PharmaSUG 2013 - Paper AD17

Automated Validation of Third Party Data Imports

Pranav Soanker, PPD Inc., Austin, Texas

ABSTRACT:

When studies are outsourced to a Contract Research Organization (CRO) by a pharmaceutical company (client) they can include the requirement of using a third party database to configure and report study data. Once a study starts, in-stream changes to the structure of these databases can impact program functionality and/or create delays in development, either of which can be time and resource intensive. Therefore at the start of the study it is critical to have a consensus between the third party vendor and the CRO on the database structure to ensure imports throughout the study are consistent. After an initial agreement is reached a test transfer is typically performed to validate the import against the agreed upon specifications. Previously this validation was a manual process that, though critical, was cumbersome and required differences to be noted and compiled separately. The method described within this paper provides an automated process that can be used at study start-up for initial validation, as well as with each new import, to identify mismatches between what was expected and what was received. This process utilizes a dataset that holds the structural specifications and compares against the data imported, which can be in SAS datasets, transport files, ASCII files, or Excel spreadsheets. The process results in reports that identify the differences and can be used to clearly communicate them to the vendor.

KEYWORDS:

CRO (Contract Research Organization), Sponsor (usually the Client/Research Agency) Vendor (any company authorized by client who may provide the data), Third Party Database, Data Imports, Validation, Automation, FTP Server, Sponsor-Defined Controlled Terminology (CT).

INTRODUCTION:

This paper demonstrates a method to validate the database (received over FTP Server) against the Sponsor provided Data Specifications Table (DST). For studies that don't have an in-house database system set-up, the Sponsor provides the specifications that provide a blue print for database structure that will be received over FTP Server from Vendor. There may be multiple versions of DST sent depending upon the current phase of study (example, on test data/before FPFV, or on entering Phase 2 of a study, or LPLV/DB lock etc). This process helps the CRO to identify any irregularities over the data received from Sponsor / Vendor and also can help estimate the scope of re-work if there is an in-stream change in database structure. Additionally before the programmer considers the DST and Database final, consistency checks for data with CT should be performed and any issues identified which aren't from expected CT should be documented and brought to the attention of Sponsor / Vendor for a possible data query.

PROCESS OVERVIEW:

The entire process of validation is conveniently divided into the below sections

1. Read in DST and create Individual Datasets with Metadata.

2. Read in Datasets from the Import Database (Data received over FTP from Vendor)

3. (a) Match Merge and create Discrepancy reports.

   (b) Use PROC COMPARE to generate domain level specific Discrepancy reports.

4. Validation Checks for Sponsor-Defined Controlled Terminology.

MECHANISM:

Section 1 explains the process of importing the DST and creates Individual Datasets with Metadata, %rd_XL macro is created which reads in the spreadsheets and creates individual datasets which stores the Metadata of the available domains/datasets. Below is the excerpt from the DST (Plate 1).

Variable	Type	Len	Label	Format
STUDYID	$	15	Study Identifier	
SUBJID	$	7	Unique Subject Identifier	siteid- scrnid
SITEID	$	3	Investigator Identifier	
SCRNID	$	3	Screening Identifier	
RANDID	$	4	Subject Identifier for the Study	
SUBJINIT	$	3	Subject Initials	
AEYN	$	1	Subject Experienced Any Adverse Events	$YN.
AESPID	$	3	Sponsor-Defined Identifier	
AETERM	$	200	Reported Term for the Adverse Event	
DICTIONARY	$	30	Dictionary Name	
VERSION	$	10	Dictionary Version	
SOC_TEXT	$	200	MedDRA System Organ Class	
PT_TEXT	$	200	MedDRA Preferred Term	
LLT_TEXT	$	200	MedDRA Lowest Level Term	
HLT_TEXT	$	200	MedDRA High Level Term	
HLGT_TEXT	$	200	MedDRA High Level Group Term	
VERBATIM	$	200	Verbatim term used for coding	
COMMENTS	$	200	Coding Comments	
AESTDT	$	10	Start Date of Adverse Event	yyyy-mm-dd
AESTDTN	N		Start Date of Adverse Event	Date9.
AEENDT	$	10	End Date of Adverse Event	yyyy-mm-dd
AEENDTN	N		End Date of Adverse Event	Date9.
AEONGO	$	1	Adverse Event Ongoing	$YN.
AEREL	$	40	Causality	
AESTART	$	1	Event started before first dose of study drug	$YN.
AESER	$	1	Serious Event	$YN.

