenshot of the ClinXPORT toolbar in Microsoft Excel

DEFINING META-DATA IN THE EXCEL WORKBOOK

To allow the SAS macros to read and identify the information defined in the Excel workbook, we use the functionality of Microsoft Excel for creating named ranges. A named range is a cell or a group of cells referenced by a name.

For example, we can see on following screenshot that the cell C2 of the study sheet is referenced with the name "studynumber" visible in the upper left corner.

PhUSE 2011

studynumber		
Demo study		
A	B	C
Sponsor	Project Code	Study Number
<Sponsor name>	<Project code>	Demo study

Figure 7. The named range "studynumber" in the Study sheet

A group of buttons from the ClinXPORT toolbar (see following figure) allow creating, managing and selecting named ranges in the worksheets. All the information defined in the workbook can then be easily identified by a name, for being used in the SAS session.

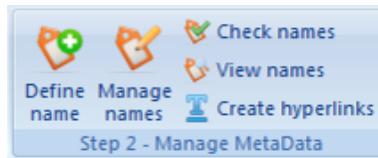


Figure 8. ClinXPORT buttons for managing named ranges

PREVIEWING OUTPUTS IN A REPORT

When the Excel workbook has been populated with the list of all outputs and their templates (usually done when creating the report analysis plan, commonly called RAP), a Microsoft Word report can be generated to allow the team for reviewing the output specifications. The template of each output listed in the domain sheet (table of contents) is copied in a page of the output document, with macro variables references automatically replaced by their values (searched in study and domain meta-data). Following figure shows the template "Demolist1" previously seen in figure 5, which has been copied in the Microsoft Word document with all red macro variable references replaced by their values.

Listing 7-5.1: Demographic information (DemoList1)

Demo study

Listing 7-5.1 (Page xx of xx)

Demographic information

All subjects

Treatment	Center/ Country	Subject	Initials	Predominant Race	Ethnicity	Sex	Date of screening	Date of birth	Age (yr)+	Height (cm)++	Weight (kg)++	BMI (kg/m2)
TRTBL1	USA	001/5105	DRH	Caucasian	Other	Female	ddMmYyyy	ddMmYyyy	10	182	96.3	29.1
		001/5105	DRH	Caucasian	Other	Female	ddMmYyyy	ddMmYyyy	10	182	96.3	29.1
		001/5105	DRH	Caucasian	Other	Female	ddMmYyyy	ddMmYyyy	10	182	96.3	29.1

Treatment: 150 mg AAA-XXXXYY

Note: + Age is calculated from date of screening and date of birth

++ Weight and height are taken from Screening Vital Signs evaluations

BMI [kg/m2] = weight[kg] / (height[m]**2)

Program: <Broad Used Macros path>\rep_list.sas 01SEP2011:15:58 Draft

Figure 9. Exporting Excel workbook as a Word document for reviewing output specifications

EXPORTING META-DATA TO SAS FOR REPORTING

Each of the worksheets contains specific named ranges (options of the study sheet, columns of the domain sheet, and elements of the template sheets) that will then be exported as SAS macro variables in order to be read, interpreted, and used in the SAS session. A button called “Export sheet” allows the user to export named ranges of the selected worksheet in a SAS meta-data file, and a button called “export all sheets” to do it for all the worksheets in the workbook.

