Paper 175-2012

Using SAS® Macros to Remediate Existing SDTM Data Sets for New Drug Application (NDA) Submission

Yanwei Han, Vertex Pharmaceuticals, Inc., Cambridge, MA, USA

ABSTRACT

This paper presents the development of macros to prepare existing SDTM data sets for submission within a New Drug Application (NDA). Within the scope of this process was (1) the late-stage checking of SDTM data sets under very tight timelines, and (2) the creation of a single SAS program to make changes to the SDTM data sets for every study within the submission.

INTRODUCTION

We created 21 SDTM submission publishing packages within one NDA from November 2010 to July 2011, including 13 phase I studies, 4 phase II studies and 4 Phase III studies.

Each SDTM submission package included (1) SDTM annotated Case Report Form (CRF), (2) Reviewer Guide (reviewer_guide.pdf), (3) define.xml, and (4) SDTM data sets

For this submission we had variety of states of SDTM data sets. For example, some data sets were no-existent; some were incomplete. The 21 studies were created in a span of more than four and a half years. The earliest study SDTM data sets were created in December 2006. At that time, we were learning about what SDTM data sets were. The latest SDTM data sets were finalized in June 2011.

For most phase I studies, SDTM data sets were outsourced and created by different vendors. We found different types of errors and warnings in the studies when we used WebSDM (Validation Checks Performed by WebSDM^{TM}) to check data. Moreover, we did not get SDTM data set programs from vendors.

We had already completed the Clinical Study Reports, so the primary challenge was to change the form of the data to comply requirements without changing any data elements that would impact the TFL programs.

STRATEGY

The following considerations were important in determining our data strategy:

- In order to create NDA submission packages, we needed to ensure that SDTM data sets consistently follow the CDISC SDTM Implementation Guide version 3.1.1 in all 21 studies.
- Due to time restrictions, we would create one program with modifications per study.
- In each study program, we would create macros that took parameters, so that we could run all data sets through the macros in most case, modifying only the parameters for each study.

PROCESS

STEP 1: CHECK SDTM DATA SETS

We Performed WebSDM SDTM data sets check. Our first step in dealing with WebSDM finding was to detect the precise cause of each validation error and to determine whether there was a true data problem or a possible SDTM compliance issue. Any actual data issues, for example “Start date expected when end date provided”, “missing unit on lab value”, were documented in a Reviewer Guide that was added as part of the submission package. SDTM data sets compliance issues, for instance, “country ‘US’ is not found in controlled terminology” would be changed in STEP 2: CLEAN UP SDTM DATA SETS.

We checked SDTM data sets based on CDISC SDTM Implementation Guide and SDTM specification. For example Finding domain - -TEST length should be equal to or less than 40 characters, but in some Phase I studies, it was 80 characters. This issue was changed in STEP 2 also.

We also cross checked SDTM data sets with SDTM annotated CRF and define.xml. For example, we made sure column “Origin” value in define.xml matched variable ‘QORIG’ value in SUPPQUAL. If the value was ‘CRF’ in define.xml then this variable can be found in annotated CRF.

STEP 2: CLEAN UP SDTM DATA SETS

Create two macros “suppqual” and “sdtm”. The macros allowed efficient conversion of all studies.

1. **Modify Supplemental Qualifier data sets. For example, data sets labels and required variable QORIG were not in SUPPQUAL in most phase I studies.**

Program Part 1 Added suppqual data set labels or required variable QORIG if needed; deleted Format, Informat; changed variable value or labels if needed

```
libname rawsdtm “…original SDTM data sets folder…”;
libname sdtm “… updated SDTM data sets folder…”;

%MACRO suppqual(data=, needqori=, qorig=, mod=, modify=);
       DATA sdtm.supp&data (LABEL="Supplemental Qualifiers for &data");
            SET rawsdtm.supp&data;

       /* add QORIG - required variable*/;
           %IF &needqori=1 %THEN %DO;
               LENGTH QORIG $20;
               &qorig
               LABEL QORIG='Origin'
           %END;

       /* make sure format and informat are all blank*/
            FORMAT _all_;
            INFORMAT _all_;

       /* change variable value or label if needed*/
            %IF &mod=1 %THEN %DO;
                &modify;
            %END;
        RUN;
%MEND;
```

