Confessions of a Clinical Programmer: Dragging and Dropping Means Never Having to Say You're Sorry When Creating SDTM Domains

Janet Stuelpner, SAS Institute, Inc.
Jack Shostak, Duke Clinical Research Institute

ABSTRACT

We have been clinical programmers for many years. Our first instinct has always been to write the SAS[®] code by hand because that is all we had in the past. That meant knowing a great deal of syntax and always having the manuals handy. It also meant pages and pages of code that were difficult to correct, difficult to maintain and hard to reuse for different compounds or devices. The first level of growth came when SAS introduced various windows and wizards (e.g., Import/Export Wizard, Report Window, Graph-n-Go) that gave us the ability to start using the wizard and then grab the SAS code and change it as necessary. The next innovations from SAS were tools like Enterprise Guide® and Clinical Data Integration® with their Graphical User Interface (GUI) that made programming a great deal easier, faster and much more efficient. You can still access all of the different data sources that you need from SAS data sets, spreadsheets and/or relational databases, but in a much easier way. You can still transform your raw data into SDTM domains using standard or custom transformations where mapping can be automatic or manual depending on what your input and output data requires. Data validation and compliance checks are much simpler because all of the tools are available for you to repeat the tasks for each protocol, compound or therapeutic area. Finally, if you do need to use legacy code or write your own routines, you can do that as well. This presentation will show you how experienced programmers can learn novel tricks and techniques with new tools, solutions and technology.

INTRODUCTION

As a clinical programmer, there are many paths to take. The main goal is always to access the data, manipulate and transform the data, analyze the data and report on the data. As a programmer, one can specialize in data management (DM) programming and spend a majority of the time cleaning the data through edit checks and the creation of patient listings and profiles. Another task of the DM programmer is to transform the data from its raw format into a standard format. This standard format could be the CDISC Study Data Tabulation Model (SDTM) that is requested by regulatory agencies upon submission of a new compound or it could be a sponsor's own standards. In the process of transforming the data, the DM programmer must make sure that the output conforms to the standard and is compliant as well as valid So another part of the job is to write programs to check the data against the standard and run the programs whenever a new study is about to be analyzed. Finally when all of the data has been transformed, the DM programmer must create a transport file that will be sent to the regulatory agency that will review the submission data.

The statistical (STAT) programmer takes the data that is cleaned and transformed by the DM programmer and creates tables, listings and graphs (TLG) for the clinical study report (CSR). Sometimes the data is taken from its raw state and transformed directly into TLGs, but most often the STAT programmer creates analysis datasets from which they can easily create the necessary output documents for the CSR. The STAT programmer is also tasked with creating ad hoc reports when needed, yearly safety updates, DSMB reports, and integrated safety and efficacy summaries.

The focus of this paper will be to study how the clinical programmer in data management (DM) has changed from creating programs in the past which used base SAS to how we can use new tools and solutions to produce the data that is needed in a new drug application submission. What did the programmer do in the past to cleanse the data and how has that process changed? Now that the data is requested to be in a standard format, what types of programs, macros and formats were used to transform the data? What is done now to make the process easier, more efficient and repeatable across protocols, compounds and therapeutic areas? From the old methodology to the new tools we will show how the transformation process can be changed and improved.

ACCESS and EDITS and STANDARDS, OH MY

Over the years, how we access the data has changed as often as the types of data that we use has changed. Data entry was part of the process of reading the data from paper case report forms (CRF) and creating SAS data sets with that data. Sometimes the lab data was written into the CRF and at other times it was sent in an electronic format that needed careful review and tricky coding to create the lab data sets. Some of the data entry systems did some preliminary edit checking (e.g., data range checks, limiting values entered, etc.), but most often the edit checks needed to be done after the data sets were created and systems were put in place to write the queries that were sent back to the clinical data collection sites. With the advent of relational databases (RDBMS) and electronic data capture, the amount and type of work needed to clean the data changed.

The way that the programmer reads the data has changed as well. There are many formats in which the data is sent to the sponsor. These include SAS data sets, MS Excel spreadsheets, RDBMS tables, ASCII files and electronic data capture. The mechanism for reading this data has changed along with the type of data that needs to be read. From writing many massive DATA steps to using LIBNAME or SAS Access engines, each type of data must be reviewed to see what the best choice is for reading it and creating the SAS data sets that will be used to process the data.

Of course, another task that was added to process was the introduction of standards into the submission process. As the regulatory agencies developed tools with which to review the data, the sponsors have been requested to develop standards. At first, companies created their own standards which, in some ways, reduced the complexity of the review process and yet in some ways introduced new issues. The Clinical Data Interchange Standards Consortium (CDISC) has worked over the last several years to create standards for the pharmaceutical, biotechnology and medical device companies to adopt. Now there is a whole new level of data management programming that needs to be done during the submission process.

THE APPROACH

The examples that we will use for this paper are based on the data that you will find in the Appendix. We chose data for the compound Nicardipine Hydrochloride (Nicardipine).Nicardipine is a calcium channel blocker, considered for the treatment of patients who have had a particular type of stroke classified as aneurismal subarachnoid hemorrhage (SAH). This type of stroke occurs when an aneurysm bursts. This causes bleeding inside the brain of the type called subarachnoid hemorrhage (SAH).

The study was designed to learn whether or not Nicardipine could prevent worsening of a stroke caused by narrowing of the blood vessels in the brain or improve the outcome following a stroke. In this case, study participants were children. This was a randomized study where some participants were assigned to a control group (placebo) and some to the experimental group (study drug).

