

Work Process from Data Management to Final Analysis File Creation Using SAS® Macros (Scope: Labs and Adverse Events)

Tony Salehi, Quintiles, Inc., Kansas City, MO

ABSTRACT

The development process for generating analysis files varies quite often between projects and even between studies within projects. This paper is intended to discuss a systematic, macro-utilized approach in SAS that generates the final analysis files. There are essentially three stages in the process flow from raw datasets to final analysis files: merging repeating and non-repeating datasets, adding treatment and demographic variables to the analysis files, and incorporating specific data manipulations to the analysis files. The adverse events data will be used as an example for this process.

INTRODUCTION

In the development process of generating analysis files, SAS datasets used in table and listing generation, there can often be wide variation between projects and even individual studies within projects. Invariably, there is some form of programmatic cannibalism from other projects or studies to assist in the creation of the analysis files. The programs are then modified to account for study-specific differences in, for instance, study design or treatment assignment. The final approach to creating these analysis files is a patchwork of new SAS code with pre-existing SAS code from other studies.

DISCUSSION

The present methodology in generating analysis files that is used in the clinical studies for any projects is a systematic, macro-utilized approach which uses the raw datasets provided by data management and creates analysis files that will be used in summary table and listing generation. A brief description of each step in the data flow will be explained. Figure 1 displays the process flow overview.

Data management uploads the raw datasets from the Oracle Clinical database onto the Unix platform. For some datasets, such as adverse events and laboratory data (e.g. hematology, serum chemistry, and urinalysis), the datasets are separated into two parts: non-repeating data, which is usually one record per patient; and repeating or time-dependent data.

The first macro that has been developed joins the non-repeating and repeating raw datasets from Oracle Clinical by Case Report Form document number (variable name, docnum). There are six parameters that are passed through this macro:

Inlib – the library reference for the location of the raw data.

Nrdata – the name of the non-repeating input dataset.

Repdata – the name of the repeating input dataset.

Drop – the variables to be dropped from the resulting joined dataset.

Outlib – the library reference for the location of the joined data. The default is WORK.

Outdata – the name of the output data. The default is &nrdata.

Once the non-repeating and repeating raw datasets have been joined together, the next macro that has been developed creates the second generation datasets by adding demographic and treatment population-related variables. Whereas the macro that

joins the non-repeating and repeating raw datasets by document number, this second generation macro uses the patient number to merge the datasets together. There are six parameters passed through this macro:

Lib – the input library. The default is RAW directory where the raw datasets are located.

Data – the input dataset.

Out – the output dataset.

Addpds – indicates whether or not there are scheduled events (value is Y or N).

Dtvar – the actual date of the scheduled visits.

Tmvar – the actual time of the scheduled visits.

In the Phase III studies, potentially clinically significant (PCS) flags are created in the vital signs data, laboratory measurements, and 12-lead electrocardiogram measurements. These PCS flags need to be created at this stage so that they will not have to be created in the actual programs that generate the summary tables. Furthermore, when generating an Integrated Safety Summary, having the PCS created in each of the study analysis file generation portion will save much time and potential confusion, since different studies may have different PCS ranges.

When the appropriate SAS programs have been written to account for the appropriate preprocessing and manipulation of the data and the macros have been written, then a batch program is created to create all the analysis files. This step is intended to automate the process so that the programmer does not have to run each program individually. Figures 2 and 3 detail the directory structures for the work flow process and accounts for both Phase I and III studies. There is an essential difference between the two studies. For the Phase III studies, all the programs and datasets reside on the Unix server. On the other hand, for the Phase I studies, only the raw datasets are on the Unix server and all the programs and final analysis files are on the NT server. Furthermore, these raw datasets must be FTP'd from Unix to the NT where they can be used to create the analysis files.