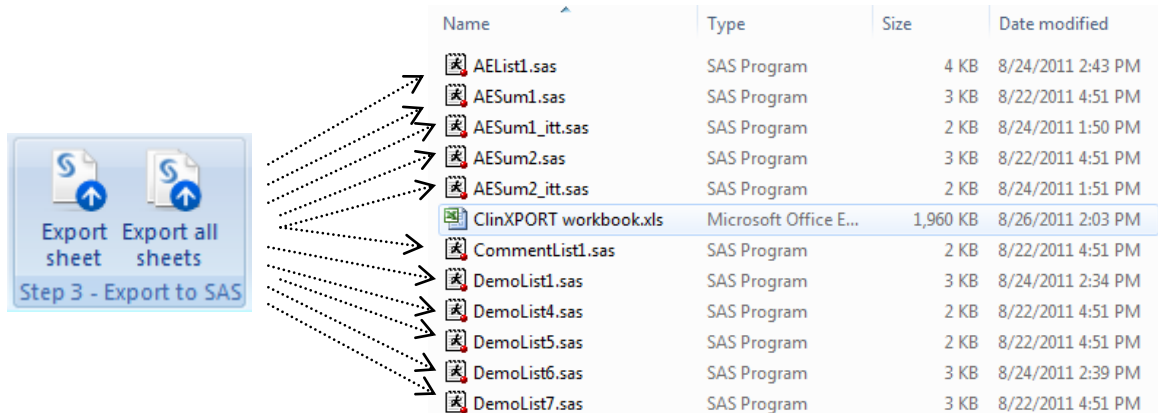


Figure 10. ClinXPORT buttons for exporting named ranges of worksheets as SAS meta-data files

As noticed before, the study meta-data is used to set up the SAS session and parameterize the output processing (auto-opening reports as they are created for example). After loading study meta-data, the SAS macros of the BUM read the domain meta-data and loop on the data rows. For each one, they load the corresponding template meta-data and execute the associated reporting program for generating the output (this can be a specific macro from the BUM if no calculation or data derivation are required).

THE SAS PART: SPECIFIC DATA DERIVATION AND CALCULATION

THE BROAD USED MACROS (BUM)

So far, ClinXPORT includes a set of 73 SAS macros. They can be divided in three types:

- The core macros: for reading meta-data, executing reporting programs, post-processing outputs and checking them
- The reporting macros: used in the reporting programs for automating repetitive tasks (calculation of summaries, data filtering, proc report automation, auto-generation of headlines, etc)
- The utility macros: used in the reporting programs for specific utility functions (checking if a value is numeric, splitting strings, deleting SAS objects, etc)

PhUSE 2011

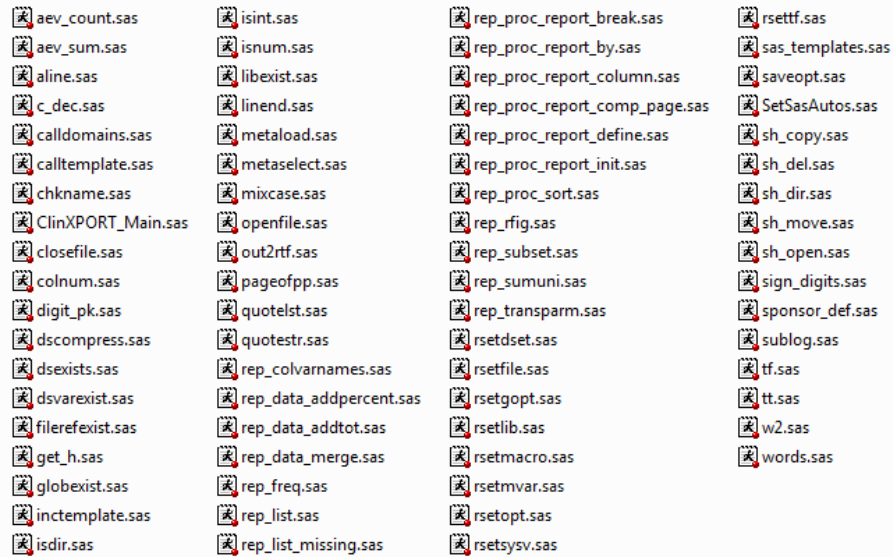


Figure 11. The ClinXPORT SAS Broad Used Macros (BUM)

These macros are automatically loaded in the SAS session and available in the reporting programs of the project. One of the macros, called “ClinXPORT_Main” can be executed with the meta-data path as input parameter, to execute the whole process of reading meta-data and creating outputs.

