

PHUSE US Connect 2024

Eliminating Duplicative Maintenance Of Statistical Programs In Parallel Analyses Using A Single Control File

Kavitha Boinapally (kboinapally@seagen.com), Seagen Inc., Bothell WA

ABSTRACT

Clinical study programming involves working on multiple deliverables in parallel with different source data. This can be efficiently achieved by using a standard folder structure and single master programming environment set-up file. With use of macro variables and SYSUSERID, the switch between analyses, randomization files, blinded and unblinded (restricted) folders can be seamless without duplicating the programs in various folders. Leveraging SYSUSERID adds additional security. A global macro variable to indicate deliverable type and custom macro variables for each such type help maintain a single set of SDTM, ADaM, and sometimes TLF programs to generate datasets and outputs in different analysis folders with different source data. Archiving all programs and files in a zip file in a snapshot folder ensures easy retrieval and rerunning of deliverables (within that snapshot folder) without disruption to the regular study area. This approach can enhance productivity and data quality.

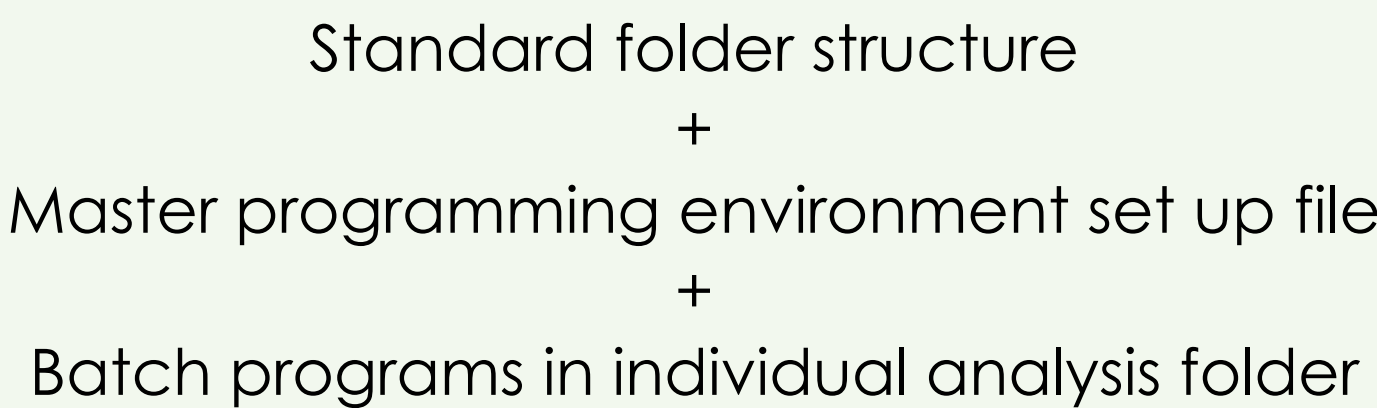
Introduction

From first patient enrollment to study completion, clinical study programming involves generating datasets and outputs for multiple deliverables for internal and external use. Depending on submission requirements some such deliverables are:

- AR:** Annual Report
- CSR:** Clinical Study Report
- DMC:** Data Monitoring Committee
- DSUR:** Development Safety Update Report
- IB:** Investigator's Brochure
- PBRER:** Periodic Benefit Risk Evaluation Report
- PSUR:** Periodic Safety Update Report

