Paper 207-2008

Practical Methods for Creating CDISC SDTM Domain Data Sets from Existing Data

Robert W. Graebner, Quintiles, Inc., Overland Park, KS

ABSTRACT

Creating CDISC SDTM domain data sets from existing clinical trial data can be a challenging task, particularly if the database was not designed with the SDTM standards in mind. A key step in the process involves determining which of the STDM domain datasets need to be produced for submission and then determining what conversion process will be necessary to produce them from the existing data. Adequate planning and documentation of the conversion process is an essential first step before programming begins. The basic component of the planning phase involves metadata mapping – determining how each of the variables in the existing data will relate to the variables contained in the SDTM domains to be produced. The documentation of the conversion process should be recorded in a format that facilitates efficient access by those involved in the planning, programming and validation phases of the conversion. Tools suited to the task of complex data mapping and data manipulation can significantly reduce cost and improve quality. This paper presents an example of a simple metadata mapping tool developed using SAS, Microsoft Excel and Visual Basic. The examples in this paper are based on the CDISC SDTM version 1.1, the SDTM Implementation Guide version 3.1.1 and SAS® version 9.1.3.

INTRODUCTION

In order to increase the efficiency of the drug development process, the Clinical Data Interchange Standards Consortium (CDISC) has developed a series of clinical study data standards to facilitate efficient transfer, access and review of clinical trial data. These standards include the Operational Data Model (ODM), the Study Data Tabulation Model (SDTM) and the Analysis Data Model (ADaM). This paper presents basic strategies and practical methods for creating SDTM domain data sets from clinical data management (CDM) system files. Before initiating the data mapping and conversion process it is crucial to have a basic understanding of the SDTM specifications. CDISC provides implementation guides for all of the CDISC data standards on their Website (**www.cdisc.org**). The SDTM Implementation Guide (SDTMIG) is an essential tool for anyone involved with the metadata mapping or programming associated with the creation of SDTM data sets. The SDTM Implementation Guide contains the specifications and metadata for all of the SDTM data domains and guidance for producing SDTM domain files. The SDTM is an evolving standard and it is important to ensure that everyone involved in the conversion process is adhering to the same version of the SDTM. It is also important to understand the difference in the version numbers for the SDTM standard and the associated implementation guide. The most recent versions in production are SDTM 1.1 and SDTMIG 3.1.1, which were released in 2005.

CDISC SDTM OVERVIEW

The purpose of creating CDISC SDTM domain data sets is to provide Case Report Tabulation (CRT) data to a regulatory agency, such as the FDA, in a standardized format that is compatible with available software tools that allow efficient access and correct interpretation of the data submitted. The SDTMIG provides documentation on metadata for the domain data sets that includes the file name, variable names, types, labels, formats, roles and controlled terminology. While most of the SDTM domain data sets have a normalized (vertical) structure, they were not designed for use in a clinical data management (CDM) system. It is highly desirable to incorporate CDISC standards to the extent practical when designing CDM data structures. Proper adherence to the standards can greatly reduce the effort necessary for data mapping. Important standards to adhere to are domain name, variable name, variable type and format. Matching the SDTM variable labels is not important. The SDTM standard labels are available in the standard metadata and the labels are not used for match merging in the mapping process. While the SDTM documentation does not specify variable lengths, it is highly desirable to maintain consistency in length among variables with the same name across domains and between studies.

While the SDTM data sets do contain some derived variables, they are not designed for use as analysis data sets. Adherence to the “one proc away”-philosophy for analysis files dictates the addition of additional derived variables and conversion to a horizontal structure. The SDTM data sets can however, be used in the creation of analysis files. The creation of standardized STDM data sets will aid in the creation of analysis files for each individual study, and the future task of integrating data from multiple studies will be accomplished with greater efficiency and quality. The ability to submit SDTM data sets in place of listings or patient profiles, resulting in additional cost reductions.

DEFINING A PROCESS

The degree to which you can define a standard process for converting clinical study data to SDTM domains depends on the environment in which you are working. In an ideal situation, the CDM data structures would be designed to be as compatible as possible with the SDTM specifications. An SDTM annotated CRF is a valuable tool to aid in the mapping process. Creating a standard metadata library would allow you to maximize the consistency within and between studies. This level of consistency would allow you to develop a library of standard annotated CRF pages and a library of SAS macros for creating SDTM domain files with a minimum amount of metadata mapping and additional programming at the study level. This level of standardization would also reduce the cost of consolidating data for integrated studies. In such an environment a very detailed and specific SDTM conversion process can be defined.

In many current situations, existing data does not contain this level of standardization or compatibility with the SDTM standards. In such cases the conversion process must be very flexible and it can only be defined in general terms. Even though the process must be designed with considerable flexibility to accommodate different CDM data structures, it is still important to have a process in place to serve as a general frame work to promote consistency in SDTM domain creation, promote the use of standard terms to enhance communication, and provide guidance to those new to SDTM. Establishing a process will also facilitate the use of standard tools for metadata mapping and documentation, SDTM file creation and SDTM file validation. The focus of this paper is on this second situation, where significant metadata mapping and programming will be necessary.

