PO-09

SAS Programming tips and techniques for Data Mapping

Sheetal Nisal, Sterling Healthstat, Inc., Plainsboro, NJ

Abstract

Data mapping is a very common process for getting the data in homogeneous standards and making it ready for analysis. In healthcare and pharmaceutical industry mapping is a very common process to analyze the data and to take informed decisions based on it. SAS® is a powerful software and can be used with lot of ease to map the data from legacy data standards to target data standards. If you are a novice user of SAS® or if you have to do data mapping first time, you may be curious to know more about what type of SAS® coding techniques you may need to use while mapping the data. Typically data mapping involves a set of well-defined inter-dependent processes. To complete each process programmatically using SAS®, data mapping specialist needs to know some basic but powerful features of SAS® programming and should be able to use those features effectively by understanding the data.

This paper illustrates such basic SAS® techniques with necessary illustrations on data mapping and transformations. In this case, for ease of understanding, typical clinical domain in CDISC submission data standards is considered as a target data standard. Application of SAS® techniques is explained with reference to steps in the typical data mapping process. Transformation of each variable and its attributes requires careful use of SAS® data steps and procedures. Quality check in data mapping process is a very important step and it involves verification of data records, ensuring proper transformations of variables wherever applicable, and having meta data aligned to the standards requirements. This paper illustrates SAS® tips and techniques which are recommended to follow while mapping the data.

Introduction

The 'life cycle' of data starts from collection of data and ends at analysis and archival of data. During this cycle, data goes through lot of transformations and different stakeholders and users of data deal with it differently based on their needs. In clinical and healthcare industry, standardization of tools and systems at data collection and database management systems often pose a need to map the data from one data standards to another data standards. Whereas healthcare industry follows Health Level 7 or HL7 standards, pharmaceutical industry has adopted CDISC standards for collection, analysis, and submission of data. There are different tools and systems used at collection and management of data. These include Oracle, Sequal, Microsoft Excel, Microsoft Access, de-limited data etc. Most of these databases have interface with SAS® software and systems. On a reporting and analysis side, SAS® is widely used system. The collected data often follows different standards depending on data collection needs. Whereas reporting techniques are fairly standardized and there is a higher possibility that the organization may follow a consistent data reporting standards. These differences in dealing with data standards lead to the 'data mapping process'.

In data mapping process, data following one standards (often referred as 'legacy standards') is transformed into another standards (often referred as 'target standards'). The data standards imply that the data is following certain type of meta data and data model. So, in a nutshell, the data mapping involves changing the meta data without changing the intent of the data. Another key aspect of data mapping involves merging and or splitting of input data into one or more target data standards depending on target meta-data needs. In following discussion we assume that mapping process is conducted by developing SAS® codes as oppose to using a standard utility or tool. The data mapping process starts with development of a layout and a guideline on how to develop mapping algorithms. Then these algorithms are used to develop a mapping program or SAS® codes. Actual data mapping is carried out by execution of these SAS® programs. After mapping, there is usually a quality check to ensure that the data is being mapped correctly. All these activities and sub-activities in mapping process can be handled dynamically using base SAS® software. In the following discussion, we will see how we can effectively use SAS® software to dynamically pull the information from one step to another in data mapping process. In this process we will consider that we are mapping Concomitant medication data in clinical trials from legacy data standards to CDISC submission data standards.

Process Overview and Key Components

As we discussed in earlier section, mapping process involves development of mapping layouts, transformation codes and quality check codes of mapped data. Mapping layouts are often maintained in the form of spreadsheets. These layouts lists the legacy and target standard meta data and explain the data transformation instructions. Such layout template is listed in Appendix-1.0. This layout serves as a basic guideline for the data transformation code development. Information from this layout is translated into SAS® code and the meta data for target standards as listed in this spreadsheet needs to be applied on the variables transformed.

If we look at the process integrity there needs to be consistent information in the following components:

1) Meta data of legacy data and legacy data standards section from mapping layout spreadsheet.
2) Meta data from mapping layout spreadsheet and section of SAS® code which defines the formats, and attributes of the variables.
3) Meta data of target data and target data standard section from mapping layout spreadsheet.

During this process, the quality of data should not be compromised. Clearly in this process any manual intervention or manual entry of data leads to human error resulting into inconsistent meta data at different components of mapping process. Base SAS® software and its simple yet powerful features can be used effectively in this process to automate and dynamically pull the information.

Another important consideration in data mapping is the differences in data models. Quite often the data records require transformations from horizontal to vertical or vertical to horizontal structure. Let's look at the clinical trial concomitant data and use of base SAS® software to map this data into CDISC Submission data standards.

