

# The Use of CDISC Standards in SAS from Data Capture to Reporting

Andrew Fagan, SAS Institute, Inc., Cary, NC, US  
John Leveille, d-Wise Technologies, Inc., Raleigh, NC, US

## ABSTRACT

This paper looks at how CDISC standards can be applied to data in a variety of SAS® applications commonly used during the clinical development process. Data originates in an EDC or CDMS system and is transferred to the SAS Data Integration Studio product in a raw or ODM structure. It is then transformed into an SDTM format using a number of features in SAS DI. From there, the SDTM format is loaded into SAS Drug Development where it can be accessed for reporting and analysis.

Many of these steps can now be automated within the SAS tools, including comparison to the SDTM model, transferring the data from SAS DI to SAS DD, and making the SDTM model available to the Data Explorer component within SAS Drug Development. These technical features are explained, as are a number of alternative methods.

## INTRODUCTION

Many companies today are in the midst of re-engineering their process for clinical data analysis. The reasons for this vary but may include: cost concerns, increasing data volumes, corporate mergers and acquisitions, the need to work with more partners, a desire to downsize their internal IT support, the aging of existing technology, or compliance concerns. In what might be considered a “traditional” model, data was entered into a Clinical Data Management System where it was managed and cleaned. A handoff to an ETL tool might follow, where the data was transformed to a more analysis-ready format. This might also be when derived data values and any encoding would be performed. Another handoff step follows, this time to an analysis tool for reporting and, from there, on to a publishing tool. Along the way, each handoff was orchestrated using a homegrown data standard, where the two technologies both needed to be customized in order to work with the other. Typically, these “standards” either evolved over time or were multiple in numbers. Obviously, any change in process or technology is very costly in this model.

In recent years, two relatively new technologies have impacted many companies: external standards and Electronic Data Capture. This is not a paper about either, and there are many sources of information available on each (see the References section). In fact, most readers of this paper are undoubtedly already familiar with each. While any data standard could be used for reference purposes, CDISC is an obvious choice to reference when working with clinical data. The real key to a data standard, at least in the eyes of a technology vendor, is adoption. If the standard becomes **The Standard** then it is very easy for technology vendors to see the case for, at a minimum, supporting, and at a maximum, integrating, the standard into their solutions. Such is the case today with CDISC. There is likely not a single company in the pharmaceutical industry today that isn't at the very least looking at how the CDISC data models should be used within their organization.

Electronic Data Capture, some may argue, does not fall into the same category of importance as CDISC. The reason we raise it and are considering it so important is because of the ramifications the use of EDC has on the entire process. Companies today are considering eliminating the entire Clinical Data Management system, with an EDC tool filling the same basic requirements. This paper is not making a statement about whether that is the right approach to take, but the fact that companies are considering it has altered the overall process. If the data is being fed, virtually real time, into an EDC tool, why should data analysis not be performed almost real time as well? If a company is using EDC technology from multiple vendors, should they have to build an individual integration to each? And if an EDC tool is really being positioned as a replacement for a CDMS, where are tasks like discrepancy management being done?

We present one possible process which incorporates the concept of various data sources, the use of CDISC data models, and various technologies which allow for the automation of many stages of the process. The overall concept is to get the data in front of the people who need to see it as quickly as possible, while leaving open the possibility to change the data sources or components at any point without major upheaval.

## DATA SOURCES

Most EDC and CDMS tools available today allow for the extraction of data in a variety of formats. Not all support the CDISC SDTM or ODM formats, or they don't support the same versions. If we could count on all data coming into the process being in SDTM format, more than half the battle would be over. But even in the future if this becomes possible, there is still the problem of how to handle historical data which was collected in some other format.

