Paper FC05

Strategies and Practical Considerations for Creating CDISC SDTM Domain Data Sets from Existing CDM Data Sets

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. However if the process is well planned and properly managed, successful results can be obtained in an efficient manner. Standardizing and mapping data elements from a form well suited to use with a clinical data management system to a form suitable for tabulation and review has many aspects to consider. When data from multiple protocols with non-standard data structures must be combined into one submission, such as for an Integrated Summary of Safety, the use of a consistent approach is critical for success. A well-planned and documented process can provide significant increases in efficiency on future conversion projects. Tools suited to the task of complex data mapping can significantly reduce the cost and improve quality. The first step to a successful conversion is to gain an adequate understanding of the STDM standards. The examples in this paper are based on the CDISC SDTM version 1.1 and the SDTM Implementation Guide version 3.1.1.

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 considerations 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 (**http://www.cdisc.org/standards/index.html**). The SDTM Implementation Guide is an essential tool for anyone involved with the metadata mapping or programming associated with the creation of SDTM data sets. A basic overview of the SDTM specifications is provided in this paper as a starting point.

SDTM OVERVIEW

The purpose of creating 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. 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 possible, and 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. Improper use of CDISC standards, such as using a valid domain or variable name incorrectly, can equally hinder the metadata mapping process and should be avoided. 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 adding many additional derived variables and conversion to a more horizontal structure. The SDTM data sets can however, be used in the creation of analysis files. The creation of standardized STDM data sets can aid in the creation of analysis files for each individual study, and the future task of integrating data from multiple studies can accomplished with greater efficiency and quality. As an additional benefit, it is possible to submit SDTM data sets in place of programmed listings or patient profiles, resulting in reduced costs.

SDTM DOMAINS

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 likely support the use of XML files.

The SDTM currently consists of 30 data 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, but not included in, one of the domain data sets. 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. 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. 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 series of variables designed to be included in that domain. 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, 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.

Not all of the variables included in the specifications are required 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. It is important to remember that not all of the variables in a domain need to be provided. Conversely, it is not necessary to include all variables from your CDM data in the SDTM domains. 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. On studies where the CDM metadata was compliant to the extent possible with 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 or SQL select to check for direct matches and document them so that the validity of the mappings can be checked. This process only serves as a first pass of metadata mapping, but it can potentially eliminate hours of tedious manual mapping. If the CDM metadata is not compliant with the SDTM or worse yet, if SDTM specifications were improperly used, then auto mapping is probably best 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 was 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 mapping possible. Several basic types are listed below.

DIRECT – In direct mapping, a CDM variable is copied directly to a domain variable without any changes other than assigning the CDISC standard label. The labels can be pulled directly from a CDSIC metadata table, so the process is simple.

RENAME – The variable name and label may change but the contents remain the same.

SCALING – A scaling mapping involves direct conversions, such as converting values to standard units.

REFORMAT – The actual value does not change, only the representation changes, such as converting a date to ISO8601 format.

COMBINING – This type of mapping involves directly combining two or more CDM variables to form a single SDTM variable.

SPLITTING –This type of mapping is used when a domain variable consists of a portion of a CDM variable.

DERIVATION – A derivation mapping involves creating a domain variable based on a computation, algorithm or series of logic rules from 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) 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, you can access the metadata in SASHELP.VCOLUMN. This file is automatically created by SAS and it is a view to an internal data set that contains the library name, data set name, variable name, type length label and format for every variable in every data set in every currently defined library.

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.