The data for our examples are very old legacy data. Therefore, the names and types of variables are very different than the ones that you will find in the SDTM 3.1.2 metadata for the DM domain. The input data sources include three datasets: ADMIN2, RANDFILE, and REGISTER. The ADMIN2 file contains data about the start and end of treatment. The RANDFILE file contains information about which treatment was received by each subject. Lastly, the REGISTER file contains information about each subject such as date of birth, gender and race. All of these fields are needed in the DM domain. The target data is the last entry in the appendix. This is the resultant DM domain for the Nicardipine study in our example.

Also included in the appendix is the metadata for the SDTM 3.1.2 DM domain as specified in the SDTM Implementation Guide (version 3.1.2). If you take a look, you will see all of the objects in the implementation guide are included so that you can see what the structure is of the target data set.

IMPLEMENTING SDTM WITH BASE SAS APPROACH

In this section, we explore how to implement the SDTM data standard with BASE SAS as your primary tool. In the simplest form, this involves importing your source data into BASE SAS, transforming that data with DATA steps, SQL, and SAS PROCS, and then saving your SDTM domains as permanent data sets. For this particular instance of

creating the DM file we sort the three source datasets by patient identifier and then merge them together. The remaining activity is to define each of the SDTM DM variables in a DATA step and save that DM file to the target LIBREF. As is the case with all legacy SAS work, we have at our disposal a code editor window and SAS documentation perhaps in hardcopy as well as online but that is it.

The SAS code for our example problem of creating Demographic Domain (DM) from our raw source data looks like this:

```
proc sort data=rawdata.admin2 out=admin2;
 by studyno;
run;

proc sort data=rawdata.randfile out=randfile;
 by studyno;
run;

proc sort data=rawdata.register out=register(rename=(sex=sexn race=racen));
 by studyno;
run;

data readata;
 merge admin2(in=a)
       randfile(in=ra)
       register(in=re);
 by studyno;
run;

data target.dm;
 set readata;
 length STUDYID SUBJID USUBJID SITEID INVID INVNAM RACE ETHNIC ARM $40
        DOMAIN $8 RFSTDTC RFENDTC BRTHDTC DMDTC $64 AGEU $10 ARMCD $20
        SEX $1 COUNTRY $3 AGE DMDY 8;
 keep STUDYID DOMAIN USUBJID SUBJID RFSTDTC RFENDTC SITEID INVID INVNAM
      BRTHDTC AGE AGEU SEX RACE ETHNIC ARMCD ARM COUNTRY DMDTC DMDY;

 STUDYID='NIC001';
 DOMAIN='DM';
 USUBJID=LEFT(PUT(STUDYNO,Z6.));
 SUBJID=SUBSTR(COMPRESS(PUT(STUDYNO,Z6.)),4,3);
 if nmiss(TXBEGDAT,TXBEGTIM)=0 then
   RFSTDTC=PUT(DHMS(TXBEGDAT ,0,0,TXBEGTIM ),IS8601DT.);
 if nmiss(TXENDDAT,TXENDTIM)=0 then
   RFENDTC=PUT(DHMS(TXENDDAT ,0,0,TXENDTIM ),IS8601DT.);
 SITEID=SUBSTR(RPTINV,1,3);
 INVID=' ';
 INVNAM=PUT(RPTINV,$INV.);
 if nmiss(DOB)=0 then BRTHDTC=PUT(DOB,IS8601DA.);
 if nmiss(DOB,ADMDAT)=0 then
   AGE=(FLOOR((INTCK('month',DOB ,ADMDAT ) - (DAY(ADMDAT ) < DAY(DOB ))) / 12));
 AGEU='YEARS';
 SEX=SUBSTR(PUT(SEXN,SEX.),1,1);
 RACE=PUT(RACEN,RACE.);
 ETHNIC=' ';
 if TRT='A' then ARMCD='NIC15';
    else if TRT='B' THEN ARMCD='PLA';
    ELSE ARMCD='';
 ARM=PUT(TRT,$TREAT.);
 COUNTRY='USA';
 DMDTC=PUT(ADMDAT,IS8601DA.);
 if nmiss(txbegdat,admdat)=0 then do;
   if txbegdat >= admdat then dmdy = txbegdat - admdat + 1;
      else dmdy = txbegdat - admdat;
 end;

 label STUDYID='Study Identifier'
       DOMAIN='Domain Abbreviation'
```

```
            USUBJID='Unique Subject Identifier'
            SUBJID='Subject Identifier for the Study'
            RFSTDTC='Subject Reference Start Date/Time'
            RFENDTC='Subject Reference End Date/Time'
            SITEID='Study Site Identifier'
            INVID='Investigator Identifier'
            INVNAM='Investigator Name'
            BRTHDTC='Date/Time of Birth'
            AGE= 'Age in AGEU at RFSTDTC'
            AGEU='Age Units'
            SEX='Sex'
            RACE='Race'
            ETHNIC='Ethnicity'
            ARMCD='Planned Arm Code'
            ARM='Description of Planned Arm'
            COUNTRY='Country'
            DMDTC='Date/Time of Collection'
            DMDY='Study Day of Collection' ;
    run;
```

As you can see, this program consists of three SORT procedure steps, a DATA step to merge the source data, and a final DATA step to derive the SDTM DM variables that are needed and save it as the final DM file.

BASE SAS APPROACH CHALLENGES

There are a number of challenges with the reliance on BASE SAS alone to perform SDTM domain data creation. A primary issue is the management of metadata since there is none provided with BASE SAS alone. One thing to note about this program is that you need to type in all of the LENGTH and LABEL statements to define the SDTM metadata for the final domain data sets. This typing of metadata is tedious, prone to error, and likely to result in inconsistencies across SDTM domain metadata for a trial. You also have no real regulation of the target metadata and no real-time validation that your resulting domain is valid SDTM data. When using this BASE SAS approach you also run into logistic and strategic issues with code maintenance and reusability of the SAS code. The BASE SAS code itself can become difficult to read which makes maintenance difficult. This kind of coding tends to become difficult to read and "one-off" in nature resulting in the fact that future SAS code reusability is limited.

