

Using SAS® ETL Studio to Convert Clinical Trials Data to the CDISC SDTM

Barry R. Cohen, Octagon Research Solutions, Wayne, PA

ABSTRACT

A new industry standard for clinical and preclinical trials data (the CDISC SDTM model) has been developed and is quickly being adopted by the pharmaceutical industry. Thus, for future Marketing Applications to the FDA, companies will need to convert their clinical and preclinical trials data from various legacy and current-standards (and non-standards) to the new industry standard SDTM. This will be a major undertaking for any organization, and a data warehouse tool can be of significant help in this process. ETL Studio is the new SAS product designed to support a warehouse building process. Octagon Research Solutions uses ETL Studio as the key software application in the complex process of converting its clients' clinical trials data into submission-ready data based upon the CDISC SDTM. This paper will discuss various aspects of Octagon's SDTM conversion process that uses SAS ETL Studio.

INTRODUCTION

The Clinical Data Interchange Standards Consortium (CDISC) has developed a standard for clinical and preclinical trials data. The standard is called the Study Data Tabulation Model (SDTM). The FDA has stated in eCTD guidance that they now would like to receive clinical and preclinical trials data from sponsors in the SDTM. And the industry expectation is that within a few years the FDA will require by regulation that sponsors submit their data in the SDTM. It is thus no surprise that the industry is quickly adopting the CDISC SDTM. Many sponsors are already converting clinical trials data from active Marketing Application projects to the SDTM, many others are assessing their SDTM-readiness and currently planning their first SDTM conversions, and still others are exploring ways to integrate the SDTM standard into their full clinical data life cycle.

Throughout these activities, sponsors are beginning to understand that an SDTM conversion project is a major undertaking. SDTM conversion projects are major undertakings, with significant work required in two top-level areas. The first area regards project/process setup, which includes (among other things): process design, process documentation, process validation, staff resource and deployment, and staff training. The second area regards the ongoing conversion work once the project/process is established. The conversion work includes mapping specifications development and program development and validation per study, for the many datasets per study, the several/many studies per project, and the stream of projects over time.

And some sponsors have made or are considering a decision to convert all their legacy studies to the SDTM structure, too, in order to build a single-standard clinical data warehouse. They will do this both as an effective way to archive all their clinical data and as a way to support data mining of this data for various purposes. In these cases, the amount of data conversion work involved will be significantly larger than that needed to support active Marketing Application projects.

Many sponsors will look to data warehousing technology to help them in the SDTM conversion process. They will obviously look to this technology if they will be building a clinical data warehouse. But data warehousing technology can also help even if they are only looking for an efficient way to do an SDTM conversion for the data of individual Marketing Application projects without (yet) warehousing the data across the multiple projects within the company. Data warehouse technology is designed to extract data from a source (e.g., existing clinical trials data in legacy format or in current-standard format), transform it in a variety of ways (e.g., as needed to become SDTM-compliant), and load it to a final target (e.g., to individual SDTM domain SAS datasets for submission to the FDA). And data warehouse technology can accomplish this process with far less help needed from SAS programmers than would be needed if all the programming were developed more traditionally without a warehousing tool.

Octagon Research Solutions, Inc. is a consulting organization with deep expertise in the CDISC standards including SDTM and the Operational Data Model (ODM). We are actively converting clinical trials data from several sponsors to SDTM, and we are using the data warehouse technology from SAS, specifically SAS ETL Studio, to support our SDTM conversion process. This paper begins with a background discussion about SDTM and how and where it can and will be integrated into the clinical data life cycle across the industry. This discussion will help set the context within which you will be using SAS ETL Studio (or using another method) to convert clinical data to SDTM. The paper will then discuss our SDTM conversion process and the role of SAS ETL Studio in that process. A short list of topics covered includes:

- SDTM – a brief state of the industry
- Where and when the SDTM standard will be integrated into the stages of the clinical data life cycle
- Why late-stage conversion to SDTM will occur for some time to come

- Basic issues of a data conversion process
- Overview of Octagon's SDTM conversion process using SAS ETL Studio
- Process design and implementation Issues we encountered when doing SDTM conversion using ETL Studio
- Conclusions from our experience doing SDTM conversion using ETL Studio

SDTM STANDARD – STATE OF THE INDUSTRY

MOTIVATION FOR ADOPTION – REGULATION AND EFFICIENCY

The SDTM is rapidly being adopted as the standard for clinical data in the industry. The immediate motivation is a regulatory one. Today, the FDA has stated in eCTD guidance that it wants to receive the clinical and preclinical data components of Marketing Applications in SDTM. The FDA has committed to using SDTM and has prepared a set of in-house tools to process SDTM data received from sponsors. Noteworthy are the WebSDM tool that they use to validate and load SDTM data to their internal repository (JANUS warehouse) for analysis and reporting, the JANUS data warehouse itself that is based upon the SDTM standard, and the Patient Profile Viewer (PPV) that produces patient profiles. There is additional incentive for a sponsor to submit its data in SDTM today because if they do so, the FDA will no longer require the sponsor to provide the data listings or patient profiles. Instead, the FDA will use its in-house tools to produce these listings. Today, SDTM data is the FDA's desire, expressed in eCTD draft guidance, and not its requirement. But the industry generally feels that it is likely the SDTM will become a requirement within a few years. So today companies are moving toward building SDTM capabilities within their organizations. Some things affecting the speed of adoption include when their next Marketing Application with a data component is being submitted, how well equipped they are in-house to plan an SDTM-implementation project, and how soon they think that SDTM will become a regulated requirement.