Call macro suppqual, for example

```
%suppqual(data=AE, needqori=1,  qorig=%str(QORIG="ASSIGNED";),    mod=,  modify=)
              /* add variable QORIG: MedDRA information was collected in SUPPAE
                 and their origin were ‘ASSIGNED’ */

%suppqual(data=CM, needqori=1,  qorig=%str(if QNAM in ('CONTINUE') then
          QORIG='CRF'; else QORIG="ASSIGNED";),  mod=,  modify=)
              /* add variable QORIG: WHODD information was collected in SUPPCM and
                 origin were ‘ASSIGNED’; ‘CONTINUE’ was from CRF */

%suppqual (data=DM, needqori=1, qorig=%str(QORIG='CRF';),
           mod=1, modify=%str(label RDOMAIN='Related Domain Abbreviation';))
              /* add variable QORIG: all SUPPDM Qualifier Variables were collected
                 from CRF; add label for variable RDOMAIN  */

%suppqual (data=EG,  needqori=,  qorig=,  mod=1,  modify=%str(if QORIG=’CRF’ then
           QORIG=’eDT’;))
              /* modify existing variable QORIG from “CRF” to “eDT” */
```

2 Modify SDTM data sets

Program Part 2 Make sure Finding domain - -TEST value equal or less than 40 characters; deleted
Format, Informat; changed variable value or labels if needed

```
%MACRO sdtm (data=, num=, label=, mod=, modify=);

       /* choose the data set that need to change –TEST length*/
       %IF &num=1 %THEN %DO;

       /* create macro to re-order data set variables, since change variable
          --TEST length to 40 characters, --TEST will be the first variable in data
          set. QC program to compare new SDTM Finding data set with original data
          set to ensure no truncated value after variable length was changed*/

        PROC SQL NOPRINT;
             SELECT name INTO : ORDER SEPARATED BY ', '
              FROM SASHELP.VCOLUMN
              WHERE LIBNAME='RAWSDTM' AND MEMNAME="&data";
        QUIT;

        DATA leng&data;
            LENGTH &data.test $40;
            SET rawsdtm.&data;
        RUN;

        /* re-order data set */
            PROC SQL NOPRINT;
                      CREATE TABLE &data AS
                      SELECT &order
             FROM leng&data;
             QUIT;
        %END;

         /* read data directly if no --TEST length change */
         %ELSE %DO;
         DATA &data;
              SET rawsdtm.&data;
         RUN;
         %END;

         DATA sdtm.&data (LABEL=&label);
              SET &data;

         /* make sure format and informat are all blank*/
              FORMAT _all_;
              INFORMAT _all_;

         /* change variable value, label, or delete variables*/
               %IF &mod=1 %THEN %DO;
                   &modify;
                %END;
          RUN;
%MEND;
```

Call macro sdtm, for example

```
%sdtm(data=EX, num=, label=%str(Exposure), mod=1, modify=%str(drop EXSTAT EXREASND;))
          /* EX should only contain medications received. Drop missing value
             variables EXSTAT EXREASND */
```

```
%sdtm(data=TI, num=, label=%str(Trial Inclusion/Exclusion Criteria), mod=1,
       modify=%str(drop IETEST1 IETEST2; if IE...IETEST=’... meaningful text”;))
         /* criterion test is >200 characters, put meaningful text in IETEST,
            delete IETEST1 IETEST2 and describe the full text in the study metadata
            in define.xml*/
```

3 Modify SDTM without macro

Program Part 3 Modify data sets without macro. For example, visit number 8000 was not included in data set SV but was in data set EG, add this visit to SV.

```
/* Read visit number=8000 from data set EG*/
DATA eg;
   SET sdtm.eg;
   LENGTH SVSTDTC SVENDTC $20;
   IF visitnum=8000;
      DOMAIN="SV";
      SVSTDTC=substr(EGDTC,1,10);
      SVENDTC=SVSTDTC;
   KEEP STUDYID DOMAIN USUBJID VISITNUM VISIT SVSTDTC SVENDTC;
RUN;

PROC SORT NODUPKEY; BY STUDYID DOMAIN USUBJID VISITNUM VISIT SVSTDTC SVENDTC;
RUN;

/* Add this visit to data set SV*/

DATA sdtm.sv(label=Subject Visits);
   SET sdtm.sv eg;
RUN;
PROC SORT; BY STUDYID DOMAIN USUBJID VISITNUM VISIT SVSTDTC SVENDTC;
RUN;
```

CONCLUSION

The strategy and modify SDTM data sets program gave us an efficient, low risk basis to be able to provide the required SDTM data sets deliverables within a limited timeframe.

ACKNOWLEDGMENTS

I would like to acknowledge Lynn Anderson, Director of Statistical Programming, Vertex Pharmaceuticals, Inc. who led the team in the statistical programming for this drug program, the submission packages that got FDA approval within 4 months and for her support in the writing of this paper.

CONTACT INFORMATION

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

Name: Yanwei Han
Enterprise: Vertex Pharmaceuticals, Inc
Address: 130 Waverly Street
City, State ZIP: Cambridge, MA 02139
Work Phone: 617-444-6736
E-mail: yanwei_han@vrtx.com
Web: http://www.vrtx.com/