(Plate 1: Excerpt from DST for AE Domain)

%rd_XL macro uses the sashelp.VTABLE & sashelp.VCOLUMN dictionaries to determine the available domains from the DST.

```
options nofmterr;

PROC DATASETS lib=work memtype=data nolist kill;
QUIT;

PROC DATASETS lib=TRANS memtype=data nolist kill;
QUIT;

** Set up Study Paths **;
%let trunk=%str(<set study path>);
%let extr=%str(<set study path>);
%let imprt=%str(<set Import Data path>);

** Set up Name of the DST Document **;
%let dst=%str(study XXX DST_CRF.xls);
%let rundate=%SYSFUNC(TODAY(),date9.);
%let tm=%SYSFUNC(TIME(), tod8.);

DATA _null_;
 CALL SYMPUT ("runtime", strip("T"||tranwrd("&tm.",":","-")));
RUN;

** Set up Output File Names **;

%let file1=%str(NOT_DST_&rundate.&runtime..xls);
%let file2=%str(NOT_IMPORT_&rundate.&runtime..xls);

** %rd_XL Macro to Read in DST and create Datasets **;

%macro rd_XL;

LIBNAME EXL "&trunk.\Documents\Clinical and DM\&dst." access=readonly;

       PROC SQL noprint;
        select count(distinct(MEMNAME)) into: total
          from sashelp.VTABLE
          where LIBNAME ='EXL' & index(strip(MEMNAME),'General')=0;
        select distinct(compress(MEMNAME,"',$")) into: s1 - :s%trim(%left(&total))
          from sashelp.VTABLE
          where LIBNAME ='EXL' & index(strip(MEMNAME),'General')=0;
       select distinct(MEMNAME) into: v1 - :v%trim (%left(&total))
          from sashelp.VTABLE
          where LIBNAME ='EXL' & index(strip(MEMNAME),'General')=0;
       select distinct(compress(MEMNAME,"',$,-")) into: c1 - :c%trim(%left(&total))
          from sashelp.VTABLE
          where LIBNAME ='EXL' & index(strip(MEMNAME),'General')=0;
       QUIT;

%do i=1 %to &total;
       PROC SQL noprint;
        select COUNT(distinct(NAME)) into: T
         from sashelp.VCOLUMN
          where LIBNAME ='EXL' & MEMNAME="&&v&i." & SUBSTR(NAME,1,1)^='F';
       select distinct(NAME) into: O1 - :O%trim(%left(&T))
          from sashelp.VCOLUMN
           where LIBNAME ='EXL' & MEMNAME="&&v&i." & SUBSTR(NAME,1,1)^='F';
       QUIT;

       DATA &&c&i.;
        length VARIABLE MEMNAME $32. LIBNAME $8.;
         set EXL."&&s&i.$"n;
          MEMNAME="&&s&i.";
          LIBNAME="DST";
```

```
       RUN;

       PROC SORT data=&&c&i. out=&&c&i.;
        by VARIABLE;
       RUN;
%end;
%mend rd_XL;

%rd_XL;
```

Section 2 details the process of Reading in all the datasets from Import Database -- The Contents procedure stores the contents of all the datasets from the Import library into a SAS Dataset (Import.sas7bdat).

```
       PROC CONTENTS data=IMPORT._all_
                     out= IMPORT(keep=LIBNAME NAME MEMNAME TYPE LENGTH LABEL Format);
       RUN;

       DATA IMPORT (keep=LIBNAME NAME MEMNAME TYPE LENGTH LABEL Format);
        set IMPORT (rename=(type=typen));
        type=strip(put(typen,best.));
        name=upcase(strip(NAME));
       RUN;
```