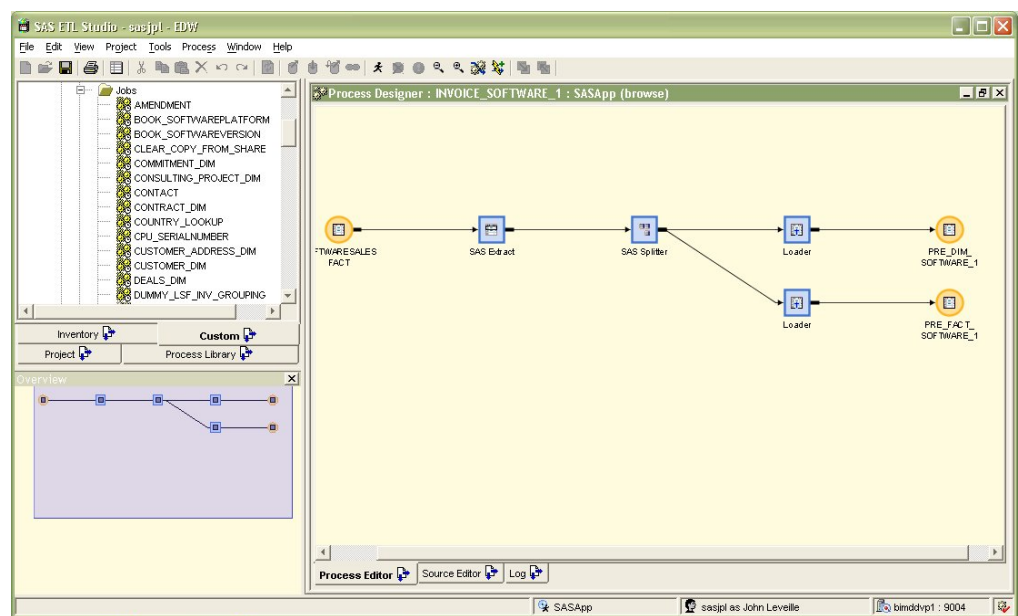
The data sources identified in this paper, therefore, cover a number of possibilities: data already in SAS datasets in some company specific format, data in a relational database, and data purported to be in a CDISC structure.

## SAS DATA INTEGRATION STUDIO

### WHAT IS SAS DATA INTEGRATION STUDIO?

SAS Data Integration Studio is a desktop application and the client portion of SAS Data Integration. It is a tool designed for the person whose job is to manage, cleanse, manipulate, transform, and/or move data. Data Integration Studio provides features that simultaneously organize and simplify these data access and data management tasks. If you have ever written SAS code to perform these kinds of tasks you have probably recognized the power and flexibility inherent in the SAS language. However, the language alone does not keep you organized. In addition, some readers may have experienced, on past projects, the evolution of their SAS code into an unwieldy collection of folders and programs that is difficult to support and understand.

Data Integration Studio sets forth a more structured paradigm for developing and managing a collection of SAS programs. In this environment you develop discrete units of SAS code and join them together into a job. Within the job you declare all of the code units as well as input and output data tables and the structure of these tables. Data Integration Studio allows the user to view a job as a complete SAS program or as a picture, using a graphical process flow diagram. It is immediately obvious how a picture of a job enhances organization, understanding, and ease of maintenance. Also, the process flow diagram can serve as a form of documentation for the work that you have done.



Another strong feature of Data Integration Studio is impact analysis. You can ask Data Integration Studio to perform impact analysis on a data node in your project. Impact analysis is essentially asking the question, "If I change this data then what other nodes may need to be executed to propagate the change?" You are asking Data Integration Studio to tell you what else in the project depends on this one element. Data Integration Studio can also perform reverse impact analysis which is asking the question, "What programs or data tables feed into this data node?" This is extremely useful when you are trying to track down the ultimate source of a data element.

Data Integration Studio can also help you write code. There are many tasks in this problem space that are routine -- tasks such as loading data from one table to another, joining data from two tables, generating record keys, transposing (also known as a table pivot) and more. Data Integration Studio has a library of typical data integration tasks that you can drop into your jobs. You customize the setting on these tasks and the Data Integration framework handles generating the appropriate code. In fact, Data Integration Studio goes even further by providing an extension API that you can use to develop your own custom transformations to meet your specific needs.

So how does Data Integration Studio do all this fancy stuff? How does it understand what is in your job? The answer is metadata. When you construct jobs in Data Integration Studio it records all of the relationships, parameter values, and even your custom SAS code in a SAS Metadata Repository. This collection of information is known as metadata. The metadata is then utilized by SAS to create process diagrams, perform impact analysis, and even run your jobs on a remote machine without the Data Integration Studio client.