BASE SAS APPROACH BENEFITS

The primary advantage, although some might actually consider this a disadvantage, of the BASE SAS approach is that you have no restrictions as to what you can do with your SAS code. You have the full arsenal of BASE SAS and can utilize any SAS procedure, macro code, or SQL procedure code you wish to solve the problem of SDTM data conversion. Some have taken the BASE SAS approach to SDTM creation work and have augmented it with commonly available tools such as Microsoft Access or Excel as a place to store and leverage metadata. This augmented approach is better than BASE SAS solutions alone because you have your target SDTM metadata in a more manageable source and you can consider the effort somewhat data driven and less prone to metadata consistency errors

IMPLEMENTING SDTM WITH SAS ENTERPRISE GUIDE

In this section we explore how you would go about implementing the SDTM domain creation with SAS Enterprise Guide. SAS Enterprise Guide gives you a graphical user interface and some additional tools that you can bring to bear upon the task of SDTM file creation. The first step in this effort was to define a LIBREF called LIBRARY that would point to the permanent format catalog associated with the source legacy datasets. Then we were able to simply drag and drop the source datasets into the Enterprise Guide Process Flow window. With the data in the process flow, it was trivial to apply PROC SORT Enterprise Guide Tasks to sort the data by patient identifier. At this point the task of SDTM conversion converges on the same process used with the BASE SAS solution alone where the data is merged, SDTM variables defined, and the permanent DM dataset saved just as before.

The following display is a screen shot of the Enterprise Guide project that creates our SDTM DM file:
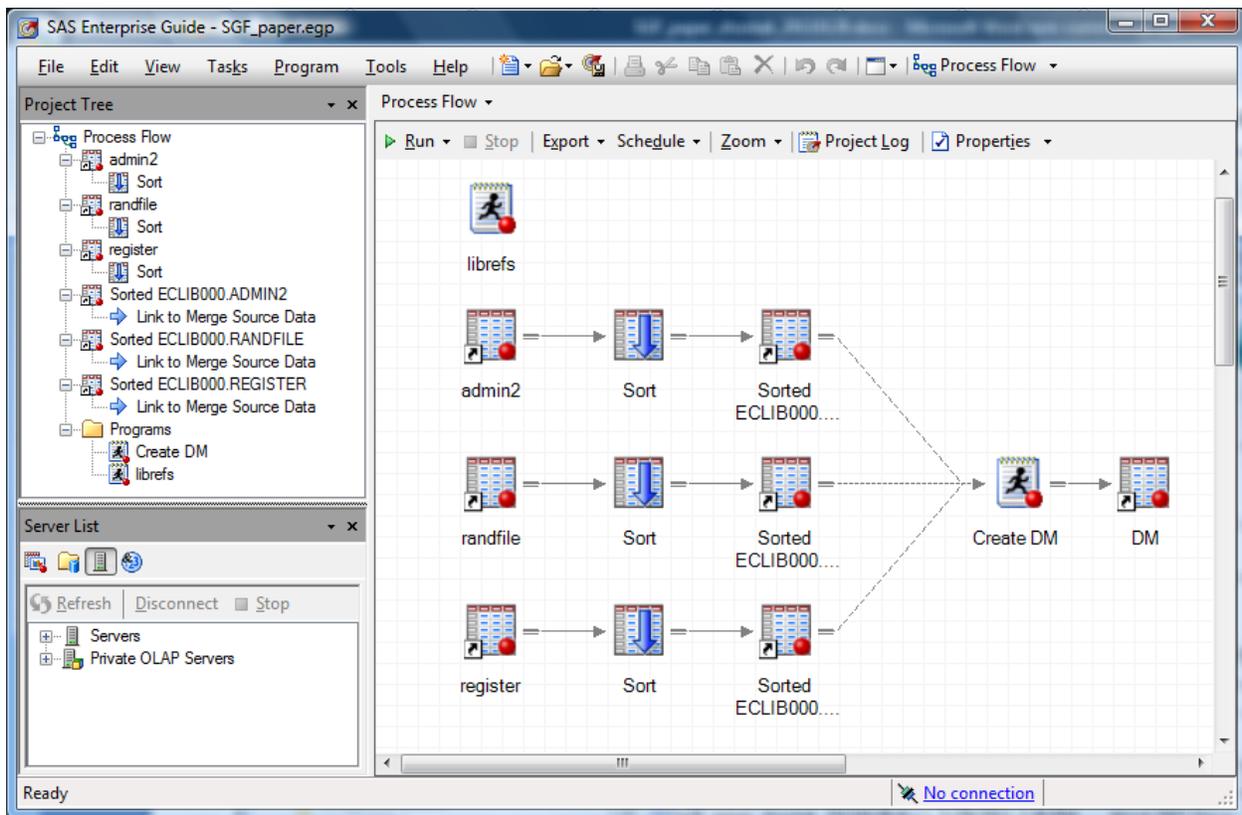
Display 1. Enterprise Guide Screen Shot of DM Creation