The other, longer-term motivation for organizations to adopt SDTM as their clinical data standard stems from processing efficiencies. Much efficiency is possible, and I mention some examples just below. In essence, the industry is recognizing that their proprietary data standards do not have particular advantage vis-à-vis their competitors, and they can gain more from working to the same standard than from developing and maintaining their own standards. Some examples:

- Easier data interchange between sponsors and contract research organizations (CROs)
- Easier data interchange between sponsors and central laboratories
- Easier data interchange during collaboration with research partners
- Avoid building and/or maintaining a data standard when using a standard built and maintained by the industry
- Easier harmonization of data and processes when acquiring or merging with other sponsors
- Ability to archive data across all studies over time in a single-standard, and the consequent ability to build a data warehouse based on that standard, which can then be used for various data mining related research activities.

THREE COMPONENTS OF SDTM

I do not describe the CDISC SDTM model in this paper. I refer you to the CDISC website (www.cdisc.org) for ample documentation about the model. However, I do want to discuss one aspect of SDTM that will help you to understand both what is involved in converting clinical data to SDTM and where/when SDTM will be integrated into the stages of the clinical data life cycle within an organization. The SDTM model is comprised of three components: nomenclature, content, and structure. Most people tend to focus first on the SDTM structure. This is understandable because it is the most obvious component and the CDISC SDTM documentation focuses primarily on structure. Structure concerns what domain datasets exist, what set of variables exist in each domain, and for these variables, what are the types, lengths, and positions in the dataset, etc. It also concerns the shape of the datasets (i.e., "long and narrow" or "short and wide").

Nomenclature concerns the standardization of names and labels used to identify items (variables) in SDTM datasets. For example, the variable named SEX is used in the demography domain, not the variable GENDER. SEX is the standard. Content concerns the standardization of the values of certain variables in the SDTM model. For example, for the variable SEX, the standard values are "M", "F", "U", not "Male", "Female", "Unknown", and not "1", "2", "9". As another example, in a future release of the SDTM model, the values for TEST and TESCD (test name and test code in various Findings domains) will be standard as well. Structure and nomenclature are stable today in the SDTM model. A limited amount of standard content, called "Controlled Terminology" exists in the SDTM model today, and more is being developed now for future release.

It is interesting, and important, to note that clinical data can be compliant with SDTM regarding nomenclature and content without being compliant regarding structure. That is, clinical data can be stored in a different structure than the one defined by SDTM, and still use SDTM-compliant nomenclature and terminology. This fact is important to understanding how, where, and when SDTM will be implemented throughout the stages of the clinical data life cycle in an organization.

CLINICAL DATA LIFE CYCLE

The clinical data business process can be described as having the following stages in its life cycle:

- **Collection** – The data is collected in this stage, either on paper-based Case Report Forms (CRF) or using e-CRF's in an Electronic Data Capture (EDC) system.
- **Processing** – The data is maintained and processed in this stage. All manners of processing needed to validate the data occur here. The collected data is frozen here once it is validated, and it is then ready for analysis. Data is stored here on an individual study basis, as opposed to trans-study. Database systems are often used in this stage, generically called Clinical Data Management Systems (CDMS). A salient example is the Oracle Clinical product. If an organization is operating according to some internal data standard, then global libraries are built in the CDMS that hold the standard metadata for items used in the various studies.
- **Storage** – This is an emerging and changing stage in the life cycle. Many companies tend to extract the frozen data from the CDMS and store it in SAS datasets for the Analysis and Reporting stage. Historically, the storage repository has simply been a file system where the various datasets of individual studies are kept together, with limited consideration for trans-study storage for trans-study analysis and reporting. The obvious exception to this is for the pooling of data across the studies of a Marketing Application, on an ad hoc basis, for Integrated Safety Summaries and Integrated Efficacy Summaries (ISS/ISE). But the Storage stage is changing today as companies begin to think about and develop clinical data warehouses to archive their data in a trans-study and trans-project manner. They will then use the warehouse as the data source for the Analysis and Reporting stage for individual studies, as the data source for the pooling of data for the ISS/ISE of submission projects, and for true trans-study data mining for a variety of purposes including clinical research, safety signaling, and others. (And I note that SDTM is now being considered in many organizations as the data standard for the warehouse).
- **Analysis and Reporting** – In this stage, the analysis and report programs are developed and executed to analyze and report on the data, per study. This tends to be a very SAS-centric stage in the life cycle. This work includes both the individual study analysis and reporting and the ISS/ISE analysis and reporting.
- **Compilation and Submission** – In this stage, the analysis and report results are integrated into the Marketing Application being prepared for the FDA. This document is referred to as the Common Technical Document (CTD) or increasingly today the eCTD as more and more companies submit electronic Marketing Applications. The data component of the Marketing Application is also assembled and submitted in this stage. If an organization is submitting their data in SDTM-compliant datasets, this is the point at which those datasets must be created. It is possible (and may prove to be desirable) to create the SDTM datasets at an earlier stage and use them for the activities at that earlier stage, (e.g., the Analysis and Reporting stage). But this is the last point at which the SDTM datasets can be created. I will refer to this as "late-stage" conversion to SDTM.
- **Review** – This stage is conducted at the FDA, as the agency reviews the submitted data and documents of the Marketing Application. As mentioned above, the FDA has now developed a series of in-house tools, based upon the SDTM standard, to facilitate the data and analysis review.

