

One Project, Two Teams: The Unblind Leading the Blind

Kristen Reece Harrington, Rho, Inc.

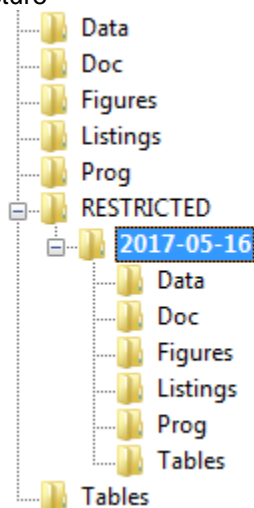
ABSTRACT

In the pharmaceutical world, there are instances where multiple independent programming teams exist to ensure blinded treatments are maintained by the appropriate parties. For the purposes of this discussion, blinded corresponds to fake data and unblinded corresponds to actual data. Within these projects, blinded programmers use temporary fake data to create programs which produce both blinded and unblinded results. To ensure the blind is maintained and ethics are upheld, only the blinded programming team produces and modifies the programs. While robust programming is key, another major contributing factor for success is communication. This discussion will explore the process of initializing a project supported by blinded and unblinded teams, successful communication techniques when real data comes into play, and ways to effectively troubleshoot validation issues without providing unblinding information.

INTRODUCTION AND GENERAL ASSUMPTIONS

This discussion focuses on studies requiring both blinded and unblinded deliverables produced by the same vendor. Typically, this dual team approach is required for safety review meetings (ex. DSMB) where both blinded and unblinded reports are delivered during the course of the study. To facilitate this discussion, let's agree on the following assumptions:

- The work supports a DSMB
- Programs are on one network, rather than multiple servers
 - Main directory will contain blinded data
 - RESTRICTED directory will contain unblinded data
 - IT restricts access to the RESTRICTED directory
 - The RESTRICTED directory is nested within the Main directory. Both directories have same structure



- All programs utilize a single setup program and switch macros
- There are two teams
 - Blinded Team hence forth referred to as Team B
 - Unblinded Team hence forth referred to as Team U
- Team B writes and maintains all programs
- Team U's programs are simple %includes of Team B's programs
- Unblinding occurs at the SDTM level
- UAT data are available for initial programming and study setup

TEAM PLANNING

The initial step for success is team planning. Appropriate staffing is essential. Team B should have a least one biostatistician and two programmers. The blinded programmers should have strong macro experience and robust programming skills. The unblinded team should have at least one programmer and one biostatistician. While the unblinded programmer will not be modifying any code for ethical reasons, this programmer should have exceptional communication and validation skills. We'll address these skill sets further along in our discussion.

Team B is responsible for all programming activities, including creation of the setup program and switch macros. The setup program consists of all appropriate libnames, SAS® options for automatic inclusions and all appropriate output paths. All paths within the setup program should contain macro variables specified within the Switch macro. The Switch macro should be a very simple macro program containing a minimum of two macro variables. The first macro variable specifies what type of data are used. The second specifies the paths. Additionally, these macro variables can be used to efficiently macrotize the treatment columns within displays.

Basic good programming practice dictates that macro programs should contain well documented header blocks. This is extremely true for the Switch macro. Furthermore, additional documentation within the Switch macro is strongly encouraged.

Example:

```

/*-----*

PROJECT:      Specify your study

MACRO:        Switch.sas

PURPOSE:      SWITCH BETWEEN DSMB OPEN AND CLOSED DISPLAYS

ARGUMENTS:    1st => <FAKE, DSMB> FAKE=scrambled codes,DSMB=real codes>
              2nd => <Date of restricted directory for DSMB YYYY-MM-DD>

RETURNS:      GLOBAL MACRO VARIABLES:
              SwitchDir => DSBM Subdirectory (e.g. \RESTRICTED\<DATE>)
                      Includes leading \ for easy substitution
                      into path specification
              %*Switch(FAKE,); *Goes to Main study folders;
              %*Switch(DSMB,2017-05-17); *Goes to Restricted folders;

USAGE:        Called by %Setup_TLF

PROGRAM HISTORY:

DATE          PROGRAMMER          DESCRIPTION
-----
01May2017    KHarrington          created
*-----*/

%macro switch(BLIND,Date_);
  %GLOBAL Switch SwitchDir TrtLb11 TrtLb12 ;
  %let SWITCH = %upcase(&BLIND);
  /*EXECUTE THIS CODE WITH DUMMY DATA*/
  %if %upcase(&BLIND) = FAKE %then %do;
    %let SwitchDir = \RESTRICTED\&Date_;
    %let TrtLb11 = Group A;
    %let TrtLb12 = Group B;
  %end;