This Enterprise Guide project you see in Display 1 follows a flow very similar to the one in the BASE SAS implementation before it where you have a few PROC SORT steps with a DATA step merging and deriving the needed variables and then saving the DM file. However in this case the whole process is presented graphically in the Enterprise Guide Process Flow window. Here is what the "Create DM" step from Display 1 looks like:

```
**** MERGE SOURCE DATA THEN DEFINE AND DERIVE DM SDTM VARIABLES;
data target.dm;
  merge sasuser.admin2
        sasuser.randfile
        sasuser.register(rename=(sex=_sex race=_race));
     by studyno;

     **** CHECK FOR UNIQUENESS OF PRIMARY KEY;
     if not (first.studyno and last.studyno) then
          put "WARN" "NING: duplicate patient identifiers " studyno=;

     **** DEFINE DM VARIABLES;
     keep STUDYID DOMAIN USUBJID SUBJID RFSTDTC RFENDTC SITEID INVID INVNAM
          BRTHDTC AGE AGEU SEX RACE ETHNIC ARMCD ARM COUNTRY DMDTC DMDY;

      attrib STUDYID length = $40 label = 'Study Identifier';
      attrib DOMAIN length = $8 label = 'Domain Abbreviation';
      attrib USUBJID length = $40 label = 'Unique Subject Identifier';
      attrib SUBJID length = $40 label = 'Subject Identifier for the Study';
      attrib RFSTDTC length = $64 label = 'Subject Reference Start Date/Time';
      attrib RFENDTC length = $64 label = 'Subject Reference End Date/Time';
```

5

```
        attrib SITEID length = $40 label = 'Study Site Identifier';
        attrib INVID length = $40 label = 'Investigator Identifier';
        attrib INVNAM length = $40 label = 'Investigator Name';
        attrib BRTHDTC length = $64 label = 'Date/Time of Birth';
        attrib AGE length = 8 label = 'Age in AGEU at RFSTDTC';
        attrib AGEU length = $10 label = 'Age Units';
        attrib SEX length = $2 label = 'Sex';
        attrib RACE length = $40 label = 'Race';
        attrib ETHNIC length = $40 label = 'Ethnicity';
        attrib ARMCD length = $20 label = 'Planned Arm Code';
        attrib ARM length = $40 label = 'Description of Planned Arm';
        attrib COUNTRY length = $3 label = 'Country';
        attrib DMDTC length = $64 label = 'Date/Time of Collection';
        attrib DMDY length = 8 label = 'Study Day of Collection';

    **** DERIVE SDTM DM VARIABLES;
    studyid = "NIC001";
    domain = "DM";
    usubjid = left(put(studyno,z6.));
    subjid = substr(compress(put(studyno,z6.)),4,3);
    if txbegdat ne . and txbegtim ne . then
            rfstdtc = put(dhms(txbegdat, 0, 0, txbegtim), is8601dt.);
    if txenddat ne . and txendtim ne . then
            rfendtc = put(dhms(txenddat, 0, 0, txendtim), is8601dt.);
    siteid = substr(rptinv,1,3);
    invid = '';
    invnam = put(rptinv,$inv.);
    brthdtc = put(dob,is8601da.);
    age = floor((intck('month',dob ,admdat ) - (day(admdat) < day(dob ))) / 12);
    ageu = 'YEARS';
    sex = substr(put(_sex,sex.),1,1);
    race = put(_race,race.);
    ethnic = '';

    if trt = 'A' then armcd = 'NIC15';
    else if trt = 'B' then armcd = 'PLA';
    arm = put(trt,$treat.);
    country = 'USA';
    dmdtc = put(admdat,is8601da.);
    if txbegdat >= admdat then dmdy = txbegdat - admdat + 1;
    else dmdy = txbegdat - admdat;
  run;
```

ENTERPRISE GUIDE APPROACH CHALLENGES

Using Enterprise Guide as the primary tool to create SDTM data suffers some similar challenges to using BASE SAS alone. You are still lacking metadata management here and as you can see above, all of the variable lengths and labels are still manually typed into the program code. Again, you have no real regulation of the target metadata and no real-time validation that your resulting domain is valid SDTM data. Although Enterprise Guide provides the helpful Process Flow GUI, the "Tasks" available in Enterprise Guide that you can drop into your Process Flow is limited to sorting, appending, and transposing the data. The SAS Clinical Data Integration product comes with more available data management tasks in the form of what it calls "Transformations" instead of "Tasks" as we will see shortly.

ENTERPRISE GUIDE APPROACH BENEFITS

There are some advantages, over BASE SAS alone at least, to using Enterprise Guide to create SDTM data. As with the BASE SAS approach, you once again have the full arsenal of BASE SAS and can utilize any BASE SAS PROC, SAS MACRO, or SAS SQL code you wish to solve the problem of data conversion. However, with Enterprise Guide you get some additional assistance in the form of automated "tasks" that you can drag and drop into your project. You can see the PROC SORT driven "Sort" task used in Display 1 above, but there are other useful tasks for SDTM

creation such as the data splitter, data appender, and data transposing (rows to columns and columns to rows) tasks that can be very helpful here. If we had a more difficult domain to create then these additional prepackaged "tasks" can be leveraged in the process flow and programming. Also it is worth mentioning with Enterprise Guide 4.3 that you get more of a true development environment in SAS than ever before. Enterprise Guide 4.3 includes code completion facilities and interactive syntax guides found in other software development environments that you will love as a SAS programmer. Because of the Enterprise Guide process flow view, the SDTM work lends itself to being more manageable and reusable long term because the programming itself tends to be less spaghetti code. Finally, just as with the BASE SAS approach, the Enterprise Guide approach could be used in conjunction with tools such as Microsoft Access or Excel to give you a minimal way of managing your SDTM metadata.