Internal
Adhoc

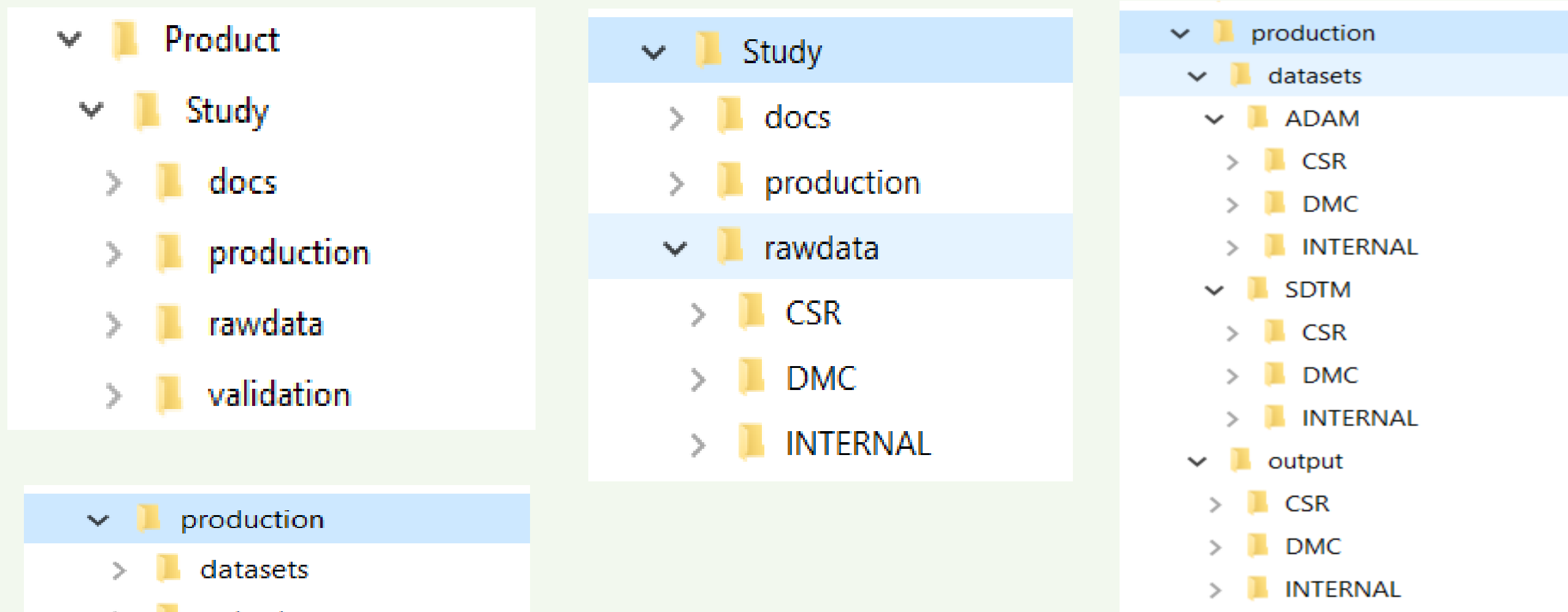
Maintaining separate sets of dataset and output programs for multiple deliverables and keeping track of changes across all these programs is cumbersome. This can be simplified by using one set of dataset programs.



output datasets in different analysis folders from one set of dataset programs

Folder Structure

Folder structure needs to be standardized across the analyses within the study. Having consistent structure will enable use of same librefs across the analyses. Example of consistent folder structure: Below structure works for different study designs with single or multiple deliverables. Showing folders for CSR, DMC and INTERNAL deliverables.



If a study needs both CDISC and non-CDISC datasets, separate folders can be created just for non-CDISC dataset programs and output datasets following similar structure as SDTM/ADAM folders.

When dataset/TLF programs are run, the output dataset/file would appear in one of the analyses folders – This is controlled in environment setup file.

Validation folder contains all subfolders same as the Production folder.

In this structure each type of dataset and TLF(program and output) folders contain separate folders for individual analysis (CSR, DMC and INTERNAL).

Under **Programs**, SDTM and ADAM folders will not contain any subfolders as there would be only one set of dataset programs for a study.

For blinded studies same folder structure will be replicated in the blinded and unblinded (restricted) production and validation areas.

Set up file

Most studies use an inclusion program to set up programming environment including set up of libnames, macro variables, study options, global titles and other study level items. To eliminative duplicative maintenance of programs, a master environment setup file that works on blinded and open label studies is a must and it defines:

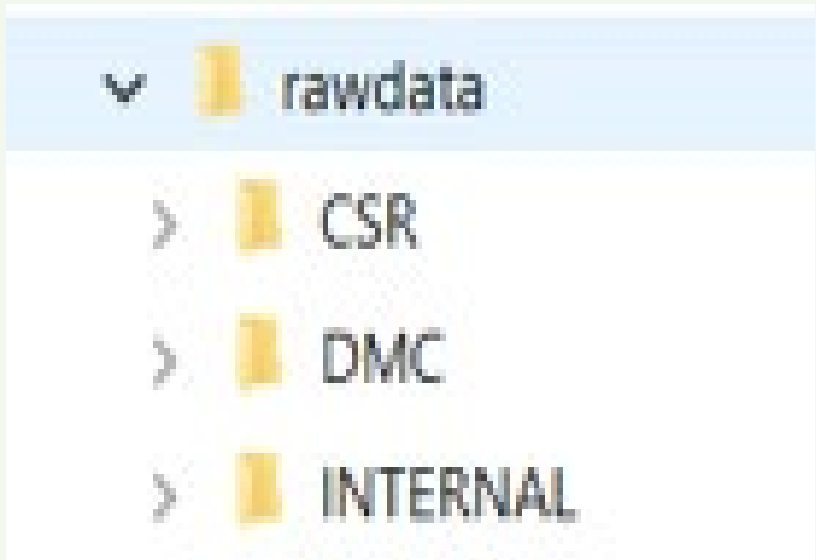
SYSUSERID: An automatic macro variable created in SAS containing the user ID running the program currently. A condition is placed in setup file based on SYSUSERID defining unblinded programmers. Based on whether the user is in the list of unblinded team members or not, setup switches between the dummy and actual randomization files and blinded and unblinded (restricted) folder automatically. With this set up raw data and programs remain in the blinded area. Only OUTPUTs (datasets and TLFs) would be generated in unblinded area.

Global macro variables:

- A macro variable to determines deliverable type: ex: ***_anltype***
 - LIBNAMEs are customized based on the value of this variable.
 - The value for this variable can be specified prior to running the set-up file in individual programs or in a batch program.
- A macro variable referring to location of randomization fileif it is not a dataset.
 - This depends on SYSUSERID.
- A macro variable referring to randomization file.
 - This depends on SYSUSERID.

LIBNAMEs: Regardless of type of deliverable and blinded or unblinded location, the libref referring to below locations in any analysis will always be same; actual location changes based on the value of ***_anltype***:

- raw data (other than unblinding data such as randomization file)
 - all raw data should be copied into individual deliverable folders depending on the type of deliverable being worked on.
- live data in data management folder (mostly to be accessed for checking data issues/resolution)
- SDTM datasets
- QC SDTM datasets
- ADaM datasets/ non-CDISC analysis datasets
- QC ADaM datasets
- For blinded studies: libref for location of randomization file if it is a dataset.



when ***_anltype***=CSR then libref RAW refers to study/rawdata/CSR

when ***_anltype***=DMC then libref RAW refers to study/rawdata/DMC etc.

Avoid physical path names or additional LIBNAMEs in other programs. Customizations for additional analyses can be added in the setup file based on study needs.

Sample code in setup file

```
/*Customization for CSR*/
%if upcase(&_anltype)=CSR %then %do;
  /*Librefs for raw, SDTM and ADaM datasets*/
  libname raw "&drive\product\study\rawdata\CSR";
  libname s "&drive\Product\Study\production\datasets\SDTM\CSR";
  libname qcsdtm "&drive\Product\Study\validation\datasets\SDTM\CSR";
  libname a "&drive\Product\Study\production\datasets\ADAM\CSR";
  libname qcadam "&drive\Product\Study\validation\datasets\SDTM\CSR";
  /*locations*/
  %let tlfprg="&drive\Product\Study\production\programs\TLF\CSR";
  %let tlfout="&drive\Desktop\Product\Study\production\output\CSR";
  %let qctlfprg="&drive\Product\Study\validation\programs\TLF\CSR";

  %let dcutoff=ddmmyy; /*provide data extract date*/
%end;

/*specify the unblinded team member user IDs*/
%let unbltm=&sysuserid=Zxxx or &sysuserid=Yxxxxx or &sysuserid=Xyyy;

/*Determine use of actual or dummy randomization file*/
%if &unbltm %then %do; /*all outputs will be directed to unblinded area*/
  %let randf=unblrndf; /*Name of unblinded randomization file*/
  %let rand=unblfold; /*Location of unblinded randomization data*/;
%end;
%else %do; /*all outputs will be directed to blinded area*/
  %let randf=blrndf; /*Name of blinded randomization file*/
  %let rand=blfold; /*Location of blinded randomization data*/;
%end;
libname rand "&rand." access=readonly; /*Libname for unblinded or blinded randomization data based on previous step*/
```

In individual or batch program prior to running the setup file ***_anltype*** should be specified to run customization specific to that analysis/deliverable:

```
%let _anltype=CSR;
%include "path to set up file\setup.sas";
```

CONCLUSION

Being able to run programs based on different source data simultaneously and able to work on parallel deliverables with no confusion is essential to meet demands in clinical study programming. This can be achieved efficiently with use of standard folder structure and master environment setup file.