For the purposes of the discussion in this paper, I will consider the Compilation and Submission stage as the last stage because my focus is on the activities conducted by the sponsor.

SDTM IN THE CLINICAL DATA LIFE CYCLE

SDTM can be integrated into any stage of the clinical data life cycle. And yet, for many companies, in the near-term, SDTM is likely to only be integrated into the last stage (Compilation and Submission). This is so because companies already have established clinical data business practices that are based on an internal data standard that has its own nomenclature, content, and structure. Of course, there are varying amounts of adherence to an internal standard as the data moves across stages (and departments) of an organization. But even without strict compliance to one standard, the practices at each stage are based upon particular data standards or non-standards that are different from SDTM. And many automated tools have been developed based upon the existing standards. Thus, a substantial change in business systems and practices will be required for an organization to integrate SDTM into any stage of the life cycle, but it will be easiest in the last stage. In essence, in the last stage, all the processing is done and the data is being compiled for submission. The data can thus be converted to SDTM at this stage without requiring a change to any existing systems and practices.

Given the substantial time needed to integrate SDTM into upstream stages of the life cycle, and the desire to submit SDTM-compliant data to the FDA in the near-term, either to meet the current FDA eCTD guidance or to meet the expected FDA regulation, sponsors will have no choice but to do a late-stage conversion of their data to SDTM. This is the first context within which most organizations will need to determine a method to convert their data to SDTM, and the first context within which they might use ETL Studio (or another method) to effect this conversion. And for this conversion, the source data will either be the SAS datasets generated and used during the Analysis and Reporting stage, or the database tables from the CDMS where the frozen, collected data is stored.

There are two important incentives to move the integration of the SDTM standard upstream into earlier stages. The first is that the processing efficiencies that come from operating according to the industry's data standard will only accrue to an organization if they integrate SDTM into the upstream stages of the life cycle. The sample list of efficiency areas that I provided above in the section titled "Motivation For Adoption – Regulation And Efficiency" all refer to work being done in earlier stages of the life cycle, and often in all earlier stages. Thus, an organization will have to implement SDTM into these upstream stages to gain the efficiencies. The second is that an early-stage integration of SDTM (at least nomenclature and content) can avoid or minimize the extra step of late-stage conversion. With upstream integration of SDTM (at least nomenclature and

content) the data is already partially SDTM compliant and only the structure will have to be changed in the late stage. Or, if the data is converted to SDTM structure earlier, too, then the organization can avoid doing the conversion at a point very late in time when time pressures on the project are greatest.

But even if SDTM is integrated upstream, and there is no conversion to SDTM per se but rather integrated use of SDTM throughout, there might still be reasons to use an ETL tool like SAS ETL Studio in the business process. I list a few examples here:

- Extraction of data from the CDMS, which is compliant with SDTM nomenclature and content but not structure, and conversion to SDTM structure in SDTM domain datasets.
- Extraction of data from the CDMS, which is compliant with SDTM nomenclature and content, and loading to a clinical data warehouse.
- Extraction of data from a clinical data warehouse, which is compliant with SDTM nomenclature and content, for building analysis datasets for the Analysis and Reporting stage, or for conversion to SDTM structure in SDTM domain datasets.

It is important to note that even if the need for an ETL tool is limited to late-stage conversion to SDTM, most sponsors are likely to encounter this need in measurable fashion for several years to come. One reason for this was mentioned above --- that it will take time to implement SDTM as the standard in the upstream stages, and substantial amounts of data will have to be converted to SDTM in the meantime for submission to the FDA. A second reason is that submission projects typically take years to complete, and the study data within them is not often switched from one data standard to another mid-stream. Thus, companies are likely to have a long period of "parallel standards" where the studies of new projects are completed using the SDTM standard, and the studies of existing projects are completed using the earlier internal standard. And these latter studies will need late-stage conversion to SDTM.