IMPLEMENTING SDTM WITH SAS CLINICAL DATA INTEGRATION

Now that we have explored the creation of SDTM files with BASE SAS and SAS Enterprise Guide approaches, it is now a good time to look at the "full monty" SAS approach to SDTM data creation work and that involves using SAS Clinical Data Integration. SAS Clinical Data Integration is an ETL tool built on top of the SAS Data Integration product and it comes with clinical trials additions that we will use in this process. To begin this process in SAS CDI, we drag and drop the SDTM DM domain from our metadata repository and you will see that in Display 2 below as a beige box called "DM" to the far right. That target domain already has defined for you the table and variable level metadata that you need and it also includes appropriate integrity constraints on the data. Now that we have our target defined, we drag and drop our source datasets and you can see those in Display 2 as the three beige boxes to the far left. The next step is to join the three source datasets via a SQL join which is done by dragging and dropping the predefined "SQL Join" SAS CDI transform. The "Extract" transformation step you see in Display 2 below is where the SDTM DM variables get defined in a process analogous to the BASE SAS and SAS EG DATA step code you saw previously. Within SAS CDI this is done within point and click driven PROC SQL code building steps. The final step is to insert the "Table Loader" transformation which takes the SAS dataset from the "Extract" step and saves the permanent DM dataset.
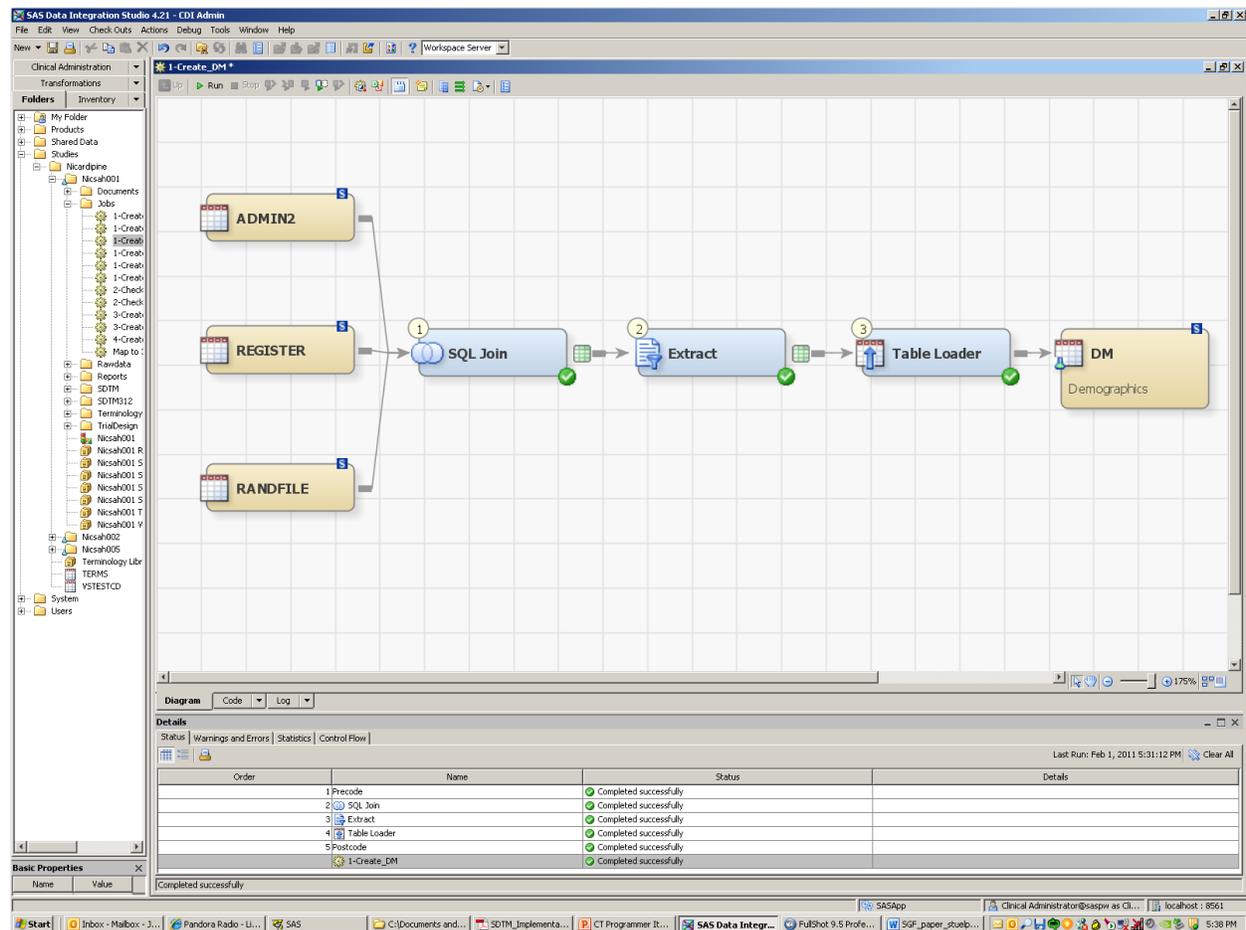
Display 2. SAS Clinical Data Integration Screen Shot of DM Creation

SAS CLINICAL DATA INTEGRATION APPROACH CHALLENGES

Since SAS Clinical Data Integration handles many facets to SDTM data creation the challenges to using it are a bit limited. Probably the biggest challenge for a SAS programmer is learning to give up slinging BASE SAS code and learning to rely on the tool to do the work. Also, SAS Clinical Data Integration relies on SAS SQL under the hood quite a bit so if you are an "old-school" BASE SAS programmer then brushing up on your SQL skills is advantageous. As with the BASE SAS and Enterprise Guide solutions, you can utilize any BASE SAS procedures you wish to use, but the key advantage to using Clinical Data Integration is in its ability to manage your metadata. Therefore, you want to avoid writing a bunch of custom SAS code because that limits the tool's ability to control the work. This can be a bit of a challenge because you have to learn to work and program largely within the confines of the transforms available within CDI. SAS CDI also makes you think about metadata management so there is a bit of setup that needs to be done in the tool in terms of defining your target data metadata up front.