If a standard process for SDTM conversion does not currently exist, it is important to define one, at least in general terms, prior to starting the conversion. The process definition is a large-scale map that defines the major steps necessary to create the desired SDTM domains from the existing data. Once the major steps are defined, the components of each step can be determined. This will allow you to define dependencies between tasks, determine where there are possibilities for performing steps in parallel, and define the types of tools that will be necessary. The steps listed below outline a basic process for SDTM conversion. Starting with the end in mind, the goal is defined, the current situation is assessed, and a path is defined between the two.

1. Determine which SDTM domains will be created
2. Determine the extent of SDTM compliance in the existing data
3. Implement automaic direct mapping where possible
4. Map remaining source data sets to SDTM domains
5. Map variables in source data sets to SDTM domain variables
6. Determine if SUPPQUAL domain or custom domains will be required
7. Generate SAS programs to perform the data conversion
8. Validate the SDTM data sets
9. Generate DEFINE.XML
10. Validate DEFINE.XML

It is important to adequately document the general process and the specific steps requires for a particular study. This includes revising the documentation if it becomes necessary to modify the process. The documentation will play a critical role in validating the process and will be very useful as a guide during future SDTM conversion projects.

SDTM DOMAINS

A basic understanding of the SDTM domains, their structure and their interrelations is vital to determining which domains you need to create and in assessing the level to which your existing data is compliant. The SDTM consists of a set of clinical data file specifications and underlying guidelines. These different file structures are reffered to as domains. Each domain is designed to contain a particular type of data associated with clinical trials, such as demographics, vital signs or adverse events. In the current specification, each of these domains will be contained in a separate XPORT data file, based on the SAS version 5 data set file format, which is in the public domain. Future versions will support the use of XML files.

The CDISC SDTM Implementation Guide provides specifications for 30 domains and new domains are being developed. It is important to check the CDISK website for the latest updates before you beging a new conversion project. The SDTM domains are divided into six classes. The 21 clinical data domains are contained in three of these classes: Interventions, Events and Findings. The trial design class contains seven domains and the special-purpose class contains two domains (Demographics and Comments). The trial design domains provide the reviewer with information on the criteria, structure and scheduled events of a clinical trail. By placing key trial design information in a concise and standard data structure, the reviewer can have ready access to details of the trial design that allow them to view the clinical data in the proper context. The focus of this paper is on creating clinical data domains from CDM system data files. A list of the SDTM clinical data domains is given below in Figure 1. Only the domains that are pertanent to a particular study need to be created. The only required domain is demographics. Demographics also differs from the other domains in the fact that it has a horizontal structure, with a single row per subject.

There are two other special purpose relationship data sets, the Supplemental Qualifiers (SUPPQUAL) data set and the Relate Records (RELREC) data set. SUPPQUAL is a highly normalized data set that allows you to store virtually any type of information related to one of the domain data sets.The initial specification for SUPPQUAL indicates that a single file should be used for all domains. The current trend, and possibly the requirement for the next version of SDTM, is to use a separate file for each domain named SUPP--, where the hyphens are replaced with the two-letter designation for each domain.

In general, the use of SUPPQUAL should be minimized. Its purpose is to provide a means of adding variables which are critical to a study, but which are not included in the specifications of the pertanent domain and are not suitable as an additional identifier, topic or timimg variable. If the number of additional variables is large or if they are not pertanent to an existing domain, then the creation of a custom domain shuld be consided. Before considering the creation of a custom domain, you should review the latest information on the CDISC Web site, it is possible that a new domain has been defined that will suite your needs. Guidelines for creating custom domains are included in the SDTM Implementation Guide. Information on RELREC is provided in the section below on key variables and relating records.

CDISC SDTM DOMAINS

CLASS	DOMAIN NAME	DOMAIN DESCRIPTION
Special Purpose	DM	Demographics
	CO	Comments
Interventions	CM	Concomitant Medications
	EX	Exposure
	SU	Substance Use
Events	AE	Adverse Events
	DS	Disposition
	DV	Protocol Deviations
	MH	Medical History
Findings	DA	Drug Accountability
	EG	ECG
	IE	Inclusion / Exclusion Criteria Exceptions
	LB	Laboratory Results
	MB	Microbiology Specimens
	MS	Microbiology Susceptibility
	PC	Pharmacokinetic Concentrations
	PP	Pharmacokinetic Parameters
	PE	Physical Exam
	QS	Questionnaires
	SC	Subject Characteristics
	VS	Vital Signs
Trial Design	TE	Trial Elements
	TA	Trial Arms
	TV	Trial Visits
	SE	Subject Elements
	SV	Subject Visits
	TI	Trial Inclusion/Exclusion Criteria
	TS	Trial Summary
Relationship Data Sets	SUPPQUAL	Supplemental Qualifiers
	RELREC	Relate Records

Figure 1. CDISC SDTM Domains

GENERAL GUIDELINES ON SDTM VARIABLES

Each of the SDTM domains has a collection of variables associated with it. There are five roles that a variable can have: Identifier, Topic, Timing, Qualifier, and for trial design domains, Rule. Using lab data as an example, the subject ID, domain ID and sequence (e.g. visit) are identifiers. The name of the lab parameter is the topic, the date and time of sample collection are timing variables, the result is a result qualifier and the variable containing the units is a variable qualifier. The SDTM guidelines contain a section on the fundamentals of the SDTM that cover this topic in detail. The SDTM fundamentals are important to understand before you begin the process of metadata mapping, particularly if you need to create custom domains.