Following figure shows the structure of a standard main program of a ClinXPORT project for calling the macro “ClinXPORT_Main” and executing ClinXPORT. This program can be automatically executed from Excel with a button called “Execute ClinXPORT”.

```
%*=====;
%* Setting paths                                     ;
%*=====;

* This path can be automatically generated with functions SAS_EXECFILEPATH and macro variable &sysin;
%let projectpath=<Path of the reporting project>;
%let BUMPATH=<Path of the ClinXPORT Broad Used Macros>;

%*=====;
%* Call to ClinXPORT_Main with relevant path          ;
%*=====;

%inc "&BUMPATH./ClinXPORT_Main.sas";
%ClinXPORT_Main(%nrquote(&projectpath./util));
```

Figure 12. Example of a main SAS program for a reporting project

There is no limitation to the number of these macros. Some can be added, modified or customized according to the real needs.

THE STRUCTURE OF SAS REPORTING PROGRAMS

Each of the reporting programs referenced in the domain sheet must have the same structure, it must be a SAS macro with two input parameters called “data” and “subset”. The BUM will automatically define the values of these two parameters from the domain meta-data and execute the macro. As we can see in the following figure, the only SAS code needed in reporting programs is for specific data processing (data is read and filtered with macros rsetdset and rep_subset, and the “proc sort” and “proc report” statements are automatically generated from template meta-data).

PhUSE 2011

```

%macro p_ECGList2(data,subset=%str(1));

    %*=====;
    %* Read and treat input data
    %*=====;

    * reset ADS dataset;
    %rsetdset(names=ads ads2, lib=work);

    * create ADS dataset by subsetting from &data using metadata definitions;
    %rep_subset(in=&data, out=ads, subset=&subset);

    %*=====;
    %* Flags out of range QT values
    %*=====;

    data ads2;
        set ads;
        if QTcB > 450 then QTcB_ =put (QTcB,6.2) || " +";
        else QTcB_ =put (QTcB,6.2);
    run;

    %*=====;
    %* Execution of proc report statement
    %*=====;

    %rep_list(ads2);

%mend p_ECGList2;

```

Figure 13. Structure of a basic SAS reporting program (Listing of ECG intervals)

The macro “rep_list” used for automating the “proc report” statement in previous figure can also be used as SAS reporting program in the domain sheet, for producing listings automatically without needing any line of SAS code.

	A	B	C	D	E	F	G	M
	Comments	Use	Domain Name	Template Name	Output Type (Table, Listing, Figure)	Output number	Table title	Program Name
1								
36		X	Demo	DemoList1	Listing	7-5.1	Demographic information	rep_list
37		X	Demo	DemoList5	Listing	7-5.2	Visit dates	rep_list
38		X	Demo	DemoList6	Listing	7-5.3	Relevant medical history/current medical conditions	rep_list
39		X	Demo	DemoList4	Listing	7-5.4	Pregnancy test results	rep_list
40		X	Demo	DemoList7	Listing	7-5.5	Meal records	rep_list
41	Compliance and/or Drug Concentration Data							
42		X	Dose	DoseList5	Listing	8-1.1	Dosage and dates of administration	rep_list
43		X	Med	MedList1	Listing	8-2.1	Prior and concomitant medications	rep_list
44	Adverse Event Listings							
45		X	AE	AEList1	Listing	6-2.2	All adverse events	p_aelist1
46	Listing of Individual Laboratory Measurement By Patient							
47		X	Lab	LabList2	Listing	5-5.3	Subject laboratory profile - Biochemistry	p_lablist2
48		X	Lab	LabList2	Listing	5-5.4	Subject laboratory profile - Hematology	p_lablist2
49	Vital Signs, Physical Findings and Other Observations Related To Safety							
50		X	VS	VSList3	Listing	9-2.1	Vital signs	rep_list
51		X	ECG	ECGList1	Listing	10-2.1	ECG interpretation	rep_list
52		X	ECG	ECGList2	Listing	10-2.2	ECG Intervals	p_ecglist2
53		X	Com	CommentList1	Listing	11-1.1	Comments	rep_list
54		X	BLD	SampleList1	Listing	12-1.1	Blood collection log	rep_list