SAS CLINICAL DATA INTEGRATION APPROACH BENEFITS

SAS Clinical Data Integration gives you the same kind of process flow view and drag and drop tasks/transforms that Enterprise Guide gives you, but it gives you what neither BASE SAS nor Enterprise Guide can provide alone. SAS Clinical Data Integration manages your metadata for your SDTM work. It controls the target SDTM metadata so compliance with a defined SDTM standard is built into your workflow. It also connects the metadata across you SDTM data creation so that you can analyze your data for changes and updates and also propagate a change across your SDTM data creation.

If you use SAS Clinical Data Integration as intended with standard transforms, it essentially enforces a bit of consistency of process in creating SDTM domains. This consistency along with the process view allows for SDTM creation jobs to be more easily maintained and also allows for reuse of jobs. SAS Clinical Data Integration also allows for "typical" SDTM generation tasks, such as study day (--DY) or ISO date (--DTC) creation, to be standardized into user written transforms that can be dragged and dropped into future SDTM jobs.

Although SAS Enterprise Guide provides a number of common "tasks" that can be dragged and dropped into your process flow, SAS Clinical Data Integration provides a much more expansive list of transformations for you to choose from. Several of those are extremely handy in terms of creating SDTM domains and those include the sort, transpose, data joiner, lookup table, data extraction, and data loader transformations.

Finally, SAS Clinical Data Integration is integrated with the SAS Clinical Standards Toolkit found with BASE SAS software. There are preexisting CDI transformations that allow you to validate your SDTM datasets based on the SDTM metadata and also to automatically generate your define.xml file which is a huge benefit.

CONCLUSION

In this paper we compared various SAS technologies in the task of converting clinical trials data into the CDISC SDTM. We looked at a BASE SAS approach, a SAS Enterprise Guide approach, and a SAS Clinical Data Integration approach. What we saw in the three approaches was that we moved from little tool support to heavy context specific tool support in the work. It used to be that when clinical SAS programmers were confronted by a data transformation task such as SDTM conversions that all we had was BASE SAS. Now we have a better GUI tool to use in SAS Enterprise Guide that helps us in SAS code development. In addition to that improved GUI, we also have a new clinical trials and CDISC friendly industry-specific tool in SAS Clinical Data Integration to use. SAS Clinical Data Integration gives us a way to manage our metadata and process in a way that we did not have before while still allowing us to write BASE SAS code when needed. It is a tool like this that we needed to have in order to have largely metadata driven transformation processes that can scale up to perform numerous data conversions in a reliable and efficient manner.

CONTACT INFORMATION

Janet Stuelpner   **Jack Shostak**
janet.stuelpner@sas.com   **jack.shostak@duke.edu**

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 trademarks of their respective companies.

APPENDIX

Subset of data used in example programs along with SDTM metadata definitions for DM domain.

Data Listing of ADMIN2

Obs	STUDYNO	TXBEGDAT	TXBEGTIM	TXENDDAT	TXENDTIM	SAHDAT	DAYSTOP
1	11001	12OCT1987	23:25	24OCT1987	10:00	10OCT1987	14
2	11002	14OCT1987	16:30	21OCT1987	10:30	12OCT1987	9
3	11003	10NOV1987	18:00	23NOV1987	19:00	09NOV1987	14
4	11004	02DEC1987	6:10	14DEC1987	18:00	30NOV1987	14
5	11005	08DEC1987	10:15	19DEC1987	0:30	06DEC1987	13
6	11006	15DEC1987	21:00	27DEC1987	20:00	13DEC1987	14
7	11007	30DEC1987	16:15	12JAN1988	16:15	29DEC1987	14
8	11008	09JAN1988	11:35	22JAN1988	8:00	08JAN1988	14
9	11009	24FEB1988	21:30	08MAR1988	24:00	23FEB1988	14
10	11010	01MAR1988	21:00	13MAR1988	19:40	26FEB1988	16
11	11011	11MAR1988	13:30	24MAR1988	18:00	10MAR1988	14
12	11012	16MAR1988	2:45	28MAR1988	18:00	14MAR1988	14
13	11013	05APR1988	20:00	18APR1988	18:00	04APR1988	14
14	11014	05APR1988	23:15	12APR1988	7:30	04APR1988	8
15	11015	14APR1988	18:00	15APR1988	20:30	13APR1988	2
16	11016	11MAY1988	20:00	24MAY1988	18:00	10MAY1988	14
17	11017	13MAY1988	21:00	26MAY1988	20:00	12MAY1988	14
18	11018	05JUN1988	15:00	17JUN1988	14:00	03JUN1988	14
19	11019	07JUL1988	17:00	15JUL1988	12:00	30JUN1988	15
20	11020	28JUL1988	23:45	29JUL1988	3:00	28JUL1988	1