DATA CONVERSION – GENERAL PICTURE

I have now described the business context for conversion of clinical data to SDTM, and I turn attention to the data conversion process, which in Octagon's case is centered on the SAS ETL Studio. For the remainder of the paper, I will be thinking primarily about late-stage conversions to SDTM, where the non-SDTM datasets used during the Analysis and Reporting stage are now being converted to SDTM. I believe this will be the likely scenario for many/most sponsors in the few or several years just ahead. I will first discuss some general issues of data conversion and I will then describe our conversion process using the SAS ETL Studio product.

GENERAL ISSUES OF DATA CONVERSION

Most generally, a data conversion effort involves extracting data from a source, transforming it in some fashion, and loading the result to a target. The acronym ETL stands for just this process: "Extract, Transform, Load". In the clinical data world, the source data for a data conversion project is likely to be either SAS datasets or database tables, and the target data is likely to be one of the same. Using SAS datasets as the example for source and target, the conversion process will encounter one or more of the following source-to-target situations, which will very much influence the programs that need to be developed to effect the conversion:

- One-to-one – All the data from one and only one source dataset is going to one and only one target dataset
- One-to-many – The data from one given source dataset is going to multiple target datasets
- Many-to-one – The data from many source datasets is used to build one target dataset

The one-to-one situation is, not surprisingly, the easiest to program. The program to build any target dataset this way would only have to open and read data from one source dataset, make the necessary transformations, and write the results to one target dataset. You would not have to address the more complicated programming issues that arise if you need to either read from your source datasets multiple times before you have written all their data to your target data sets, or write to your target datasets multiple times before you have finished reading the data from all of your source datasets.

Transformations can be seen in three groups, as follows:

- Changes to nomenclature – This largely involves changes to variable names and labels, to what things are called if you will.
- Changes to content – This largely concerns changes to the values of the data to meet an accepted standard. For example, a source variable with the values 1 and 2, (which correspond to Male and Female), is changed to have the standard values M and F in the target variable. Or a set of non-standard names for ECG tests in the source dataset is changed to a set of standard test names in the target dataset. Or a date variable that is numeric in the source dataset is changed to a character representation in the target dataset. Or the units for the results of a particular test in the source dataset are changed to standard units in the target dataset.
- Changes to structure – This basically concerns the set of datasets used, the particular variables found in the particular datasets and their types, lengths, and positions, and the organization of data by rows and columns in the datasets. The structural issue of mapping source datasets and variables to target datasets and variables was alluded to above in the one-to-many and many-to-one situations. Another structural change can occur if the organization of the data is changing

from “short and wide” datasets to “long and narrow” datasets. For example, a source dataset structure where test results for multiple tests done at the same visit are on the same record (i.e., short and wide) is changed to a target structure where each test at a given visit has its own record and that record holds both the test name and the result (long and narrow). Another structural change can occur if there are items needed in a target dataset than do not directly exist in any of the source datasets.

Another conversion issue concerns the approach chosen to develop the programs that do the actual conversion. There are situations where the conversion is needed only once or a limited number of times, and for a small number of datasets, and the programming will thus be done on an ad hoc basis. But when there are large amounts of data to convert, and the program development effort will consequently be large, the conversion will likely involve some kind of software application. Such applications will have these two key features (among others): (1) The conversion programs for a given set of data are auto-generated by the application, which reads and interprets conversion specifications; (2) Specifications for tasks (such as transformations) that re-occur across studies being converted are developed once are reused many times. These conversion software applications can either be developed in-house or acquired in the marketplace. The commercially available products are often referred to as ETL applications, or Data Warehouse applications, or Data Integration applications. SAS ETL Studio is such an application.

Yet another conversion issue concerns the exact set of steps are needed in the process of conversion and what staff skills are needed for each step. The exact steps may vary based upon whether you will develop ad hoc conversion programs or use a conversion application. And if you use an application, the exact steps may vary by the design and features of the application. For example:

- For ad hoc program development: Do you need a separate mapping specifications step, where someone specifies the details of how the source data will map into the target datasets? Is this step done by the programmer and if not, then by what type of staff?
- For conversion applications: Do you still need a separate mapping specifications step when using a conversion application as opposed to ad hoc program development? With an ETL tool, is the mapping specifications step done inside the ETL tool? What staff skills are needed for working with the ETL application? Are they programmer skills or other skills? Are they different from the mapping specification skills? Can a person develop the conversion jobs or processes inside the ETL tool without writing any program code? Is this the same if the conversion application is built in-house or acquired commercially?

One more issue concerns system and process validation. In a regulated industry, such as the pharmaceutical clinical trials industry, where the programs and processes must be validated, there are implications for ad hoc development of conversion programs versus using a conversion application. Specifically, all ad hoc programs must be validated. The more programs you write this way, the more validation you must do. In contrast, if you use a conversion application, the application must be validated by you if you build it or by the third party vendor if you purchase it. Then you must validate the process you conduct using the application. But you do not need to validate the programs generated by the application.

OCTAGON'S SDTM CONVERSION PROCESS

I now turn attention to Octagon Research's data conversion process for SDTM, which uses SAS ETL Studio as the conversion application. First I describe our process overall, discussing how our process works in regard to the various general conversion issues raised in the section above. Then I discuss some noteworthy issues of using SAS ETL Studio in our SDTM conversion process.