## THE TYPICAL DATA INTEGRATION STUDIO USER

The typical Data Integration Studio user is someone whose job is to create or manage data for the business. In the pharmaceutical industry there is often a specific group within the organization that performs these duties. They are responsible for receiving study data, collecting and storing it, maintaining versions, archival, and preparing these data for analysis by biostatisticians. These individuals likely already know how to program in SAS or they have SAS programmers on their teams. They are aware of data standards and the importance of applying these standards within a data management process. They are responsible for creating analysis-ready data, and they are also required to accomplish this within some kind of reliable, repeatable framework.

The typical Data Integration Studio user is not someone that is in a hurry to create data. Creating data immediately is best done with an ad-hoc SAS program. This is not to say that using Data Integration Studio is slow. Rather, it is appropriate to say that creating data with Data Integration Studio is deliberate and methodical. This takes a little longer than simply writing some SAS code, but the benefits you get from the deliberate, well-defined construction of a job in Data Integration Studio will pay dividends for years to come through ease of maintenance, process documentation, and precise definitions of input and output data structures.

## MAKING INPUT DATA AVAILABLE TO DATA INTEGRATION STUDIO

SAS has a long tradition of being one of the best software packages for accessing data in external systems. Data Integration Studio continues the SAS tradition by leveraging all the capabilities of Base SAS, SAS/ACCESS, and SAS library services in products like SAS/CONNECT and SAS/SHARE to access remote data and data stored in external formats. The most common data storage formats in the clinical space are raw data in operating system files (ASCII text, XML, MS Excel), relational database data, and SAS data sets.

Raw data can be imported by a custom SAS program using the Data Step or the IMPORT procedure. Anyone who has used SAS has probably performed this exercise, if not once, many times. In the Data Integration Studio environment you can use all of your Base SAS programming knowledge to write programs that import raw data. After you develop an import program you can drop that code into your Data Integration Studio job and define one or more target tables to receive the output from the program. Defining the target table is the part that may seem tedious to the traditional SAS programmer who is new to Data Integration Studio. However, once this task is done, Data Integration Studio has a complete description of your target table metadata, and you can use all of the features of Data Integration Studio to continue processing these data. Raw data can also be imported using one of the external file wizards available in Data Integration Studio. If your external data is in a well-defined format then using one of these wizards is a good choice. Simply select the option in Data Integration Studio to define a new source data object. The application will present you with a list of input data types including a few different wizards for defining external file data.

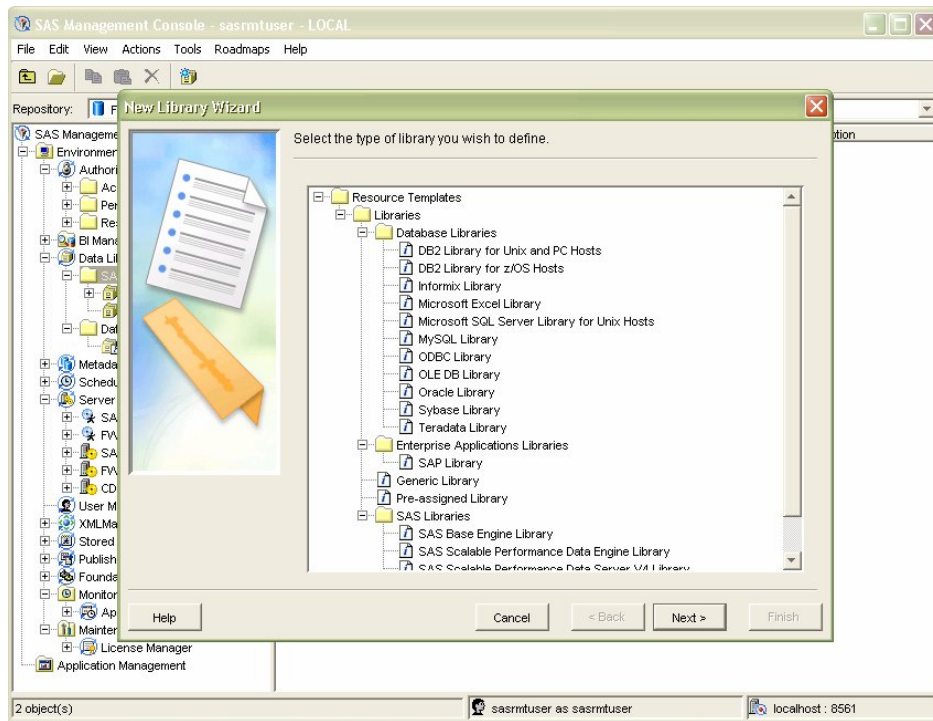
Relational databases and SAS data sets are well-defined, rectangular data stores and they are accessed through a SAS library definition. The traditional SAS programmer uses a libname statement in his/her program to define a SAS library. For example

```
/*Get some local SAS data*/  
Libname mysas "C:\projects\data1\sasdata";  
  
/*Get some data from the Oracle database instance 'DEV2'*/  
Libname myoracle oracle PATH="DEV2" username="" password="";
```