Variables that are common across domains include the basic identifiers study ID (STUDYID), a two-character domain ID (DOMAIN) and unique subject ID (USUBJID). In studies with multiple sites that are allowed to assign their own subject identifiers, the site ID and the subject ID must be combined to form USUBJID. All other variable names are generally formed by prefixing a standard variable name fragment with the two-character domain ID.

It is also important to understand which variables should be included in each domain to which you will be mapping study metadata. The SDTM specifications do not require all of the variables associated with a domain to be included in a submission. The SDTM is a standard designed to accommodate the wide range of trials that are conducted in the Pharmaceutical and Biotechnology industries, and some variable may not be necessary for a particular trial. Your metadata mapping will not necessarily include all of the variables associated with the domains you are creating nor will it necessarily include all of the variables contained in the CDM database. Any questions regarding which variables to submit should be addressed with your reviewer. In regard to complying with the SDTM standards, the implementation guide specifies each variable as being included in one of three categories: Required, Expected, and Permitted. An explanation of each is given below.

REQUIRED – These variables are necessary for the proper functioning of standard software tools used by reviewers. They must be included in the data set structure and should not have a missing value for any observation.

EXPECTED – These variables form the fundamental core of information within a domain. They must be included in the data set structure; however it is permissible to have missing values.

PERMISSIBLE – These variables are not a required part of the domain and they should not be included in the data set structure if the information they were designed to contain was not collected.

The implementation guide provides information on the expected structure of each domain data set. For each variable, a name, label and type are provided. The length of the variables is not specified. The file structure is designed to comply with the XPORT file format, which is based on the SAS version 5 data set specifications. Variable names have a maximum length of 8, labels a maximum length of 40 and character variables a maximum length of 200. These restrictions may change in the future as the use of XML becomes standard.

To accommodate character variables longer than 200, the first 200 characters should be stored in the domain variable and the remaining text should be stored in the SUPPQUAL domain. For the sake of readability, the text from the source variable should be split between words, into substrings of length 200 or less. The first substring is stored in the appropriate variable in the parent domain. Each of the remaining substrings should then be stored in the variable QVAL in an observation within SUPPQUAL. In SUPPQUAL, the variable QLABEL should contain the same label as the domain variable and the variable QNAM should contain the name of the variable in the parent domain with a sequential integer from 1 to 9 appended. If the name of the parent domain variable has a length of 8 then the sequential number replaces the last character of the name. The variable IDVAR and IDVARVAL are used to relate the records in SUPPQUAL back to the appropriate record in the parent domain.

In addition, some variables require the specification of a controlled terminology or format. In such cases, the implementation guide specifies whether the controlled terminology is provided by an external source (e.g. MedDRA) or by the investigator. It is generally recommended that the text used in defining controlled terminology be placed in all uppercase. Exceptions to this rule are controlled terminology from external sources or designations such as units, which employ a generally accepted use of mixed case text. When defining controlled terminology, it is important to prevent ambiguity.

MAPPING EXISTING DATA TO SDTM DOMAINS

Before beginning the task of developing programs to create SDTM domain data sets from your existing data, it is important to have a “road map” to design and document the process. As with planning any journey, the first step is to specify your current location and the location of your destination. By comparing alternate routes before starting the actual trip, you can avoid getting lost or needing to back track.

The first step in the mapping process involves the comparison of the study metadata with the SDTM domain metadata. If the CDM metadata is compliant to a significant extent with the SDTM metadata, it is possible to use

automated mapping as a first pass. If CDISC standard data set and variable names were properly used in the CDM data sets, it is possible to use a DATA step merge or SQL join to combine rows of study metadata with matching rows of SDTM metadata based on variable name, type and format. Note that the SDTM standards do not specify variable length. They do provide the standard variable label, so it is important to make sure you are keeping the SDTM label rather than your CDM data label. Automatic mapping can potentially results in a significant reduction in cost, however it is important to check the validity of the mappings. This process only serves as a first pass of metadata mapping, in most cases some manual mapping will be necessary. If the CDM metadata is not compliant with the SDTM or worse yet, if SDTM specifications were improperly used, then auto mapping should be avoided.

The next step involves manually mapping the study data sets to the domain data sets and then mapping each individual variable to the appropriate domain. Depending on how the CDM data sets are structured, you may map each CDM file to a single domain, split its variables among multiple domains, or combine variables from multiple CDM files into a single domain. There are several possible types of variable mappings. In some cases it may be necessary to use more that one method in order to create the desired SDTM variable from the existing data. A list of basic variable mappings is given below.