```

```
/*EXECUTE THIS CODE WITH REAL TREATMENTS*/
%if %upcase(&BLIND) = DSMB %then %do;
    %let SwitchDir = ;
    %let TrtLb11 = Control;
    %let TrtLb12 = Actual Treatment Label;
%end;
%mend switch;
%*Switch(FAKE,); /*Goes to Main Study Folder*/
%Switch(DSMB,2017-05-16); /*Goes to Restricted Directory*/
```

The Switch macro is included using an %include statement within the Setup macro program. The &SwitchDir macro variable created in the Switch macro should be utilized for all path specifications, such that the appropriate libnames are utilized and all files are written out to the appropriate locations. All SDTM, ADaM, Table, Listing, and Figure (TLF) programs will include the Setup macro by way of an %include statement.

PREPARATION – PRE-DATA

Once both teams are identified, it is important to meet internally to discuss the key elements of the study prior to receipt of any data. The entire team should know exactly what data are blinded, and consequently what data have unblinding potential.

Potential unblinding items can be domain specific, as well as analysis specific. For example, the SDTM domains EX and EC are typically blinded as they contain dosing information. Usually, this means that these domains are empty on the blinded side. Robust programming is required to ensure a smooth transition from blinded to unblinded when working with these types of data.

Some statistical analyses are population specific; therefore, they too have unblinding potential. For example, Fisher's exact test may be used if the population of either treatment group is below five patients. Logistic Regression may be used when the population of either treatment group is above five patients. This information is important for the blinded programmers; however, it is imperative to the unblinded programmer as it directly relates to how non-validating items are communicated as well as ensuring the appropriate analyses are being executed.

Both teams should understand the requirements for the deliverables. Do the blinded displays only contain a Total column? Will they have the same number of treatments as the unblinded displays? Will they contain fewer treatments? This information is critical for both robust programming, and for QC.

Both teams should be well versed not only in the method of running both the blinded and unblinded programs, but in the overall process, including locations of required files. As previously agreed upon, unblinding occurs at the SDTM level. This requires macro coding to ensure the appropriate treatments are assigned for both the blinded and unblinded program runs. Utilizing the Switch macro is the easiest approach. The blinded programmer needs to understand this process to ensure robust programming. It is important for the unblinded programmer to be familiar with this process to aid in communicating validation discrepancies.

At this point in planning, all individuals should be aware of the first deliverable timeline. It is crucial to set internal timelines for both teams. These internal timelines should allow sufficient time for several rounds of program execution for both teams. As you begin a study, it is important to be mindful of the full SDTM through TLF run time on the blinded side. This will help you set internal timelines that allow for program execution, log cleaning and validation investigation.

Adequate testing of your macro code or macro variables that are specific to treatment codes and directory paths should be done prior to IT restricting any directories. Dummy data, or UAT data, should be utilized to run simple programs. Checks that logs and output are directed to the appropriate locations dependent on the macros should be thoroughly vetted. This ensures that the basic concepts of your macro code are executing appropriately, and allows both teams to focus on the data once it arrives. Once IT restricts the directories, this process should be repeated to ensure all programs are still working as expected. Once this is verified, the team is ready for data!