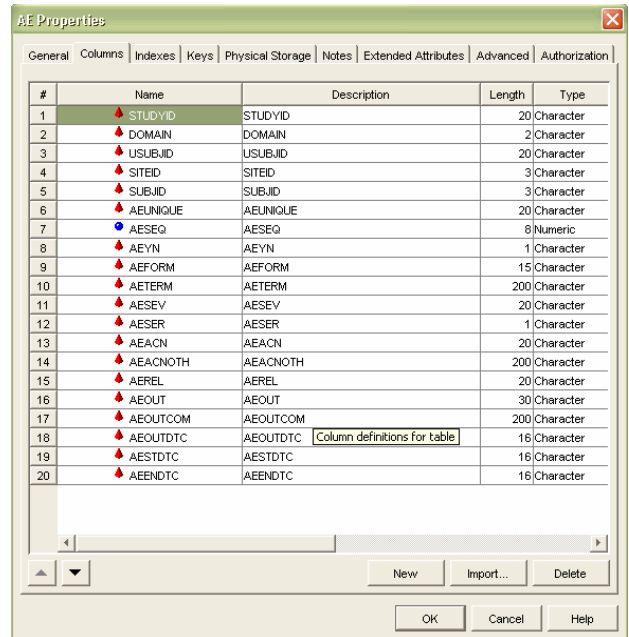
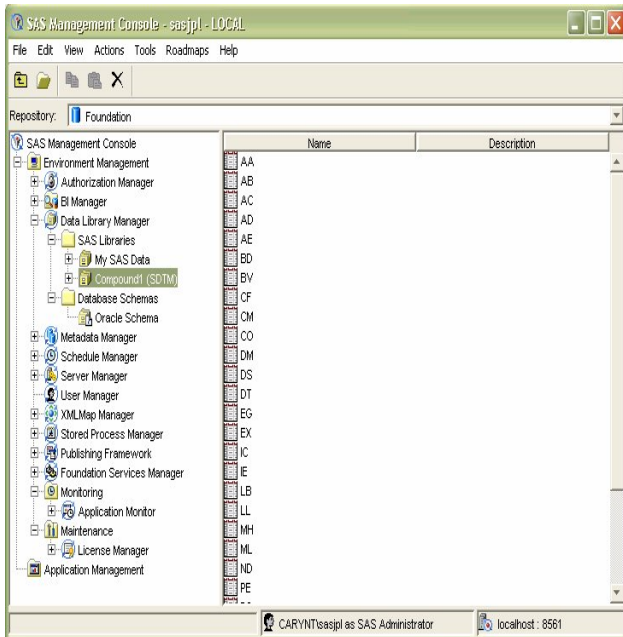
The problem with this approach in the Data Integration Studio environment is that Data Integration Studio is designed for the enterprise. This means that programs should be created without any direct ties to an individual user's workstation. In the example above, the *mysas* library assignment will only work on the user's workstation. If this library was assigned on a remote server the job will fail because the data would not be there. The library would get assigned to C:\projects\data1\sasdata on some remote server and that path probably does not exist there. In addition, making a library assignment for your own personal use doesn't allow others in your organization to take advantage of these data.