Data Listing of RANDFILE

Obs	STUDYNO	CENTER	CTRREGNO	TRT	TRTNAME
1	11001	01	1	Placebo	PLACEBO
2	11002	01	2	Nicardipine .15	IV NICARDIPINE
3	11003	01	4	Nicardipine .15	IV NICARDIPINE
4	11004	01	30	Placebo	PLACEBO
5	11005	01	31	Placebo	PLACEBO
6	11006	01	29	Placebo	PLACEBO
7	11007	01	45	Nicardipine .15	IV NICARDIPINE
8	11008	01	46	Nicardipine .15	IV NICARDIPINE
9	11009	01	66	Nicardipine .15	IV NICARDIPINE
10	11010	01	67	Placebo	PLACEBO
11	11011	01	116	Nicardipine .15	IV NICARDIPINE
12	11012	01	93	Placebo	PLACEBO
13	11013	01	115	Placebo	PLACEBO
14	11014	01	117	Placebo	PLACEBO
15	11015	01	9118	Nicardipine .15	IV NICARDIPINE
16	11016	01	166	Nicardipine .15	IV NICARDIPINE
17	11017	01	171	Placebo	PLACEBO
18	11018	01	205	Nicardipine .15	IV NICARDIPINE
19	11019	01	9249	Placebo	PLACEBO
20	11020	01	282	Nicardipine .15	IV NICARDIPINE

Obs	STUDYNO	RPTCTR	RPTHOSP	RPTINV	CTRREGNO	SAHDAT	ADMDAT	DOB	SEX	RACE
1	11001	01	011	KASSELL	1	10OCT1987	12OCT1987	24OCT1968	Male	Caucasian
2	11002	01	011	KASSELL	2	12OCT1987	12OCT1987	20JUL1908	Male	Caucasian
3	11003	01	011	KASSELL	4	09NOV1987	09NOV1987	03JAN1941	Male	Caucasian
4	11004	01	011	KASSELL	30	30NOV1987	30NOV1987	18SEP1915	Female	Caucasian
5	11005	01	011	KASSELL	31	06DEC1987	07DEC1987	10JUL1907	Female	Caucasian
6	11006	01	011	KASSELL	29	13DEC1987	14DEC1987	11AUG1934	Female	Caucasian
7	11007			KASSELL	45	29DEC1987	29DEC1987	27MAR1944	Male	Caucasian
8	11008	01	011	KASSELL	46	08JAN1988	08JAN1988	28NOV1931	Female	Caucasian
9	11009			KASSELL	66	23FEB1988	23FEB1988	25JUN1922	Female	Caucasian
10	11010			KASSELL	67	26FEB1988	01MAR1988	06APR1943	Female	Caucasian
11	11011			KASSELL	116	10MAR1988	11MAR1988	05NOV1960	Female	Caucasian
12	11012	01	011	KASSELL	93	14MAR1988	15MAR1988	07OCT1954	Female	Caucasian
13	11013			KASSELL	115	04APR1988	04APR1988	09JAN1924	Female	Caucasian
14	11014	01	011	KASSELL	117	04APR1988	05APR1988	15FEB1941	Male	Caucasian
15	11015			KASSELL	9118	13APR1988	13APR1988	25APR1933	Male	Caucasian
16	11016			KASSELL	166	10MAY1988	10MAY1988	18JUL1953	Female	Black
17	11017			KASSELL	171	12MAY1988	12MAY1988	11SEP1926	Female	Caucasian
18	11018			KASSELL	205	03JUN1988	04JUN1988	09OCT1957	Male	Caucasian
19	11019			KASSELL	9249	30JUN1988	06JUL1988	01APR1925	Female	Caucasian
20	11020			KASSELL	282	28JUL1988	28JUL1988	21OCT1941	Female	Caucasian

SDTM 3.1.2 Demography Metadata

Domain Prefix	Variable Name (minus domain prefix)	Variable Name	Variable Label	Type	Controlled Terms or Format	Role	CDISC Notes (for domains) Description (for General Classes)	Core
DM	STUDYID	STUDYID	Study Identifier	Char		Identifier	Unique identifier for a study.	Req
DM	DOMAIN	DOMAIN	Domain Abbreviation	Char	DM	Identifier	Two-character abbreviation for the domain.	Req
DM	USUBJID	USUBJID	Unique Subject Identifier	Char		Identifier	Identifier used to uniquely identify a subject across all studies for all applications or submissions involving the product. This must be a unique number, and could be a compound identifier formed by concatenating STUDYID-SITEID-SUBJID.	Req
DM	SUBJID	SUBJID	Subject Identifier for the Study	Char		Topic	Subject identifier, which must be unique within the study. Often the ID of the subject as recorded on a CRF.	Req
DM	RFSTDTC	RFSTDTC	Subject Reference Start Date/Time	Char	ISO 8601	Record Qualifier	Reference Start Date/time for the subject in ISO 8601 character format. Usually equivalent to date/time when subject was first exposed to study treatment. Required for all randomized subjects; will be null for all subjects who did not meet the milestone the date requires, such as screen failures or unassigned subjects.	Exp
DM	RFENDTC	RFENDTC	Subject Reference End Date/Time	Char	ISO 8601	Record Qualifier	Reference End Date/time for the subject in ISO 8601 character format. Usually equivalent to the date/time when subject was determined to have ended the trial, and often equivalent to date/time of last exposure to study treatment. Required for all randomized subjects; null for screen failures or unassigned subjects.	Exp
DM	SITEID	SITEID	Study Site Identifier	Char		Record Qualifier	Unique identifier for a site within a study.	Req
DM	INVID	INVID	Investigator Identifier	Char		Record Qualifier	An identifier to describe the Investigator for the study. May be used in addition to SITEID. Not needed if SITEID is equivalent to INVID.	Perm
DM	INVNAM	INVNAM	Investigator Name	Char		Synonym Qualifier	Name of the investigator for a site.	Perm
DM	BRTHDTC	BRTHDTC	Date/Time of Birth	Char	ISO 8601	Record Qualifier	Date/time of birth of the subject.	Perm
DM	AGE	AGE	Age	Num		Record Qualifier	Age expressed in AGEU. May be derived from RFSTDTC and BRTHDTC, but BRTHDTC may not be available in all cases (due to subject privacy concerns).	Exp
DM	AGEU	AGEU	Age Units	Char	(AGEU)	Variable Qualifier	Units associated with AGE.	Exp
DM	SEX	SEX	Sex	Char	(SEX)	Record Qualifier	Sex of the subject.	Req
DM	RACE	RACE	Race	Char	(RACE)	Record Qualifier	Race of the subject. Sponsors should refer to "Collection of Race and Ethnicity Data in Clinical Trials" (FDA, September 2005) for guidance regarding the collection of race (http://www.fda.gov/cder/guidance/5656fnl.htm) See Assumption below regarding RACE.	Exp
DM	ETHNIC	ETHNIC	Ethnicity	Char	(ETHNIC)	Record Qualifier	The ethnicity of the subject. Sponsors should refer to "Collection of Race and Ethnicity Data in Clinical Trials" (FDA, September 2005) for guidance regarding the collection of ethnicity (http://www.fda.gov/cder/guidance/5656fnl.htm).	Perm
DM	ARMCD	ARMCD	Planned Arm Code	Char	*	Record Qualifier	ARMCD is limited to 20 characters and does not have special character restrictions. The maximum length of ARMCD is longer than for other "short" variables to accommodate the kind of values that are likely to be needed for crossover trials. For example, if ARMCD values for a seven-period crossover were constructed using two-character abbreviations for each treatment and separating hyphens, the length of ARMCD values would be 20.	Req
DM	ARM	ARM	Description of Planned Arm	Char	*	Synonym Qualifier	Name of the Arm to which the subject was assigned.	Req
DM	COUNTRY	COUNTRY	Country	Char	(COUNTRY) ISO 3166	Record Qualifier	Country of the investigational site in which the subject participated in the trial.	Req
DM	DTC	DMDTC	Date/Time of Collection	Char	ISO 8601	Timing	Date/time of demographic data collection.	Perm
DM	DY	DMDY	Study Day of Collection	Num		Timing	Study day of collection measured as integer days.	Perm