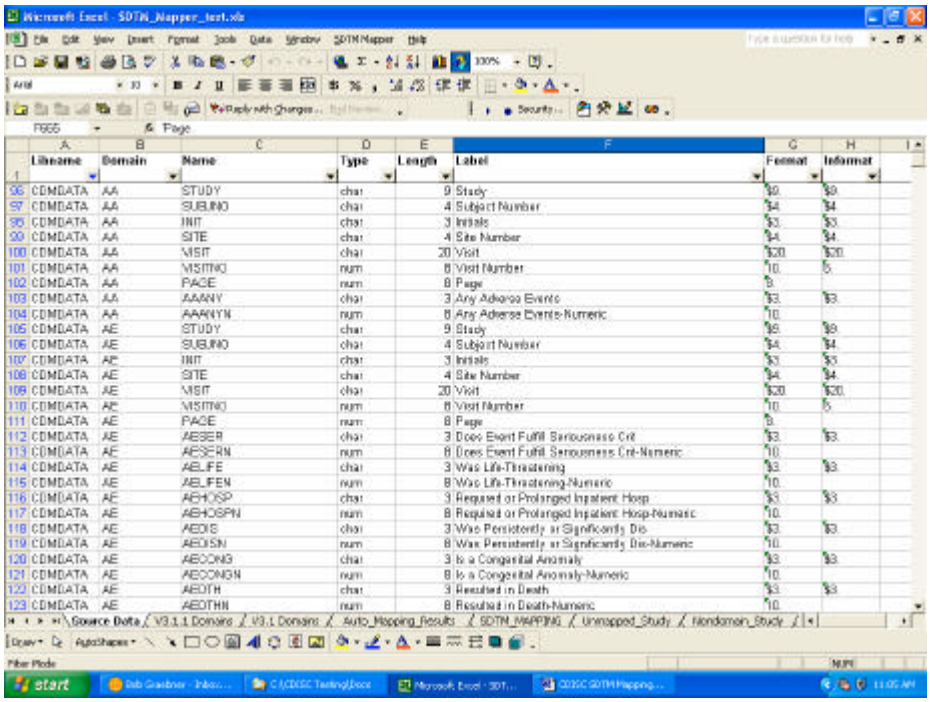
The SDTM design provides several ways to relate records within and between domains. Records within a domain can be related by assigning them the same value for –GRPID. The RELREC data set can be used to relate multiple records in multiple domains. Each record in RELREC with the same value of RELID defines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

The use of software tools can greatly increase the efficiency of mapping study data to SFTM domains and producing the domain data sets. The process of mapping study data to the SDTM domains can be complex. 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 on the mapping process can be used to generate documentation of the process and to generate the SAS source code necessary to perform the derivation domain data sets. Once the domain data sets have been produced, software tools can be invaluable in validating that the domain data sets produced conform to the SDTM standard, and in producing the Define.XML file.

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.

It is possible to create your own simple, but effective tools to aid in the metadata mapping process. I use a tool I developed using Microsoft Excel® and VBA macros. The SDTM metadata mapping tool allows users to manage and document the mapping of study data to CDISC SDTM domains and it produces 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 worksheets containing SDTM domain metadata, study metadata obtained from the SAS view SASHELP.VCOLUMN, and mapping/derivation information. The study metadata sheet is shown in Figure 2.



	Libname	Domain	Name	Type	Length	Label	Format	Informat
96	CDMDATA	AA	STUDY	char	9	Study	$9.	$9.
97	CDMDATA	AA	SUBJNO	char	4	Subject Number	$4.	$4.
98	CDMDATA	AA	INIT	char	3	Initials	$3.	$3.
99	CDMDATA	AA	SITE	char	4	Site Number	$4.	$4.
100	CDMDATA	AA	VISIT	char	20	Visit	$20.	$20.
101	CDMDATA	AA	VISITNO	num	8	Visit Number	10.	5
102	CDMDATA	AA	PAGE	num	8	Page	8.	
103	CDMDATA	AA	AAANY	char	3	Any Adverse Events	$3.	$3.
104	CDMDATA	AA	AAANYN	num	8	Any Adverse Events-Numeric	10.	
105	CDMDATA	AE	STUDY	char	9	Study	$9.	$9.
106	CDMDATA	AE	SUBJNO	char	4	Subject Number	$4.	$4.
107	CDMDATA	AE	INIT	char	3	Initials	$3.	$3.
108	CDMDATA	AE	SITE	char	4	Site Number	$4.	$4.
109	CDMDATA	AE	VISIT	char	20	Visit	$20.	$20.
110	CDMDATA	AE	VISITNO	num	8	Visit Number	10.	5
111	CDMDATA	AE	PAGE	num	8	Page	8.	
112	CDMDATA	AE	AESER	char	3	Does Event Fulfill Seriousness Crit	$3.	$3.
113	CDMDATA	AE	AESERN	num	8	Does Event Fulfill Seriousness Crit-Numeric	10.	
114	CDMDATA	AE	AELIFE	char	3	Was Life-Threatening	$3.	$3.
115	CDMDATA	AE	AELIFEN	num	8	Was Life-Threatening-Numeric	10.	
116	CDMDATA	AE	AEHOSP	char	3	Required or Prolonged Inpatient Hosp	$3.	$3.
117	CDMDATA	AE	AEHOSPN	num	8	Required or Prolonged Inpatient Hosp-Numeric	10.	
118	CDMDATA	AE	AEDIS	char	3	Was Persistently or Significantly Dis	$3.	$3.
119	CDMDATA	AE	AEDISN	num	8	Was Persistently or Significantly Dis-Numeric	10.	
120	CDMDATA	AE	AECONG	char	3	Is a Congenital Anomaly	$3.	$3.
121	CDMDATA	AE	AECONGN	num	8	Is a Congenital Anomaly-Numeric	10.	
122	CDMDATA	AE	AEDTH	char	3	Resulted in Death	$3.	$3.
123	CDMDATA	AE	AEDTHN	num	8	Resulted in Death-Numeric	10.	