Section 3(a) details the Match Merge and to create Discrepancy Reports – In this step the two datasets created from DST and Import viz. DST & IMPORT are merged and three datasets Common, not_DST and not_IMPORT are created.

```
       PROC SORT data=DST;
        by MEMNAME name;
       RUN;

       PROC SORT data= IMPORT;
        by MEMNAME name;
       RUN;

       DATA common
            not_DST not_IMPORT;
        merge DST(in=in1) import(in=in2);
        by MEMNAME name;
         if in1 & in2 then output common;
         if ~in1 & in2 then output not_DST;
         if in1 & ~in2 then output not_IMPORT;
       RUN;
```

** Create Discrepancy Reports for Variables not in DST & not in IMPORT **

```
        PROC EXPORT data=Not_dst
         outfile="&trunk.&extr.&file1." dbms=excel replace;
         sheet="NOT_DST_IN_IMPORT";
        RUN;

        PROC EXPORT data=Not_import
         outfile="&trunk.&extr.&file2." dbms=excel replace;
         sheet='NOT_IMPORT_IN_DST';
        RUN;
```

Section 3(b) details the generation of Domain level specific Discrepancy reports using **%rd_IMP** and **%rd_CMP** macros,

```
%macro rd_imp;

LIBNAME trans "&trunk.\Databases\Transformed\Output";

    PROC SQL noprint;
        select count(distinct(MEMNAME)) into: imptot
          from sashelp.VTABLE
          where upcase(strip(LIBNAME)) ='IMPORT';

       select distinct(compress(MEMNAME,"',$")) into: s1 - :s%trim(%left(&imptot))
          from sashelp.VTABLE
          where upcase(strip(LIBNAME)) ='IMPORT';
       select distinct(MEMNAME) into: v1 - :v%trim (%left(&imptot))
          from sashelp.VTABLE
          where upcase(strip(LIBNAME)) ='IMPORT';
       select distinct(compress(MEMNAME,"',$,-")) into: c1 - :c%trim(%left(&imptot))
          from sashelp.VTABLE
          where upcase(strip(LIBNAME)) ='IMPORT';
    QUIT;

    %do i=1 %to &imptot;

      PROC SQL noprint;
       select COUNT(distinct(NAME)) into: T
        from sashelp.VCOLUMN
         where upcase(strip(LIBNAME)) ='IMPORT'
               & MEMNAME="&&v&i." &  SUBSTR(NAME,1,1)^='F';
       select distinct(NAME) into: O1 - :O%trim(%left(&T))
         from sashelp.VCOLUMN
          where upcase(strip(LIBNAME)) ='IMPORT'
               & MEMNAME="&&v&i." & SUBSTR(NAME,1,1)^='F';
      QUIT;

       PROC CONTENTS data=IMPORT."&&s&i."n out=trans.&&c&i. noprint;
       RUN;

       DATA trans.&&c&i.(drop=NAME);
        length VARIABLE MEMNAME $32. LIBNAME $8.;
         set trans.&&c&i.;
          MEMNAME="&&s&i.";
          LIBNAME="IMPORT";
          VARIABLE=strip(NAME);
       RUN;

       PROC SORT data=trans.&&c&i. out=trans.&&c&i.;
        by VARIABLE;
       RUN;

%end;

%mend rd_imp;

%rd_imp;
```

```
** For Validation, only DST available domains are considered **;

 PROC SQL noprint;
  select distinct(MEMNAME)
   into :dstlist separated by ' '
    from dictionary.columns
     where LIBNAME='WORK'; ** All DST Only in Work Library**;
 QUIT;

%macro rd_cmp;

 %let slc=%eval(%SYSFUNC(count(%cmpres(&dstlist.),%str( ))+1);

  %do i=1 %to &slc.;

   %let compdst=%SYSFUNC(scan(&dstlist,&i,' '));

   PROC COMPARE base=&compdst. compare=TRANS.&compdst.
                out=test_&compdst. outbase outcomp outdif noprint;
   RUN;

  PROC EXPORT data=test_&compdst.
   outfile="&trunk.&extr.Recon_&compdst._&rundate.&runtime..xls."
   dbms=excel replace;
   sheet="DIFF_in_&compdst";
  RUN;


%end;

%mend rd_cmp;

%rd_cmp;
```