OVERVIEW

There are four steps to our SDTM conversion process, as follows. There is a fifth step, validation of our work, but we see validation as integrated into each of the other four steps. The steps are identified here, and then the first two, which are the ones needed to produce the SDTM datasets, are discussed in separate subsections below.

- Development of mapping specifications
- Development of conversion jobs in ETL Studio
- Development of the Define document
- Development of the Annotated CRF

DEVELOPMENT OF MAPPING SPECIFICATIONS

This is the first step in our conversion process. A “mapping specialist” conducts the step. The mapping specialist examines each item in each source dataset of the study and determines if the item should be migrated to the target SDTM datasets, and to which SDTM domain dataset and variable the item will migrate. Some items in the source datasets are not collected clinical data and will not migrate to the target SDTM datasets. These items tend to be operational items added to the source datasets by the sponsor. During this step the mapping specialist also identifies additional data that is needed in particular SDTM domain datasets being created but is not coming directly from the source datasets. There is a fair amount of this situation in

SDTM conversion, and it is one reason that the mapping specifications step is involved and is more than just drawing lines connecting the variables in the source and target datasets.

The mapping specialist needs the following skills to accomplish this work: (1) strong clinical data knowledge; (2) strong SDTM knowledge; (3) knowledge of the SAS Viewer, SAS Display Manager, and the use of SAS formats, in order to thoroughly examine the contents and structure of source SAS datasets; (4) knowledge of Excel to enter, manage, filter, and sort on data (since the specifications are developed in Excel).

The mapping specialist uses the following resources in his/her work: (1) the source datasets; (2) the source-annotated CRF; (3) the protocol document. The mapping specialist starts by reviewing the protocol and source datasets to gain an overall understanding of the data being converted to SDTM. The specialist then runs our custom SAS program that reads the contents of all the SAS datasets in the source library and generates an Excel spreadsheet with one row for each source dataset-variable combination in the source data and populates this row with a variety of metadata about the item in that row. We call this a source-centric spreadsheet in that it is oriented to what is in the source data, not the target data. There are additional columns on each row where the mapping special then indicates to which target SDTM dataset-variable combination the source variable will migrate, and any transformation that is necessary. Finally, the specialist adds rows to the mapping specs spreadsheet for items that are needed in the various target SDTM datasets that are not directly in the collected data in the source datasets. An example of this would be the CAT (Category) and SCAT (Subcategory) variables of the various SDTM domains. This data is not typically collected and stored in the source datasets, but it is often printed somewhere on the CRF page. The specialist finds this information and adds it to the mapping specifications. Another example of this would be test names that must be present in the --TEST variable in SDTM Findings domains. These names are typically printed on the CRF page but not necessarily captured as data in the source datasets.

The mapping specialist produces the mapping specs spreadsheet, and he/she also produces an SDTM-annotated CRF. This CRF will be used as a resource in subsequent steps.

A good number of issues arise during this step that must be resolved before it is completed. Some pertain to clarifying what is in the source data. Others pertain to SDTM-related decisions the sponsor must make. Some arise because there are sometimes choices as to where particular source data will reside in SDTM. Others arise because someone other than the mapping specialist must provide certain sponsor-defined information. An example of this might be the actual values for --CAT and --SCAT variables in a domain. Another example might be the rule the sponsor wants to follow for computing Reference Start and End Day in the SDTM data. As stated, the number of issues arising can be measurable and their resolution is critical to a correct and complete conversion to SDTM. Thus, at Octagon, we use our own in-house developed (and commercially available) process management software, ViewPoint[®], to manage this process.

There is a Quality Control (QC) review when the mapping specs spreadsheet is completed. Basically, a second mapping specialist reviews the work in the spreadsheet and reports errors back to the first specialist for correction. In matters of judgment, where the two specialists do not agree, the issue is brought to Octagon's resident SDTM experts for resolution.

DEVELOPMENT OF CONVERSION JOBS IN ETL STUDIO

The ETL Studio environment is one where the user (whom we refer to as the "ETL developer" or just "developer") defines a job that loads one or more source datasets, maps source variables to variables in target datasets, transforms the source variables where needed, and finally loads the finished variables to the target datasets. The developer executes the completed job in the ETL Studio environment. When the job executes, ETL Studio generates a SAS program according to the definition of the job, and submits that program to the SAS environment for execution, with the results (i.e., Log) returned to the ETL Studio environment and the resultant dataset(s) placed in a specified SAS library.

Note that this is not traditional SAS program development, nor is a SAS programmer doing it. The ETL developer is not a SAS programmer and he/she is not writing SAS code. Rather, the developer is using the interface of the ETL Studio environment (menus, selection lists, graphic schematics, etc.) to define a job, and ETL Studio then writes the program from this job definition.