Figure 14. Using macro rep_list as reporting program for creating listings automatically

IN PRACTICE: INTERACTIONS BETWEEN EXCEL AND SAS

EXECUTING AND OPENING PROGRAMS AND OUTPUTS FROM EXCEL

A button called “Execute ClinXPORT” allows the SAS programmer to easily execute the SAS reporting project from Microsoft Excel. When the SAS main program is executed, all the outputs crossed in the column “Use” (see figure 14) of the domain sheet are generated. An option from the study sheet allows to automatically open the generated

PhUSE 2011

reports as going along. A short version of the SAS log keeping only important messages (error, warnings, and specific note messages) can also be generated and automatically opened at the end of the execution.

As described above, some reports may thus be created without having to open SAS software or generating SAS codes. Not only for standard SAS programs that can be reused, but also for all the listings, even new ones, which can be designed in Excel templates, and created automatically with the macro rep_list (see figures 5, 14 and 16 for an example with the template “demolist1”) if all the required data derivations have been performed in input derived dataset.

When selecting a row in the domain sheet, some buttons allow opening the corresponding output or reporting program (if existing) by reading the report or program path, and output or program name from the selected row.

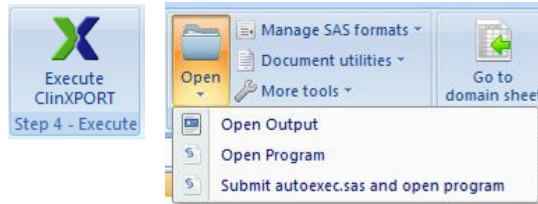


Figure 15. Executing and opening programs and outputs from Excel

For the SAS programmer, it is possible to load a specific study environment in SAS (BUM, data libraries, SAS formats, custom macros, etc), when opening the SAS reporting program with the button “Submit autoexec.sas and open program”. A SAS program called “autoexec.sas” must be created in the project folder, and it will be automatically loaded in the SAS session when opening the program. In that case, the macro variables defining dataset input and subset are also loaded.

MODIFICATION OF OUTPUT LAYOUT

The output layout can be fully controlled and customized from the Excel workbook. Some options in the study sheet allow adding horizontal lines at the top or the bottom of output tables, or putting titles and footnotes of in-text tables in the body part of the documents. In the template sheets, if the user changes the column widths or header alignments as for a common Excel table, the changes are reflected on the output reports.

For example, in figure 16, we apply following modifications (in blue) in the template “DemoList1” (see original template in figure 5):

- Adding a reference to named range “templatename” in title to display the template name “demolist1” in output title
- Setting the format of BMI from 4.1 to 6.2
- Changing title of column “Predominant race” to “Race”
- Removing Ethnicity column
- Enlarging first treatment column and replacing treatment name by treatment group