Data Listing of SDTM 3.1.2 DM Domain

Obs	STUDYID	DOMAIN	USUBJID	SUBJID	RFSTDTC	RFENDTC	SITEID	INVID	INVNAM
1	NIC001	DM	011001	001	1987-10-12T23:25:00	1987-10-24T10:00:00	011		KASSELL
2	NIC001	DM	011002	002	1987-10-14T16:30:00	1987-10-21T10:30:00	011		KASSELL
3	NIC001	DM	011003	003	1987-11-10T18:00:00	1987-11-23T19:00:00	011		KASSELL
4	NIC001	DM	011004	004	1987-12-02T06:10:00	1987-12-14T18:00:00	011		KASSELL
5	NIC001	DM	011005	005	1987-12-08T10:15:00	1987-12-19T00:30:00	011		KASSELL
6	NIC001	DM	011006	006	1987-12-15T21:00:00	1987-12-27T20:00:00	011		KASSELL
7	NIC001	DM	011007	007	1987-12-30T16:15:00	1988-01-12T16:15:00	011		KASSELL
8	NIC001	DM	011008	008	1988-01-09T11:35:00	1988-01-22T08:00:00	011		KASSELL
9	NIC001	DM	011009	009	1988-02-24T21:30:00	1988-03-09T00:00:00	011		KASSELL
10	NIC001	DM	011010	010	1988-03-01T21:00:00	1988-03-13T19:40:00	011		KASSELL
11	NIC001	DM	011011	011	1988-03-11T13:30:00	1988-03-24T18:00:00	011		KASSELL
12	NIC001	DM	011012	012	1988-03-16T02:45:00	1988-03-28T18:00:00	011		KASSELL
13	NIC001	DM	011013	013	1988-04-05T20:00:00	1988-04-18T18:00:00	011		KASSELL
14	NIC001	DM	011014	014	1988-04-05T23:15:00	1988-04-12T07:30:00	011		KASSELL

Obs	BRTHDTC	AGE	AGEU	SEX	RACE	ETHNIC	ARMCD	ARM	COUNTRY	DMDTC	DMDY
1	1968-10-24	18	YEARS	M	Caucasian		PLA	Placebo	USA	1987-10-12	1
2	1908-07-20	79	YEARS	M	Caucasian		NIC15	Nicardipine .15	USA	1987-10-12	-2
3	1941-01-03	46	YEARS	M	Caucasian		NIC15	Nicardipine .15	USA	1987-11-09	-1
4	1915-09-18	72	YEARS	F	Caucasian		PLA	Placebo	USA	1987-11-30	-2
5	1907-07-10	80	YEARS	F	Caucasian		PLA	Placebo	USA	1987-12-07	-1
6	1934-08-11	53	YEARS	F	Caucasian		PLA	Placebo	USA	1987-12-14	-1
7	1944-03-27	43	YEARS	M	Caucasian		NIC15	Nicardipine .15	USA	1987-12-29	-1
8	1931-11-28	56	YEARS	F	Caucasian		NIC15	Nicardipine .15	USA	1988-01-08	-1
9	1922-06-25	65	YEARS	F	Caucasian		NIC15	Nicardipine .15	USA	1988-02-23	-1
10	1943-04-06	44	YEARS	F	Caucasian		PLA	Placebo	USA	1988-03-01	1
11	1960-11-05	27	YEARS	F	Caucasian		NIC15	Nicardipine .15	USA	1988-03-11	1
12	1954-10-07	33	YEARS	F	Caucasian		PLA	Placebo	USA	1988-03-15	-1
13	1924-01-09	64	YEARS	F	Caucasian		PLA	Placebo	USA	1988-04-04	-1
14	1941-02-15	47	YEARS	M	Caucasian		PLA	Placebo	USA	1988-04-05	1