The developer needs the following skills in this role: (1) clinical data knowledge (but not necessarily as much as the mapping specialist); (2) SDTM knowledge (but not necessarily as much as the mapping specialist); (3) knowledge of the SAS Viewer, SAS Display Manager, and the use of SAS formats in order to thoroughly examine the contents and structure of source SAS datasets; (4) some kind of data processing/management/programming experience, (but not as much as a fully experienced programmer, and the experience does not have to be with SAS data and/or SAS programming). The developer uses the following resources during this work: (1) source datasets; (2) source-annotated CRF; (3) mapping specs spreadsheet; (4) SDTM-annotated CRF. And the product of this step is the SDTM datasets for the study.

There is one exception to the statement that the developer does not write any SAS program code. It concerns defining the transformations in ETL Studio. An ETL Studio job is comprised of processes. Each process is one step of the full job. Some examples of processes are: splitting one source dataset into two target datasets, joining two source datasets into one target dataset, sorting a source dataset and saving a new target dataset as output, and transposing a dataset from short-wide to long-narrow structure. Each process has a Mapping Properties window where the variables for both the source and target

datasets are visible and where the mapping of source variables to target variables is expressed. This window also includes an item, per target variable, named "Expression", where a variable-level transformation is defined. Simple transformations can be defined by writing SAS code directly into this Expression area. Examples would be: a SAS statement using the PUT function to convert a numeric variable to character representation, a SAS statement using a SAS date function to convert a date variable to ISO8601 representation. If more involved SAS code is needed for the transformation, ETL Studio provides an Expression Builder that is similar to the interactive query builders that are available in some SAS applications and other applications. Here, the developer can point and click within the interface to build transformation logic expressed in SAS code. An example would be a set of SAS statements that provide decoded values for a formatted variable of many values.

There is also one role for a SAS programmer in the ETL Studio environment when using it for SDTM conversion. It concerns the fact that there may be a certain kind of processing that you cannot define inside ETL Studio, or cannot define easily. If so, a SAS programmer will need to program this process as a SAS program, outside the ETL Studio environment. The program will be placed in the ETL Studio process library, and then engaged by the ETL developer as he/she defines ETL Studio jobs. I note, though, that the amount of program code that we develop this way is quite small compared to the amount of code we develop by using the processes already provided by SAS inside the ETL Studio process library.

Notice that we have an initial mapping specifications step, done by a mapping specialist, and then a job definition step inside ETL Studio, done by the ETL developer, that is somewhat similar and has redundant features. The mapping specifications step is focused on specifying the conversion to SDTM completely and correctly. In essence, this is the "what" step or the "requirements" step in program development. The Excel environment is stronger for doing this particular work than is the ETL Studio environment. The Excel environment, and our approach to using it, allows us to easily see all the source items and all the target items in one sheet where we can filter and sort quickly and easily on source datasets and variables and on target datasets and variables. It also allows us to see each source-to-target mapping and its transformation on the same row or immediate next row as opposed to on rows that are far apart (and thus not on the video screen at the same time). The mapping specialist needs these facilities and flexibilities to efficiently and effectively specify the conversion.

The ETL Studio environment certainly has facilities for expressing a mapping from source dataset and variables to target dataset and variables, and for expressing transformations. But the implementation of these features does not provide quite the same facilities and flexibilities as those in Excel. Thus, for now, we are developing the mapping specification on a spreadsheet, and the ETL Studio developer then uses this spreadsheet as a guide as he/she does the ETL Studio development work. There is a certain amount of repeated work in this. But it is not nearly a complete repeat of work because the ETL Developer is also doing other things that are concerned with defining and executing ETL Studio jobs. However, we are presently exploring ways to either load the work done on the Excel mapping specifications sheet into ETL Studio so the ETL Studio developer does not repeat any work, or to modify the ETL Studio interface so it provides the same facilities and flexibilities our mapping specialist uses in Excel. And if we do solve this by modifying the ETL Studio interface, then we will consider if the mapping specialist work and ETL developer work can eventually be integrated into one job role. In essence, we feel that we have the right tool and are on the right track, and as time allows given our heavy operational workload doing SDTM conversions for clients right now, we will finish addressing this issue.

Our process also includes a step where we check our work done in ETL Studio. We are not validating the ETL Studio product because it has been validated by SAS. Rather, we are checking that we have produced correct SDTM datasets by following our validated work procedures correctly. Some key points:

During our ETL Studio development work:

- We have SOP's and Guidelines that we developed for the ETL Studio developers' work, and we train all our developers in them as well as in general use of the ETL Studio application. We have a validated process for our developers to follow.
- We use valid SDTM template datasets as the definition for the target datasets we produce in SDTM. These templates are available by downloading the Excel spreadsheet from the CDISC members-only area of their website that has the structure and metadata information for the SDTM domain datasets. This information can be used to build empty SAS datasets for the target SDTM domains. (Note: If you license ETL Studio, it is worth checking with SAS to see if these template datasets are now available to you from SAS).