POST-DATA

As programming with the clinical data begins, it is important to run tests between the Team B and Team U. While Trial Design domains are typically the first programs developed, the critical starting point for testing is the SDTM domain DM. Once the production DM domain and validation programs are stable within Team B, Team U should run the programs to ensure the Switch macro and subsequent macro variables are working as expected. This is also the best time to test that the handling of the blinded vs unblinded treatments is correct. This can be done by the Team U programmer utilizing a PROC FREQ on both the blinded DM dataset and the unblinded DM dataset and comparing the results. The results from the unblinded DM dataset should then be cross checked against the randomization data. The Team B programmer should do similar cross checks against the SAP to ensure the fake data has the expected treatment distribution. These cross checks should be independently vetted by the Biostatisticians from both teams as well.

Once SDTM programming and validation are complete within Team B, the programs should be run by Team U. Ensuring that SDTM domains validate within both teams prior to initializing ADaM programming ensures smoother transitions throughout the project. This process should be repeated once ADaM programming is complete within Team B, prior to TLF programming, as well.

VALIDATION COMMUNICATION

Communication is critical to every project. For projects with blinded and unblinded teams, appropriate communication is absolutely necessary. Ensuring that no unblinding information is relayed is of the utmost importance. As Team U cannot modify programs, it is imperative for the Team U programmer to refer back to the discussion regarding potentially unblinding domains/data before beginning to relay validation discrepancies to Team B.

At the beginning of a study, when data are sparse, communicating validation discrepancies without inadvertently unblinding can be tricky. This is where Team U's effective communication and strong validation skills come into play.

With some studies, simply communicating that the production output has values where the validation output is completely empty can be unblinding due to study critical subsets. Knowing your study and your data facilitates maintaining the blind. In these types of instances, the Team U programmer must lead the Team B programmers to the correct updates without providing any unblinding information. At times the simple answer is for Team B programmers to check their subsets, while at other times the answer is more complex.

In more complex scenarios, reviewing the program logs may lead the Team U programmer to some clues as to where the discrepancies are occurring. There may be instances where different model statements are being used between production and validation, or an incorrect analysis is performed based on sample size. These discrepancies are difficult to communicate without potentially unblinding the team. Asking Team B to add notes that output to the log specifying which population dependent analyses are executing is a helpful check for the blinded programmers to note whether they've added dependent code without providing unblinding information.

Example:

```
PROC SQL NOPRINT;
    Select count(*) from AnalysisDS into: trtobs1 where trt = 1;
    Select count(*) from AnalysisDS into: trtobs2 where trt = 2;
QUIT;
%if 0 < &TrtObs1 < 5 or 0 <&TrtObs2 < 5 %then %do;
    %put PERFORMING FISHERS EXACT BASED ON &TrtObs1 AND &TrtObs2 ;
    <additional code>
%end;
```

```
%if &TrtObs1 > 5 and &TrtObs2 > 5 %then %do;  
  %put PERFORMING LOGISTIC REGRESSION BASED &TrtObs1 AND &TrtObs2;  
  <additional code>  
%end;
```

By checking the log of a program containing notes similar to the above example, one can easily determine whether the appropriate analysis was performed.

CONCLUSION

Prior to the first data transfer, targeted planning needs to occur. Staffing, communication plans, identification of potentially unblinding material, clearly documented macros, and internal timelines need to be thoroughly discussed to ensure success throughout the project. After the initial data transfer, additional testing should take place to ensure the appropriate foundation has been laid. Once the programs are in place for both teams, the focus shifts to validation. As Team U cannot modify programs, communicating discrepancies between Teams B and U must be done with full knowledge of potentially unblinding data in order to maintain the blind. This is where Team U leads Team B.

Communication and strategic planning significantly impact the success of a study. Resourcing Team B with programmers who can make programmatic updates based off of very little instruction is essential. Resourcing Team U with individuals who have exceptional communication and data investigation skills is vital. Planning and communicating all aspects of programming ensures success.

CONTACT INFORMATION

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

Kristen Reece Harrington
Rho, Inc.
919-595-6377
Kristen_Harrington@rhoworld.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks 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