PhUSE 2011

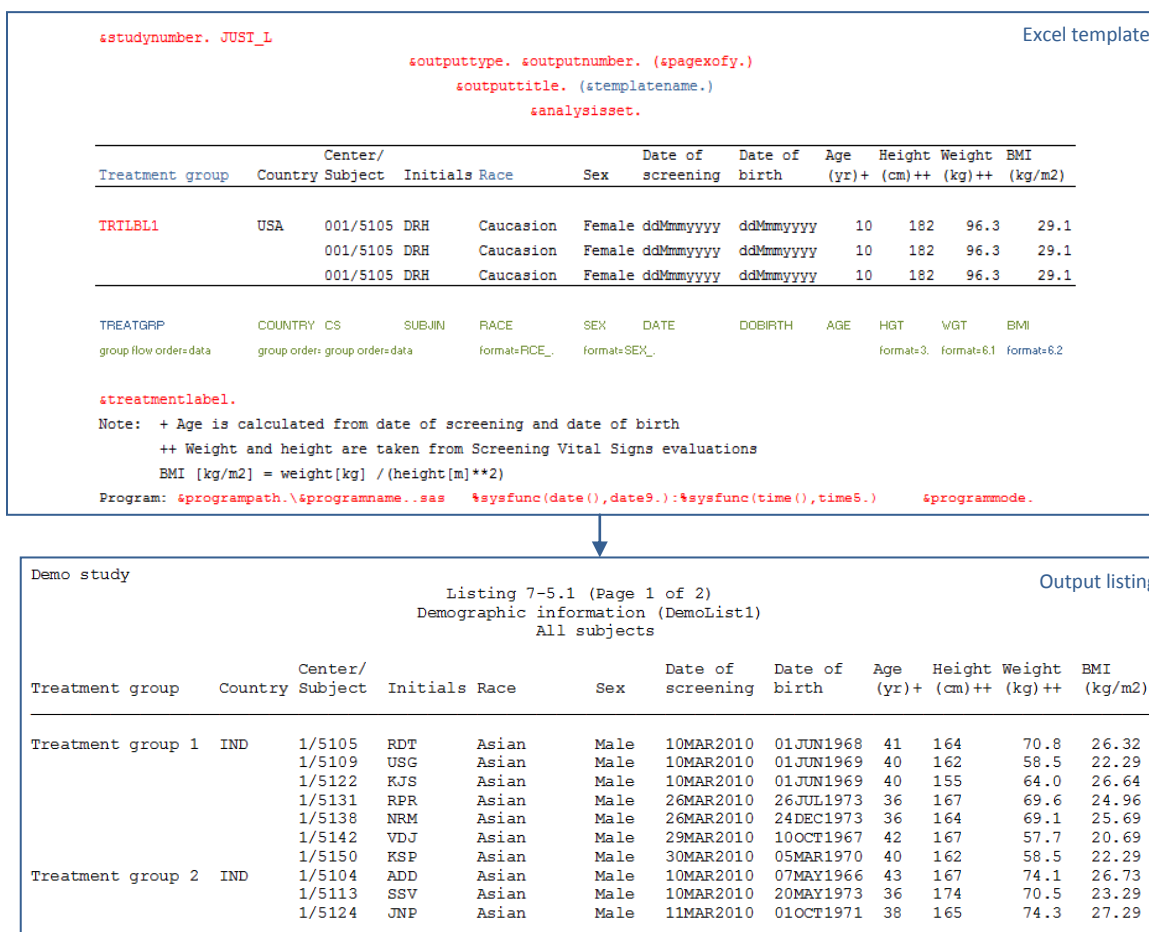


Figure 16. Modifying output layout from Excel without SAS coding

Thanks to this, if one shell for an output is modified by somebody who is not the SAS programmer, the SAS program is automatically updated, saving time and a chance to miss a modification.

CONCLUSION

By extracting all the study-specific information from the SAS reporting programs and the output templates into two worksheets of an Excel workbook, the reusability of a reporting project is dramatically increased. The validation effort is also significantly decreased as the SAS programs can be reused with minor or without any changes (for standard outputs), or completely replaced by ClinXPORT macros for producing listings automatically.

The system presented in this paper is very flexible, as it can be adapted to any folder structure, SAS reporting programs, user-defined macros, output layouts, etc. It has been tested and used in production environment for several pharmaceutical companies with specific standard operating procedures and output specifications, and is in continuous improvement to fit new needs and standards of clinical trial reporting. Feel free to contact use for any additional information.

CONTACT INFORMATION

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

Julien Vanwinsberghe
ClinBAY
30 rue Bon Air
B-1470 Baisy-Thy, Belgium
Work Phone: +32 (0)67709084
Email: julien@clinbay.com
Web: www.clinbay.com

Francois Vandenhende
ClinBAY
30 rue Bon Air
B-1470 Baisy-Thy, Belgium
Work Phone: +32 (0)67709081
Email: francois@clinbay.com
Web: www.clinbay.com

Brand and product names are trademarks of their respective companies.