DIRECT a CDM variable is copied directly to a domain variable without any changes other than assigning the CDISC standard label

RENAME only the variable name and label may change but the contents remain the same

STANDARDIZE mapping reported values to standard units or standard terminology

REFORMAT the actual value being represented does not change, only the format in which is stored changes, such as converting a SAS date to an ISO8601 format character string

COMBINING directly combining two or more CDM variables to form a single SDTM variable

SPLITTING a CDM variable is divided into two or more SDTM variables

DERIVATION creating a domain variable based on a computation, algorithm, series of logic rules or decoding using one or more CDM variables

While any mapping that involves changing or combining CDM variables to form a domain variable could be referred to as a derivation, further categorizing the type of mapping facilitates assigning a standard process (e.g. a SAS macro or block of SAS source code) to perform the mapping operation.

Effective manual mapping requires a method of managing and accessing the metadata for both your existing data and the SDTM domains. If your study data resides in SAS data sets, and you define a SAS library for their location, SAS will automatically provide a view to an internal table that contains the structure information for all data sets in any defined library. This metadata can be easily accessed by either specifying SASHELP.VCOLUMN as an input data set in a DATA step, or by selecting rows and columns from the table DICTIONARY.COLUMNS using PROC SQL. This file contains the library name, data set name, variable name, type, length, label, format and more for every variable in every data set in every currently defined library. The amount of information in this view can be overwhelming and it is usually necessary to use a where clause to obtain only the specific information needed. The fact that it contains metadata for all currently accessible data sets facilitates easy metadata comparisons across data sets or across studies, such as determining which variables have identical or similar names.

KEY VARIABLES AND METHODS OF RELATING RECORDS

Every domain contains a required set of variables that form a unique key for that record. These include STUDYID, DOMAIN and USUBJID. DOMAIN contains the two-character domain name and is hard-coded into each record. USUBJID is a unique subject identifier within a study. Therefore, if multiple sites are used and subject numbers overlap between sites, then USUBJID must combine the initial site and subject numbers. An additional required key variable is –SEQ, where the two hyphens represent the domain name. When a subject has more than one record in a domain, then –SEQ is used to form a unique key. An additional, sponsor-defined key is –SPID. This variable is typically used for external identifiers, such as a sample number assigned by a lab.

The SDTM design provides several ways to relate records within and between domains. Records within a domain can be related by assigning them the same value for –GRPID. The RELREC data set can be used to relate multiple records in multiple domains. Each record in RELREC with the same value of RELID defines a relation. Each record also contains the key variables necessary to point to a record or group of records in a domain.

CDISC SDTM METADATA MAPPING TOOLS

The use of software tools is essential to the efficient creation of SDTM data sets. The process of mapping study data to the SDTM domains can be complex. The large number of variables involved and the many different

transformations required make mapping without a tool tedious and error prone. When decisions are made regarding process steps, it is important that the process be documented for consistency and repeatability. Direct electronic access to metadata for both the study data and the SDTM domains facilitates an efficient mapping process. Automation of basic processes can save significant amounts of time. Metadata about the mapping process can be used to generate documentation of the process and to generate the SAS source code to perform the derivation of domain data sets. Once the domain data sets have been produced, software tools documenting the metadata mapping can improve the efficiency of validating the domain data sets and producing the define.xml file.

The use of a metadata mapping tool can also be extended to the creation of ADaM analysis data sets from the SDTM data sets. A typical ADaM data set is created by merging data from two or more SDTM data sets, restructuring the data to a form convenient for analysis and creating derived variables. The use of a metadata mapping tool for creating ADaM data sets will provide similar advantages to those for producing SDTM data sets. Including metadata on both transformations in one system will provide complete documentation of the creation of the analysis data sets. The process by which each variable was created can be traced back to the original source. This approach will also simplify maintaining consistency between the SDTM and ADaM data sets. The CDISC specifications state that any variables copied from an SDTM domain into an ADaM data set must retain all of the attributes found in the SDTM domain. By storing the metadata for SDTM and ADaM in the same system and form it is easy to ensure that this condition will be met.

The SAS® Metadata Server and the SAS® Data Integration Studio provide a very powerful environment for mapping study data and producing domain data sets. This environment provides direct access to study metadata and CDISC SDTM domain metadata. The visual interface allows you to define data transformation and mapping steps using icons that represent predefined process steps. The system is extensible, allowing you to add new capabilities and the sequence of steps used in your process is stored in metadata.

DEVELOPING A SDTM METADATA MAPPING TOOL

It is possible to create your own simple, but effective tools to aid in the metadata mapping process. Leveraging the power of SAS and Microsoft Excel together allows you to create a practical metadata mapping tool with relatively little programming. The combination of SAS and Excel allows you to combine a user interface with the familiarity of an Excel workbook with the power of SAS to access and manipulate data in a variety of forms. Important skills needed to develop such a tool includes a solid understanding of SAS DATA step programming, basic SAS Macro programming skills, and a working knowledge of Visual Basic and the Excel object model.