At the end of the ETL Studio development work:

- SDTM-compliance checking – Here we check the structure, nomenclature, and standard content (i.e., controlled terminology) of the SDTM datasets. We do these checks programmatically.
- Validation of target data against source data – Here we check that the right source data wound up in the right place in the right target dataset. In essence, an SDTM domain dataset built via conversion from source data might be SDTM-compliant and still have the wrong data values in it. You need to check the target data against the source data to insure complete correctness. This type of checking is harder to do programmatically, especially because the source data changes from study to study so the checking programs must be more general/flexible. However, we are progressing on a library of flexible programs to do this type of checking.

OTHER SDTM CONVERSION ISSUES IN ETL STUDIO

Many process design and implementation issues arose for us as we began to use ETL Studio for SDTM conversions. I present a few noteworthy ones in this section. These are issues that I believe most or all organizations using ETL Studio to convert legacy clinical data to SDTM will encounter.

Ordering the Processing of Source and Target Datasets – You may have 20-30 source datasets in a study being converted to SDTM, and your mapping specifications may have determined that you are producing 20-25 SDTM target datasets. How do you determine how many ETL Studio jobs you will need, what source datasets will be processed in what jobs, and what target datasets will be built in what jobs? The greatest challenge for this issue is that in an SDTM conversion project you will not much one-to-one correspondence of source datasets to target datasets. The reality of the situation is that data from a given source dataset is going to multiple target datasets, and data in a given target dataset is often coming from multiple source datasets.

We designed the following approach that is working well for us. We view the processing as occurring in two waves. The first wave is source-centric, which means we extract the data from each source dataset, transform it where needed, and load it in a series of interim target datasets, one for each SDTM domain to which this data is going. Usually we use one ETL Studio job for each source dataset being handled, although there are instances when we extract from multiple source datasets in the same job. We store the interim datasets in a work library. The second wave is target-centric. Here we have an ETL Studio job for each SDTM target dataset we are creating. We extract data from all the interim datasets holding data for the given target, do additional transformations where needed, and load all the data to the target dataset.

This approach works well in response to the complexities of the one-to-many and many-to-one mappings that occur during SDTM conversion. And it has added value in response to additional complexities that arise concerning assignment of sequence numbers to records in various SDTM domains. This is discussed in the next issue just below.

Assignment of Sequence Numbers to Records in SDTM Domains - All SDTM domains (except Demography) have record sequence numbers, given to ensure uniqueness within a domain for a subject. These sequence numbers can then be used to relate records. A typical case of relating records is the one of relating records in a COMMENTS or SUPPQUAL dataset with their associated records in the parent domain. For example, you might have an AE domain and a SUPPAE dataset and need to relate SUPPAE records back to their associated records in the parent AE domain. Often, this is done using the sequence number variable AESEQ. AESEQ in the AE domain is populated with values, and then the AESEQ values are stored on the various records in SUPPAE (using the IDVAR and IDVARVAL variables) to make the link back to the correct parent record in the parent AE domain.

Staying with the AE example, a challenge arises during the conversion of the source AE data to the SDTM AE domain and SUPPAE dataset. It regards putting the sequence numbers onto the various records in SUPPAE. The problem is that when you disassemble the source AE dataset into two interim datasets during the first wave of processing (as per the section just above “Order of Processing...”), you do not yet have the values of AESEQ available in the AE domain to be able to write them onto the SUPPAE records. In fact, the AESEQ values will not be available until the second wave of processing when you assemble all interim datasets that pertain to the AE domain into the final target AE domain.

However, by processing in two waves, and saving interim datasets between waves, it is reasonably easy to build a temporary sequence number in the first wave that maintains the relationship between records in the interim datasets. Then, in the second wave, when the final sequence numbers in the final AE domain have been generated, we can re-link the records between the final AE domain and interim SUPPAE dataset using the temporary sequence number, add the final sequence numbers to records in SUPPAE, and drop all the temporary sequence numbers.

Reuse of ETL Studio Jobs and Processes - There is obvious value to being able to reuse work you developed on one study in another study. In ETL Studio, this work might be an entire job you defined, or the definition of individual processes within a job. You can share whole jobs or individual processes in ETL Studio by copying and pasting them from one study repository to another. Or, you can share them with many studies at once by copying them into a global repository. Our experience is that we have successfully shared both jobs and processes. We have not done much job sharing because the exact work we need to do changes from study to study and from client project to client project. So it is not often that a whole job for one study of one client will work for another study or another client. Regarding sharing of processes, most sharing has been for processes that we have written as an external SAS program and loaded into the process library. If the process to share is one that was provided by ETL Studio in its native process library and we just set various parameters when we used it, then it is usually just as easy for the developer to reselect the native process for the next job and reset the needed parameters.

CONCLUSION

The Clinical Data Interchange Standards Consortium (CDISC) has developed a standard for clinical and preclinical trials data. The standard is called the Study Data Tabulation Model (SDTM). The FDA has stated in eCTD guidance that they now would like to receive clinical and preclinical trials data from sponsors in the SDTM. And the industry expectation is that within a few years the FDA will require by regulation that sponsors submit their data in the SDTM. The industry is quickly adopting the CDISC SDTM. Many sponsors are already converting clinical trials data from active Marketing Application projects to the SDTM, many others are assessing their SDTM-readiness and currently planning their first SDTM conversions, and still others

are exploring ways to integrate the SDTM standard into their full clinical data life cycle. The initial motivation for this is regulatory but there are many opportunities for processing efficiency, particularly surrounding data interchange among the various organizations involved in the clinical data life cycle.