Figure 2. Study Metadata Sheet

The Excel AutoFilter 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 is shown in Figure 3.

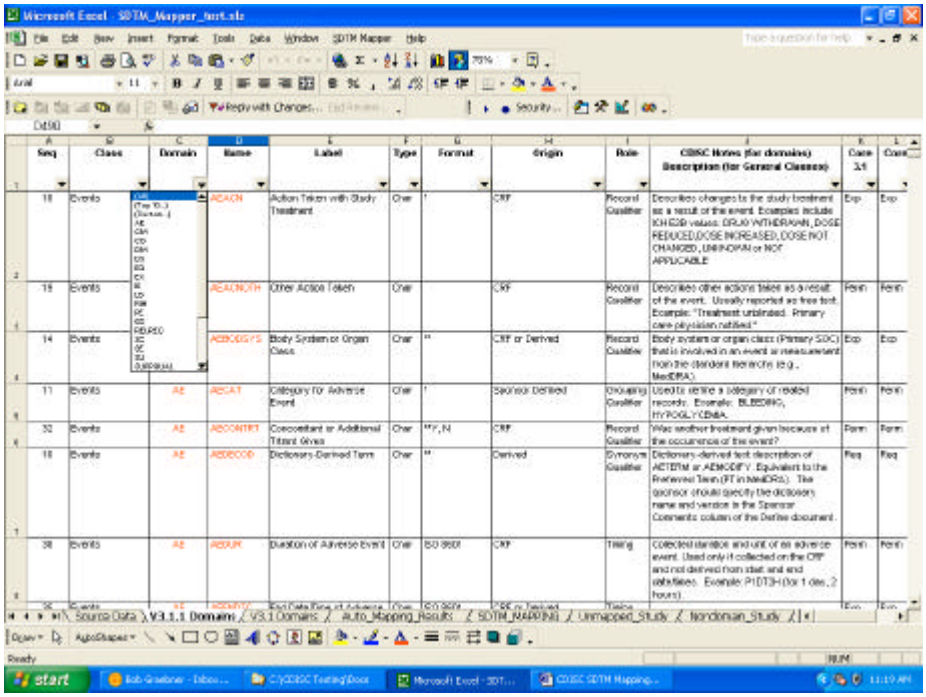


Figure 3. SDTM Metadata sheet with Autofilter selection list

The user interface includes a new menu item called **SDTM Mapper** that is temporarily added to the Excel main menu. Current active menu 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 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 corresponding study variable from a pick list. 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_'. If additional study variables are required to derive an SDTM domain variable, they can be added to the s_addvars column. Blocks of SAS source code can be entered into the SAS_code column and they will be included the programs that are generated by the mapping tool. The mapping sheet with the variable selection user form is shown in Figures 4 and 5.