A key reason for the power of pairing SAS with Excel is the flexibility SAS provides for exchanging data with Excel. The SAS Excel libname engine allows you to read and write from Excel worksheets as though they were a SAS data set. The IMPORT and EXPORT procedures allow you exchange data for an entire data set as a stand-alone process or from within a SAS program. Dynamic data exchange (DDE) allows you to define a DDE triplet that defines a range of cells in Excel to be treated as a flat file in SAS. The SAS® Add-In for Microsoft Office allows you to use SAS as a powerful data access, manipulation and analysis back end for Excel applications. SAS also provides the XML libname engine to facilitate reading and writing XML files. In version 9, SAS added ODM native mode support (xmltype = CDISCODM) to the XML engine. The SAS CDISC procedure currently provides read and write capability for ODM, and content and structure validation for SDTM.

The example presented here is a simple tool developed using Microsoft Excel, Visual Basic and SAS. The SDTM metadata mapping tool allows users to manage and document the mapping of study data to SDTM domains and it can produce text files containing SAS source code to be used as a starting point for programs to generate SDTM domain data sets from the study data sets. The tool consists of an Excel workbook with three main worksheets: an SDTM domain metadata dictionary, a study metadata dictionary with CDM data set specifications imported from the SAS view SASHELP.VCOLUMN, and a SDTM mapping sheet containing variable mapping and derivation information.

An advantage of using Excel is that there is a great deal of functionality available without any programming. One example of this is the Excel auto filter. When an auto filter is set for a column, a selection button appears in the label cell. Clicking on it displays a pick list containing all of the unique items in that column. If an item is selected, the sheet will then only display rows that contain that value in that column. This feature makes it easy to view subsets of the metadata. For example, you can view all of the variables in a particular data set or domain, or you can view all of the occurrences of a given variable name across all domains. The sheet containing the SDTM metadata dictionary is shown in Figure 2, the study metadata sheet is shown in Figure 3.