Described below is a typical portion of SAS code that utilizes the macros.

```
%**** initialization of variables and macro invocations **** ;
%include "../setup.sas";

%*** Use the include statement to perform additional data
preprocessing ***;

%macro _run2gen(prog);

%include "&_pgmpath/project/compound/study/programs/derive/
&prog.sas";

%mend _run2gen;

%**** Run treatment assignment program for all pts in study ****;
%_run2gen(treatmnt);

%**** bring in the raw adverse events data. **** ;
%**** AE.SSD is the non-repeating data, AEAE.SSD is the
repeating data ***;
```

```
%join_oc(inlib=raw,
         nrdata=ae,
         repdata=aeae,
         drop=dropvar1 dropvar2 dropvar3 dropvar4 dropvar5,
         outlib=work,
         outdata=ae)

%**** secgen macro incorporates txt assignments and txt
population flags **** ;

%secgen(lib=work,data=ae,out=work.ae1)

%**** differentiate between txt-emergent AE.s and pre-tx ae.s****;
%_run2gen(ae);
```

CONCLUSION

The process presented in this paper describes a method of incorporating raw datasets from data management and analysis files that are used in summary tables and listings using a macro-utilized approach. This is intended to encourage other SAS programmers in similar situations in the clinical trial process to discover new and improved approaches in preparing final analysis files for report summaries. The reader should note that, with any macro-utilized approach in programming, there are

individual preferences and other varying methods that will achieve the same desired result.

ACKNOWLEDGMENTS

The author would like to acknowledge Jay Zhou for the review of this paper.

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

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

Tony Salehi
 Quintiles, Inc.
 10245 Hickman Mills Drive
 Kansas City, MO 64134
 Work Phone: 816-767-3842
 Fax: 816-767-7347
 Email: tony.salehi@quintiles.com

Figure 1: Process Flow Overview

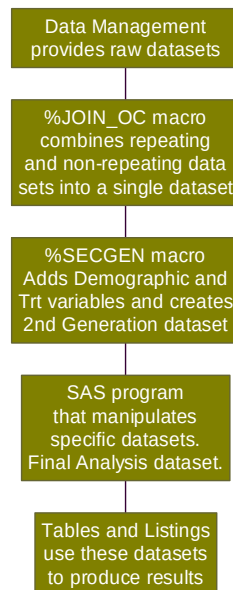


Figure 2: Directory Structure for Process Flow
(Phase III Studies on Unix Platform)

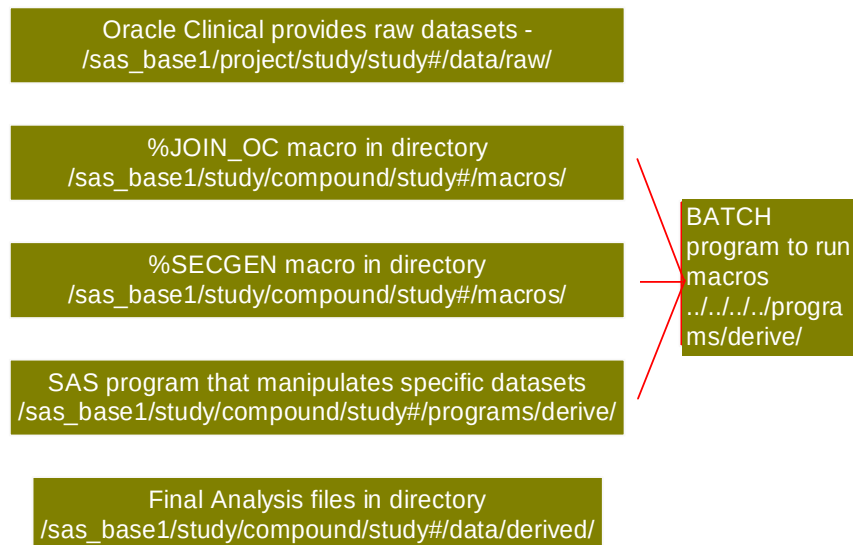


Figure 3: Directory Structure for Process Flow
(Phase I Studies on NT Platform)