The SDTM is comprised of nomenclature, content, and structure. This view of the SDTM is important to understanding where and when SDTM can and will be integrated into the various stages of the clinical data life cycle. The stages are: Collection, Processing, Storage, Analysis and Reporting, Compilation and Submission. For many sponsors, for some time to come, SDTM will only be integrated into the last stage. This is because much time is needed to integrate a new data standard throughout the whole life cycle. Many sponsors will continue to handle their study data according to their current standard (or non-standard) and do this "late-stage conversion" while they work in parallel on integrating SDTM as their new standard throughout their clinical data life cycle. The primary reasons for integrating SDTM fully into the life cycle are (1) once your whole process is based upon the SDTM data standard, you do not have to convert your data to SDTM at the end of the process, and (2) you must use the SDTM data standard in all stages of the life cycle to capitalize on the significant opportunities for efficiency.

SDTM conversion projects are major undertakings, with significant work required in two top-level areas. The first area regards project/process setup, which includes (among other things): process design, process validation, process documentation, staff resourcing and deployment, and staff training. The second area regards the ongoing conversion work once the project/process is established. The conversion work includes mapping specifications development and program development and validation per study, for the many datasets per study, several/many studies per project, and stream of projects over time.

Data warehouse technology, also known as data integration technology or ETL technology, is designed to support the data conversion process. SAS ETL Studio is a member of this technology class.

Octagon Research Solutions is a consulting organization with deep domain knowledge regarding CDISC standards and their use in the industry. We have developed an SDTM conversion process that we are actively using today to convert our clients' data to SDTM. Our process is centered on the SAS ETL Studio product. With our domain knowledge and ETL Studio, we have designed a process that has addressed the key challenges of conversion to SDTM, including:

- Data transformations due to changes in nomenclature, content, and structure
- The one-to-many and many-to-one relationships between source datasets and SDTM domains
- The assignment of subject-specific record sequence numbers and the specification of relationships between records in parent SDTM domains and related SDTM datasets. And doing this when the one-to-many and many-to-one relationships among source and target datasets forces an order of processing that does not lend itself to easily generating the sequence numbers.
- Automated quality control checking.
- Reuse of work across studies and projects

We have been successful in our project objectives. Some particular strengths of ETL Studio that have contributed to this success are:

- ETL Studio generates the conversion programs from our mapping specifications. Our staff generates mapping specifications and expresses them inside the ETL Studio environment. They do not write programs but rather write specifications. This means two important things for us: (1) We do not validate an ongoing series of newly developed conversion programs, and (2) We have an efficient use of staff resources because SAS programmers do not do our ongoing conversion work.
- ETL Studio allows reuse of processes and jobs across studies.
- ETL Studio extracts directly from SAS datasets and loads directly to SAS datasets without an extra conversion step on our part on the front end and back end. This means we avoid a corresponding extra validation step on the front end and back end.
- ETL Studio is extensible in that we can write transformation utilities as external SAS programs, where needed, and load them into the ETL Studio environment for use.

Our conversion process, and the role of ETL Studio within it, is young and evolving. These are the primary focus right now of the next steps in this evolution:

- We are considering extending the ETL Studio interface to better support the particular mapping specifications development process we follow for SDTM conversion.
- We are working on a process for loading the SDTM target datasets we produce to a clinical data warehouse in addition to our present loading them to study-specific and project-specific file systems.
- In conjunction with data warehouse building, we are exploring ways to leverage the data that ETL Studio puts in the SAS metadata server about each study during the conversion process. We are particularly looking at use of SAS reporting tools from the Business Intelligence Platform.

REFERENCES

SAS Institute Inc. 2004. "SAS® 9.1.3 ETL Studio: User's Guide". Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2004. "Using SAS® ETL Studio to Integrate Your Data Course Notes". Cary, NC: SAS Institute Inc.

Susan Kenny and Michael Litzinger: "Strategies for Implementing SDTM and ADaM standards", Paper FC03 at PharmaSUG 2005, <http://www.pharmasug.org/2005/FC03.pdf>

ACKNOWLEDGEMENTS

I would like to thank David Evans, Chief Information Officer, Octagon Research Solutions, for his insights into the current state of adoption of SDTM in the industry and the likely paths and timeframes of future adoption. I would also like to thank Tom Guintier, Vice President of Clinical Data Strategies, Octagon Research Solutions, for his review of this paper and his insights into the evolving role that ETL Studio can play in the SDTM conversion process.

CONTACT INFORMATION

Barry R. Cohen
Director, Clinical Data Strategies
Octagon Research Solutions, Inc.
Wayne, PA 19087
610 535 6500 x635
bcohen@octagonresearch.com
www.octagonresearch.com

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.