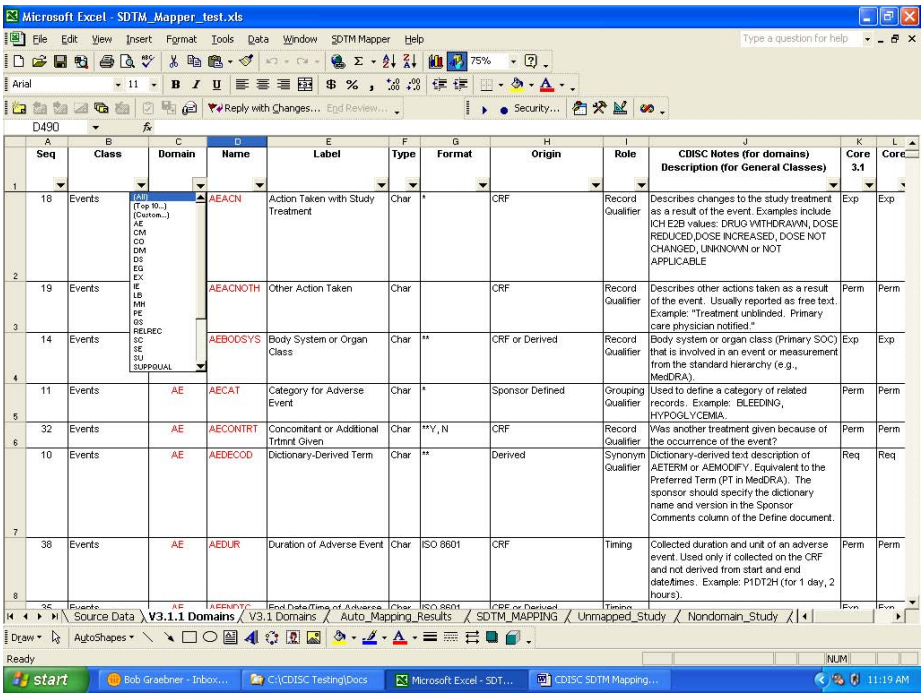

Figure 2. SDTM Metadata dictionary with auto filter selection list

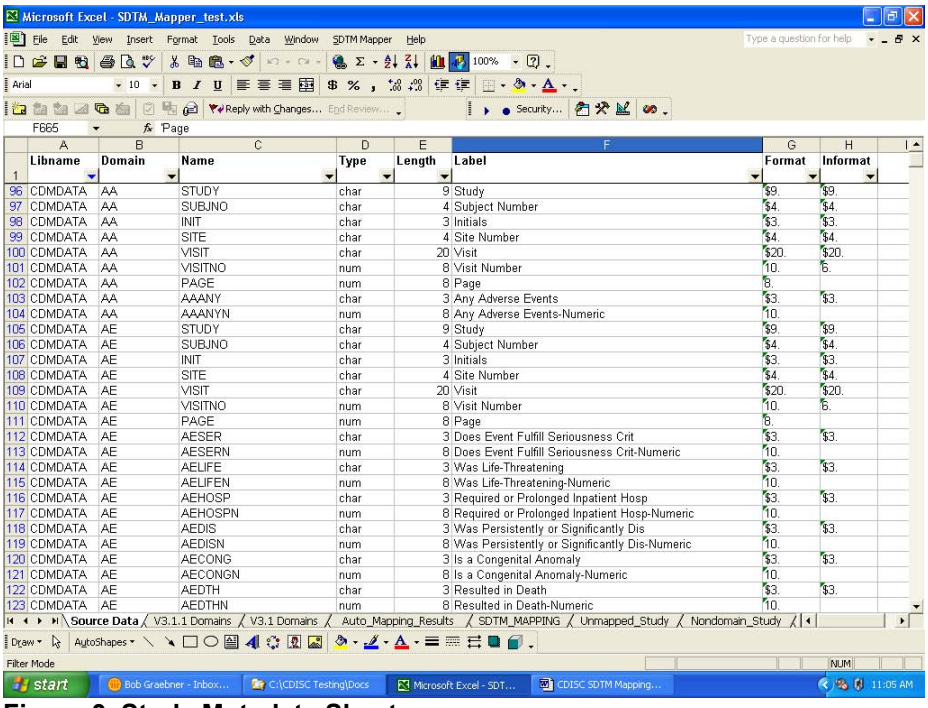

Figure 3. Study Metadata Sheet

The user interface includes a new main item called **SDTM Mapper** that is temporarily added to the Excel main menu just before the **Help** menu item, or within the Add-Ins tab if you are using Excel 2007. Current active submenu items include **Map Study Variables** and **Generate SAS Code for Domain**. The functionality behind these menu options is provided by a series of Visual Basic modules containing subroutines and functions stored within the workbook. Mapping study variables involves selecting the row corresponding to a given SDTM domain variable in the SDTM_MAPPING sheet, then selecting the desired study variable from a pick list that uses the study metadata dictionary as its row source. Once a variable is selected, the metadata for that variable is added to the same row in the appropriate columns of the SDTM_MAPPING sheet. The names of the study data metadata columns all begin

with ‘s_’ to differentiate them from the columns containing metadata for the STDM domain. If additional study variables are required to derive an SDTM domain variable, they can be added to the s_addvars column. Blocks of executable SAS source code or a SAS macro call can be entered into the SAS_code column. The SAS code is included the SAS program text files that are generated by the mapping tool and it also provides documentation on how the variable was created. If only basic instructions or pseudo code are available, they can be entered as a SAS comment statement. A valuable addition to this sheet would be a column to containing the derivation or imputation description or algorithm. This would ensure that the method used to create a variable can be easily understood by those who do not program and the contents could serve as a source for ComputationMethod items in the define.xml file. The mapping sheet with the variable selection user form is shown in Figures 4 and 5.