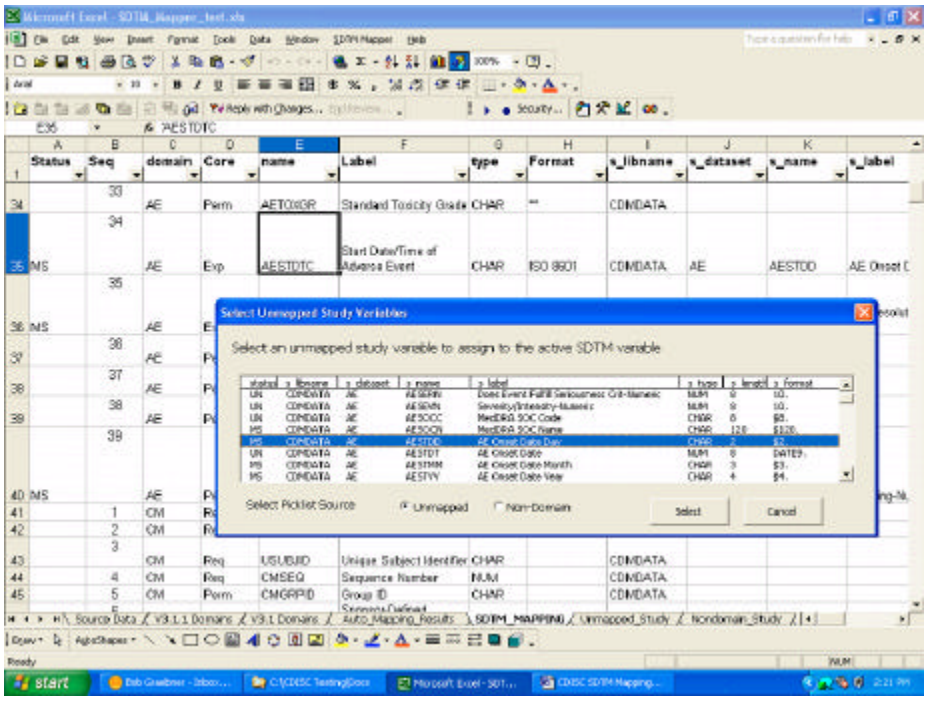


Figure 4. Metadata mapping sheet and variable selection list

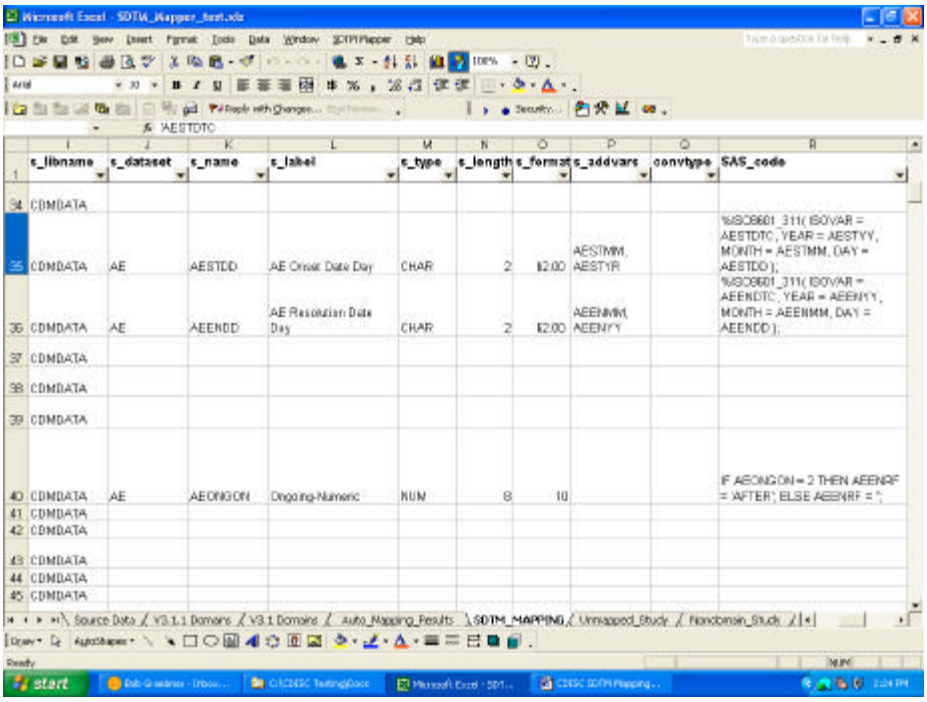


Figure 5. Metadata mapping sheet showing additional variables and SAS code

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
- RETAIN and KEEP statements containing all of the selected variable names
- A LABEL statement containing the name and 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 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 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.