Reading the Data

In clinical trials, most of the time the database and data collection systems have SAS® interface which allows data mapping personnel to get the data on SAS® server in the form SAS datasets. These could be in SAS data format or SAS transport file format. In any circumstances, if the data is not available in SAS® format, data mapping personnel can use the SQL pass through facility or SAS Connect utilities to extract the data from database to SAS server. For more details on these techniques, please refer to the manuscript "Using SAS, SAS/ACCESS, AND SQL pass through to query and join Oracle Tables", by Barbara Okerson. Another input form the data can be is in Microsoft Excel or Microsoft Access. Data mapping personnel or SAS® programmer can use the dynamic data exchange to read the data from Excel. This is a very powerful technique to control reading of data from each spreadsheet and specified cells from Excel. See the sample code for example:

```
filenameconmeddde 'Excel|C:\temp\trial01\Data\[conmed.XLS]data!R2C2:R229C12';
data cm;
  infileconmedmissover;
  input subjectvisit med $ startdatestarttimestopdatestoptimemedclass $

            indication $ result $;
run;
proc print data=bean; run;
```

The above code reads few variables from the concomitant medication data from the Excel spreadsheet. These basic techniques of reading the data and writing the data into Excel spreadsheet can be used for the meta data as well. Like we discussed earlier, the meta data from the legacy data needs to be dynamically populated from the data to mapping layout spreadsheet. This can be achieved in two steps:

1) Reading the meta data and storing it (if required). This can be done using the 'contents' procedure.

```
Proc contents data = conmed out = cmmeta;
Run;
```

2) Then the saved meta data can be written into Excel mapping layout spreadsheet. This can be done in multiple ways. Either programmer can use proc export or dynamic data exchange, or such meta data can be 'reported' using report procedure and ODS tagsets options. Refer to the manuscript "Moving Data and Results Between SAS® and Microsoft Excel", by HarryDroogendyk for more details on different options to read the SAS data into Excel and the system requirements for each options.

Similar to reading the legacy meta data, the target meta data can be obtained directly from the CDISC portal. Now, with the above steps the mapping specialist can get both the legacy and target meta-data into mapping layout spreadsheet seamlessly without manual intervention.As shown in appendix-1.0, the mapping layout spreadsheet allows the user to spell out details regarding the data transformation logic of each variable.

Creation of SAS® Code Template

After completion of data mapping layout, the next step in data mapping process is to create a SAS®C ode template which programmer can use to write a code for detailed data transformations. This SAS® code template can be used for consistency of SAS® code development and to ensure that the meta data and data transformation logic spelled out in the mapping layout spreadsheet can be used 'as is' without any further unanticipated changes. This is a good programming practice for data mapping work. The data exchange techniques discussed in earlier section for transfer of meta data from Excel to SAS can be used to transfer the 'target meta data' information from the Excel mapping layout to SAS® program format. Alternatively, such meta data information can be exported in text or rich text format and from there it can be copied into SAS code. Once this meta data information is moved to SAS code template, it can be edited a bit to accommodate it into SAS® formats or into attribute functions.

For example, consider the following information from mapping layout spreadsheet:

Domain Prefix	Variable Name	Variable Label	Type	Controlled Terms or Format	Role	CDISC Notes (for domains) Description (for General Classes)
CM	STUDYID	Study Identifier	Char		Identifier	Unique identifier for a study.
CM	DOMAIN	Domain Abbreviation	Char	CM	Identifier	Two-character abbreviation for the domain.

Above information can be transferred to SAS® code template in the following manner:

```
attrib

        STUDYID length = $20 label = 'Study Identifier' format $20.

        DOMAIN length = $2 label = 'Domain Abbreviation' format $2.

;
```

In the same manner the meta data for all variables from the target data standards from mapping layout spreadsheet can be transferred to the SAS® code attribute statements. This segment of SAS® code will create an empty dataset

structure with target data standards meta data associated with it. In this process, all the information from mapping layout spreadsheet is being pulled dynamically from Excel to SAS® code. Such code templates can be produced for every target dataset to be produced in mapping process. One such sample SAS® code template is illustrated in appendix-1.1. In the same manner, variable formats can be pulled together into SAS® code template and can be applied directly to the variables.

Data Transformations

After automating the process and creating SAS® code templates data mapping personnel and SAS® programmers need to use some basic procedures and data steps in SAS® to conduct the data transformations. Some of these steps and procedures along with their application are stated below:

Merging of Data: Quite often the input or legacy data for each domain is split across multiple datasets. In clinical trials, often structure of input dataset is aligned to the way case report form is designed. Considering such layout, there is a strong chance that the data for one clinical domain may span across multiple datasets. In some cases the target data standards require the use of multiple type of datasets. Such requirements need merging of datasets. As an example, in case of concomitant medication, CDISC requires a variable 'CMCLAS'. This 'permissible' variable is derived by merging the dictionary codes with actual concomitant medication data.

In data mapping 'subject ID', 'dates/time of medication', and dose are used as a primary key variables for merging process. These merging can be used by merge statement or by procsql.

Transposition of Data: Most of the data collection systems use horizontal structure for collecting and managing data. Reporting and analysis however requires vertical structure of data in some cases. Such horizontal to vertical structure data transformation is one of the key activity in mapping process. Proc transpose in SAS® is used extensively for such transposition. In some cases, programmers prefer to use arrays for such transposition purpose.