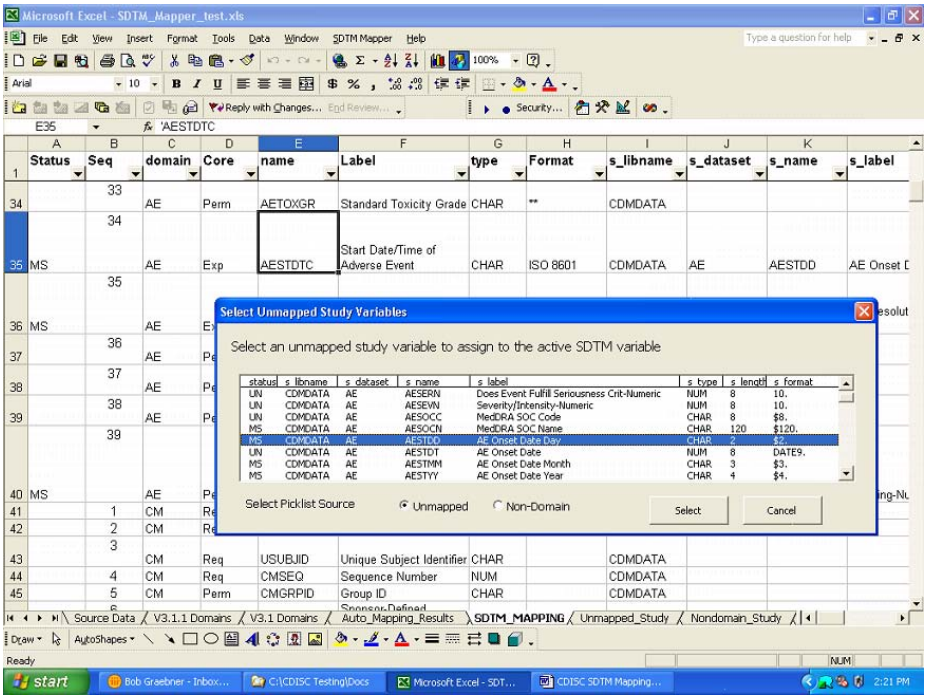

Figure 4. Metadata mapping sheet showing the study variable pick list

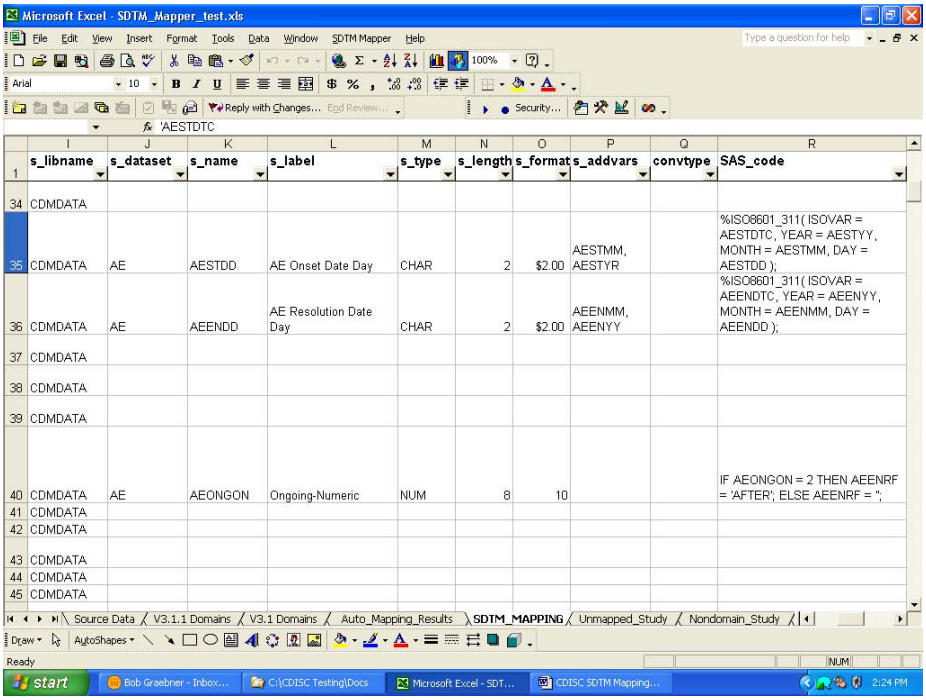

Figure 5. Metadata mapping sheet showing additional variables and SAS code

To generate a text file containing SAS source code, the user selects Generate SAS Code for Domain from the SDTM Mapper menu and then selects the desired domain. A Visual Basic module utilizes the metadata in the SDTM_MAPPING sheet to generate a text file with SAS source code that includes:

- A program header comment block which indicates the name of the SDTM domain is produced and the names of the source data sets
- RETAIN and KEEP statements containing all of the selected variable names
- A LABEL statement containing the name and standard CDISC label for all selected variables
- A DATA step to create the domain data set with a SET statement if it is created from a single source data set or a MERGE statement if it is created from two or more data sets
- All blocks of SAS source code from the relevant rows of the SDTM_MAPPER sheet

Because the metadata is used to generate the SAS source code you will end up with code that includes all of the necessary variables, with correct names, labels and types. While the text file is not meant to be a read-to-run program, it helps increase efficiency and consistency by eliminating most of the tedious tasks associated with developing conversion programs and allows the programmer to focus on the challenging issues of data mapping and derivation.

A simple application like this can be useful in situations where timelines are tight and do not afford the opportunity to develop a full-scale application. It is designed as a flexible, “in the trenches” tool. In addition to filling immediate project needs, such an application can serve as a prototype for testing new ideas and as a focal point while defining and refining the user requirements for a more robust, enterprise-level application. When using an Excel application of this type, it is important to limit the extent to which users and modify the functionality. The most critical safeguard is to password protect the Visual Basic source code modules so that only those with sufficient skill and adequate knowledge of the application can modify them.

DOMAIN DATA SET VALIDATION

The SAS CDISC procedure is a very valuable tool for validating SDTM domain data sets is. With SAS version 9.1.3, Proc CDISC can be used to validate domain data sets. Future version will provide additional functionality. For validating STDM domain data sets, I developed a SAS macro that utilizes PROC CDISC. The macro has three parameters:

- DOMAIN - The two-letter of the SDTM domain to validate
- SUPPQUAL - If this parameter is not missing, the SUPPQUAL data set is validated
- COMM - If this parameter is not missing, the comments (CO) data set is validated

Only the domain name is required. The category parameter of PROC CDISC is automatically set by the macro. If the SUPPQUAL parameter is not missing, then the rows in SUPPQUAL that pertain to the specified domain are test merged with the domain data set. An error statement is generated in the SAS log for any SUPPQUAL rows that do not have a match in the domain data set. The same process is done with the comments data set if the COMM parameter is not missing. Any findings from PROC CDISC are also included in the log. This can include:

- An ERROR for any required variable that is not found or has a missing value, or any expected variable that is not found
- A WARNING for any expected variable that has a missing value
- A NOTE for any permissible variable that is not found

Note that unless you have the Beta patch for PROC CDISC, the SDTM 3.1.1 ISO8601 format is not supported and dates with missing components will generate an error in the log.

DATES, TIMES AND THE ISO8601 FORMAT

The CDISC standard uses the nonproprietary ISO8601 format to represent date and time values. This standard expresses dates and times with character strings in a format that can readily be understood by humans and interpreted by software. A full representation of a date and time value would be of the form YYYY**-**MM**-**DD**T**hh**:**mm**:**ss. Years are represented using four digits, the remaining date and time components are all two digits with leading zeros if necessary. The date components are separated by a hyphen and the time components are separated by a colon. For values containing date and time, an upper case letter ‘T’ is used to separate the date and time. There are no spaces between components and delimiters. The ISO8601 standard allows for the use of either the basic format, without delimiters, or the extended format described above. The SAS XML libname engine provides both basic and extended formats and informats. The CDISC specification requires the use of the extended format with delimiters.