Section 4 explains the Sponsor-Defined Controlled Terminology Checks, the sponsor provides the expected Controlled Terminology and associates the identified variable with a format, a catalogue of all formats are created using FORMAT procedure.

```
proc format cntlout=Extract.formats;

  value $YN

   'Yes'   =   'Y'

   'No'   =   'N'

   ;

RUN;
```

The formats are then linked up with the corresponding variables and a list of variables with CT formats can be created and consistency checks can be done using a simple Proc FREQ.

SAMPLE REPORTS:

Summary Reports for list of Discrepancies found DST versus Import Data comparison

NOT_IMPORT_01APR2013T02-12-48.xls	18 KB	Microsoft Excel Wor...	4/1/2013 2:12 AM
NOT_DST_01APR2013T02-12-48.xls	157 KB	Microsoft Excel Wor...	4/1/2013 2:12 AM
NOT_IMPORT_09JAN2013T16-47-38.xls	18 KB	Microsoft Excel Wor...	1/9/2013 5:47 PM
NOT_DST_09JAN2013T16-47-38.xls	10 KB	Microsoft Excel Wor...	1/9/2013 5:47 PM
NOT_IMPORT_09JAN2013T16-23-16.xls	22 KB	Microsoft Excel Wor...	1/9/2013 5:23 PM
NOT_DST_09JAN2013T16-23-16.xls	15 KB	Microsoft Excel Wor...	1/9/2013 5:23 PM
NOT_IMPORT_18DEC2012T12-02-02.xls	22 KB	Microsoft Excel Wor...	12/18/2012 1:02 PM
NOT_DST_18DEC2012T12-02-02.xls	15 KB	Microsoft Excel Wor...	12/18/2012 1:02 PM
NOT_IMPORT_18DEC2012T11-46-20.xls	22 KB	Microsoft Excel Wor...	12/18/2012 12:46 PM
NOT_DST_18DEC2012T11-46-20.xls	15 KB	Microsoft Excel Wor...	12/18/2012 12:46 PM

Domain level specific Discrepancy reports for DST versus Import Data comparison

Recon_DM_01APR2013T03-54-16.xls.XLS	21 KB	Microsoft Excel Wor...	4/1/2013 3:54 AM
Recon_DA_01APR2013T03-54-16.xls.XLS	22 KB	Microsoft Excel Wor...	4/1/2013 3:54 AM
Recon_CONTACT_01APR2013T03-54-16.xls.XLS	19 KB	Microsoft Excel Wor...	4/1/2013 3:54 AM
Recon_CM_01APR2013T03-54-16.xls.XLS	30 KB	Microsoft Excel Wor...	4/1/2013 3:54 AM
Recon_CGI_01APR2013T03-54-16.xls.XLS	17 KB	Microsoft Excel Wor...	4/1/2013 3:54 AM

CONCLUSION:

For various milestones in a study it is imperative that the database has to be checked for consistency and any changes/deviations must be communicated to Sponsor / Vendor. Any changes in the database that are approved at a later stage(s) are identified and the development program (s) functionality risk and scope of re-work can be assessed as a result of change(s). This method described in this paper provides a scope to validate the database and send the discrepancy reports to sponsor before considering the database as 'Final'. Additionally these discrepancy reports provide documented proof of any data irregularities and also help to track changes in the database.

ACKNOWLEDGEMENT:

Thanks to *Jeanina (Nina) Worden & David Gray* for reviewing my paper and providing their valuable input(s). Thanks to my wife *Sravani* for motivating me to complete this paper and my daughter *Likitha* who kept me awake late nights.

CONTACT INFORMATION:

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

Pranav Soanker, M.S.

PPD Inc,

7551 Metro Center Drive, Suite 300, Austin, Texas 78744

Phone: +1 (512) 747 5061

Email: Pranav.Soanker@ppdi.com