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 data and time format to represent dates and time. 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 would be of the form YYYY-MM-DDThh:mm:ss. In the ISO8601 standard the delimiters are optional, in CDISC they are required. Hyphens separate the year, month and day. An upper case letter ‘T’ separates the date and time, and colons separate the hour, minutes and seconds. 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 offers formats for the ISO8601 date format, however they are applied to SAS dates, which cannot have a missing component due to their numerical representation. It is possible to develop simple SAS macros to handle both ISO8601 standards. An example of such a macro is given below.

```
%macro ISO8601_311(isovar=, year=, month=, day=, hour=, min=, sec=);
     %if &isovar = %then %put **** ERROR: ISOVAR PARAMETER MUST BE SPECIFIED WHEN
   CALLING THE MACRO ISO8601 ****;
     %else
       %do;
         length &isovar $19;
         /**** DETERMINE IF RESULT WILL INCLUDE A DATE PORTION ****/
```

```
%if &year ^= or &month ^= or &day ^= %then
        %do;

          %if &year ^= %then length iso&year $4;;
          %if &month ^= %then length iso&month $3;;
          %if &day ^= %then length iso&day $2;;
          %if &year ^= %then

            %do;
              if &year = '' then iso&year = '-';
                            else iso&year = &year;
              &isovar = strip( iso&year );
            %end;

          %if &month ^= %then
            %do;

              if &month in: ('   ', '...', 'UNK') then iso&month = '-';

              else
                do;
                  select( upcase( substr( strip( &month ), 1, 3 ) ) );
                    when('JAN') iso&month = '01';
                    when('FEB') iso&month = '02';
                    when('MAR') iso&month = '03';
                    when('APR') iso&month = '04';
                    when('MAY') iso&month = '05';
                    when('JUN') iso&month = '06';
                    when('JUL') iso&month = '07';
                    when('AUG') iso&month = '08';
                    when('SEP') iso&month = '09';
                    when('OCT') iso&month = '10';
                    when('NOV') iso&month = '11';
                    when('DEC') iso&month = '12';
                    otherwise   iso&month = &month;
                  end;
                end;

              &isovar = strip( &isovar ) || '-' || strip( iso&month );

            %end;

            %if &day ^= %then
              %do;

                if &day in: ('  ', '..', 'UN') then iso&day = '-';
                                               else iso&day = &day;

                &isovar = strip( &isovar ) || '-' || strip( iso&day );

              %end;

        %end;
```

```
         %if &hour ^= or &min ^= or &sec ^= %then
           %do;

             %if &hour ^= %then length iso&hour $2;;
             %if &min ^= %then length iso&min $2;;
             %if &sec ^= %then length iso&sec $2;;
             if &isovar ^= '' then &isovar = strip( &isovar ) || 'T';

             %if &hour ^= %then
               %do;
                  if &hour = '' then iso&hour = '-';
                                else iso&hour = &hour;
                  &isovar = strip( &isovar ) || strip( iso&hour );
               %end;

             %if &min ^= %then
               %do;
                 if &min = '' then iso&min = '-';
                              else iso&min = &min;
                 &isovar = strip( &isovar ) || ':' || strip( iso&min );
               %end;

             %if &sec ^= %then
               %do;
                 if &sec = '' then iso&sec = '-';
                              else iso&sec = &sec;
                 &isovar = strip( &isovar ) || ':' || strip( iso&sec );
               %end;

           %end;

    %end;

%mend ISO8601_311;
```

DEFINE.XML

FDA guidance for electronic submissions specify 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 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. The schema for the SDTM Define.XML is based on an extension of the CDSIC ODM. Details on Define.XML are published in the Case Report Tabulation Data Definition Specification document available at the CDISC website listed at the beginning of this paper.

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. 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