SAS® functions: In order to ensure that the data records in target data structure meet the target data standards requirements, quite a few SAS® functions are required to use extensively in the data mapping process. Some functions with their examples in clinical data mapping are stated below:

1) Concatenation function: Many target data standards variables (as an example USUBJID) are result of concatenation of more than one variable records. These derived variables are created using concatenation function.
2) Scan, Index, Indexc, indexw and substr function: This is mostly used for splitting character records to derive one variable values out of another variable record. As an example, the domain name abbreviation can be obtained from any of the existing variable record values.
3) Conversion functions (input and put): Character to numeric, numeric to character, and numeric to date conversions are quite frequent type of conversion in data mapping. These type of conversions are required because of differences in data standards and their meta-data.
4) Changing cases of character records: UPCASE, LOWCASE etc. to ensure that the case of records is aligned to target data standards requirements.
5) Other functions: In some cases data transformations requires user to utilize other functions like lag (which returns the value from queue).

SAS® Formats: Target data standard often requires different variable formats than those are defined for legacy variables. In some cases like CDISC submission data standards, date variables follow ISO formats. SAS format statements are extensively used in mapping process to define numeric, character, and date formats as per the target data standards requirements. In many cases, target data standards follow user defined or industry specific controlled terminology applied to certain variables. Such controlled terminology application is done using SAS® format statements.

Validation procedures:

Mapped data and the overall coding steps in mapping process requires validation to ensure that the mapping algorithm is getting correctly interpreted in the SAS® code. SAS® has quite a few robust and easy to use procedures for that purpose. Some key procedures used to do data mapping validation are listed below:

Proc print: Used to do a quick subset print of transformed values of variable.

Proc contents: To ensure that the target meta data is correctly defined.

Proc compare: To ensure that the target meta data of mapped data is completely identical with the target meta data defined in the mapping layout spreadsheet.

Procfreq: To compare frequency counts of mapped data against the input datasets used.

Proc datasets: For managing libraries and members within it.

Proc sort: For sorting the data before merging.

In addition to the above procedures programmers and data mapping personnel may use some basic statistical procedures like proc means and procunivariate to do a quick check on data.

Conclusion

In order to execute the data mapping process efficiently and quickly there are many 'mapping tools' available in market. SAS® has its own SAS® clinical data integration, which itself is a robust data mapping tool. Often data mapping is done by individuals with standards or data management background. With knowledge of SAS® system, SAS® programmers also do data mapping extensively. If the organization has skilled personnel with strong clinical data knowledge, and knowledge of base SAS® system, SAS® programmers can automate the mapping process by using simple, yet powerful SAS® procedures available in base SAS® software. With these simple procedures and techniques, SAS® programmers can develop an automated utility for mapping process. Such utility can dynamically transfer information from one stage to another stage. If programmers spend some time in developing standardized SAS® codes for each stage, it is not necessary for an organization to buy licenses of data mapping tools. Data mapping involves lot of subjective decisions which can not be factored in with standardized data mapping tools. In such circumstances, knowledge and application of simple procedures and techniques from base SAS® system is a most efficient and convenient option for data mapping work

Appendix- 1.0: Data Transformation Layout Template

Legacy Standards								Target Standards						
	Variale								Variable					
Dataset Name	Name	Label	Type	Length	Format	Origin	Mapping Instruction	Domain Prefix	Name	Label	Type	Length	Controlled Terms	Origin

Appendix- 1.1: SAS® Code Template for Data Mapping Program

SAS Program Name: cm.sas

Purpose: To map the legacy concomitant medication data into CDISC submission data standards.

Input: Conmed.sas7bdat *Input data for concomitant medication.* , WHO.sas7bdat (*WHO dictionary data.*), data mapping layout information.

Output: CM.sas7bdat *Concomitant medication submission ready data.*

Author: <Author Name>

Date: <Date of latest revision of code>

Revision History:

1.2- Final revision. Amended the code based on input from data management regarding variable cmtrt.

1.1- Amendment based on Quality check reviews.

1.0- Initial revision

---

libnameidata '/…./';

libnameodata '/……/';

/* Reading the data and assigning attributes based on data mapping layout. */

datacmin;

```
        setidata.conmed;

        attrib<var1><attributes>

                <var2><attributes>……..<var-n><attributes>;

run;

/* Data transformation code segment. */

…..

/*Final mapped data.*/
```

Reference

1) "USING SAS, SAS/ACCESS□, AND SQL PASSTHROUGH TO QUERY AND JOIN ORACLE TABLES: An Example Using the Health Care Finance Administration's SDPS (Medicare) Database", by Barbara B. Okerson, Ph.D. Mid-South Foundation for Medical Care, Inc.

2) "Moving Data and Results Between SAS® and Microsoft Excel", by HarryDroogendyk, Stratia Consulting Inc., Lynden, ON, Canada, presented at SAS Global Forum 2012.

3) Clinical Data Interchange Standards Consortium (CDISC), Submission Data Standards, on http://www.cdisc.org

4) SAS® system support documentation. On http://support.sas.com/documentation/

Contact Information: Your comments and questions are valued and encouraged. Author can be contacted at sheetal.nisal@sterlinghealthstat.com