Partial dates and times can be stored in this format, however the ISO8601 standard of handling partial dates was modified. In the original standard, the representation would start with the largest scale component (e.g. year) and continue until a missing component occurred. The representation would end at that point, resulting in a reduced precision representation. For example, if a date was recorded with a year and day, but missing month, it would only be stored in ISO8601 format as a year. With the new standard, hyphens could be inserted for the two missing month digits, resulting in a missing component representation. The SDTM 3.1 standard utilizes the reduced precision method, the 3.1.1 standard uses the missing component standard. The current version of SAS 9 was developed based on the SDTM 3.1 implementation guide however there are updates available to comply with the SDTM 3.1.1 implementation of the ISO8601 date and time formats. The examples below show the full representation of 10:30 AM on March 3, 2008, and the partial representations if the day was missing.

Full Datetime Representation:	2008-03-18T10:30:00
Reduced Precision Representation (SDTM 3.1)	2008-03
Missing Component Representation (SDTM 3.1.1)	2008-03—T10:30:00

There are many features in SAS that facilitate reading and writing dates and times in the ISO8601 format. SAS provides a wide range of ISO8601 date and time formats and informats with the XML libname engine. When working with SAS data sets there are several informats that can be used to read ISO8601 text strings in as a SAS date. This might be necessary if the ISO8601 formats were used in creating the source data sets and you need to perform computations or comparisons of dates to create your SDTM domains. Partial dates or times will result in a missing value for the SAS date or time variable. The applicable SAS informats are listed below.

Reading ISO8601	Dates:	ANYDTDTE10. or YYMMDD10.
	Times:	ANYDTTME8. or TIME8.
	Datetime:	ANYDTDTM19.

SAS provides many functions that are useful in creating dates and times in the ISO8601 format. The individual date and time components can be extracted, formatted and combined with the appropriate delimiter characters to form the equivalent ISO8601 representation.

DEFINE.XML

FDA guidance for electronic submissions specifies that all electronic submissions include a Data Definition Document that describes the structure and content of the data included in the submission. In 1999 the FDA standardized on the use of SAS version 5 XPORT (.XPT) files for study data, and Portable Document Format (.PDF) files for metadata. In 2003 the FDA expanded the list of acceptable file types to include Extensible Markup Language (.XML) files. By transitioning from the use of define.pdf to define.xml, the metadata for the submission will be in a machine-readable form that can be used by standard data review tools. Placing both study data and metadata in a standard XML schema will facilitate validation and transfer into a data warehouse. The schema for the SDTM define.xml is based on an extension of the CDSIC ODM, which is a specification of a standard XML schema designed to facilitate efficient and robust storage and interchange of clinical trial data and associated metadata. Details on define.xml are published in the Case Report Tabulation Data Definition Specification (CRT-DDS) document available at the CDISC website listed at the beginning of this paper. CDISC also provides standard style sheets that can be used to render the define.xml file into a readable form. The United States Food and Drug Administration (FDA) also provides guidance on preparing files for electronic submission.

The creation of the define.xml file must conform to the CDISC standards. The XML must be well-formed, standard XML without any proprietary XML tags, such as you can find in an Excel file saved as XML. The XML specification does not define a single file structure definition as is common with proprietary file formats such as SAS data sets or Excel spreadsheets. The ‘X’ stands for extensible. Within the XML specification, matching tags are used to delimit items. In XML however, it is possible to define new tags to meet specific needs. It is essential that the tags used conform to the CDISC ODM standard.

The define.xml file is comprised of several sections. The file header contains information that identifies the file as XML and specifies the XML version used. The file also contains SDTM study-level metadata. The table of contents section contains domain-level metadata including the data set name for each domain, a description, structure description (e.g. one record per subject per event), the purpose, a list of the variables that form the key and a link to the actual data set. Another section is the Data Definition Table (DDT) that contains the variable-level metadata. Validation of the define.xml must be done on several levels including checks for conformance with the define.xml specification, checks for internal integrity between elements and checks for external integrity with other files referenced in define.xml such as domain data files and an annotated CRF in PDF format.

CONCLUSION

The mapping of existing study data to CDISC SDTM domain data sets can be a daunting task. Developing an adequate understanding of the SDTM standard is an important first step. Proper planning and the use of metadata mapping tools can increase both the efficiency of the process and the quality of the resulting data sets. The use of standard processes and tools will increase the return on your development investment if they are flexible enough to be used on future conversion projects. If you are allowed to submit SDTM domain data sets in lieu of study report listings, patient profiles or monitoring board report listings, the cost of creating the STDM domain data sets can be offset. The ability of reviewers to readily access tabulation data can potentially eliminate some of the costs associated with ad-hoc requests. Having you study data in a standardized format can facilitate significant gains in efficiencies when creating analysis file data sets or when combining data from different trials for an integrated study.

CONTACT INFORMATION

Robert Graebner
Quintiles, Inc.
P.O. Box 9708
Overland Park, KS

Email: bob.graebner@quintiles.com
mgraebner@kc.rr.com