For these reasons, SAS Data Integration requires you to define SAS libraries using a definition that works across the enterprise. The SAS 9 platform gives you this capability by allowing you to define libraries within the SAS Metadata Server. The Metadata Server, the same component that manages the repository containing your Data Integration Studio metadata, also manages a global metadata repository called Foundation. The Foundation repository contains the metadata for all of your library definitions and tables.

The SAS Management Console is an application provided with SAS 9 for managing the Foundation repository metadata. The following screen shot shows the SAS Management Console along with a dialog for the New Library Wizard.



In order to define a library that includes either SAS or relational database data, you start the SAS Management Console and connect to your Foundation repository Metadata Server. If you want relational database data you define a database schema with login information. Next, you can step through a library definition wizard and define your library pointing to that database schema, or you can define a SAS data library pointing to an operating system folder. Once the library is defined you must perform an additional step and import the metadata for all of the tables in that library. Data Integration Studio must be able to access the tables in order to understand the metadata within them. After the library is defined and the tables are imported, you can select the library in SAS Management Console, see the tables, and drill down into the tables to view the column metadata to confirm the successful import. The following pair of screen shots shows a new library and tables in the SAS Management Console and the columns from the AE table as displayed through the table Properties dialog.



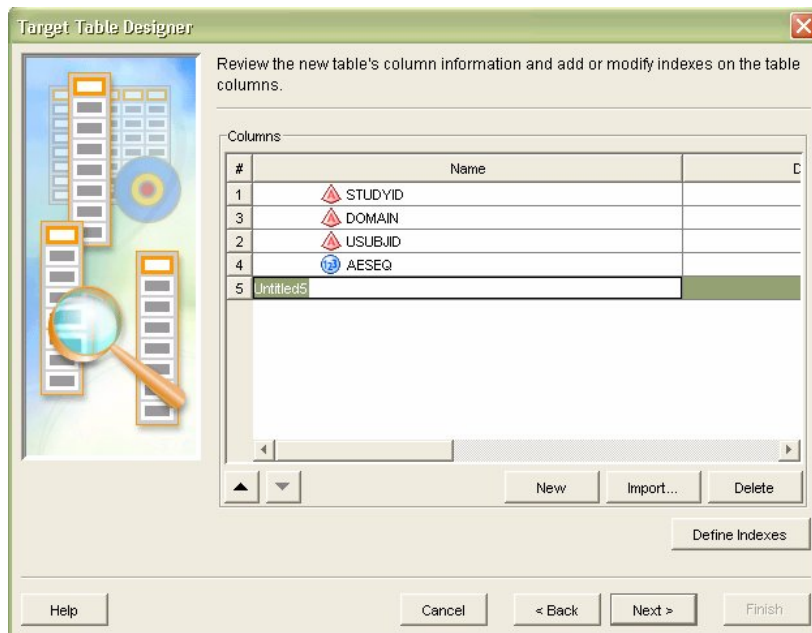
## MAKING CDISC METADATA MODEL AVAILABLE TO DATA INTEGRATION STUDIO

The CDISC Study Data Tabulation Model (SDTM) 3.1.1 is a published standard and a roadmap for alignment for many companies in the pharmaceutical industry. This standard represents an important, concrete definition that promises to ease the exchange of data between companies and between computer systems within and across companies.

Data management personnel within the pharmaceutical company are increasingly confronted with the requirement to interface their data processes with the SDTM model and Operational Data Model (ODM). Fortunately, Data Integration Studio is well-equipped not only to answer this need, but to ensure compliance with a standard and even assist you in moving to a new release of the standard in the future. (Every good standard deserves a revision, no?)

To start, you must make the CDISC model available to your Data Integration Studio project. There are a few different ways to do this:

- Manually create the SDTM metadata
- Point Data Integration Studio at existing data already in the SDTM structure
- Use PROC CDISC to read metadata in CDISC XML format and produce SAS metadata
- Import metadata from another repository

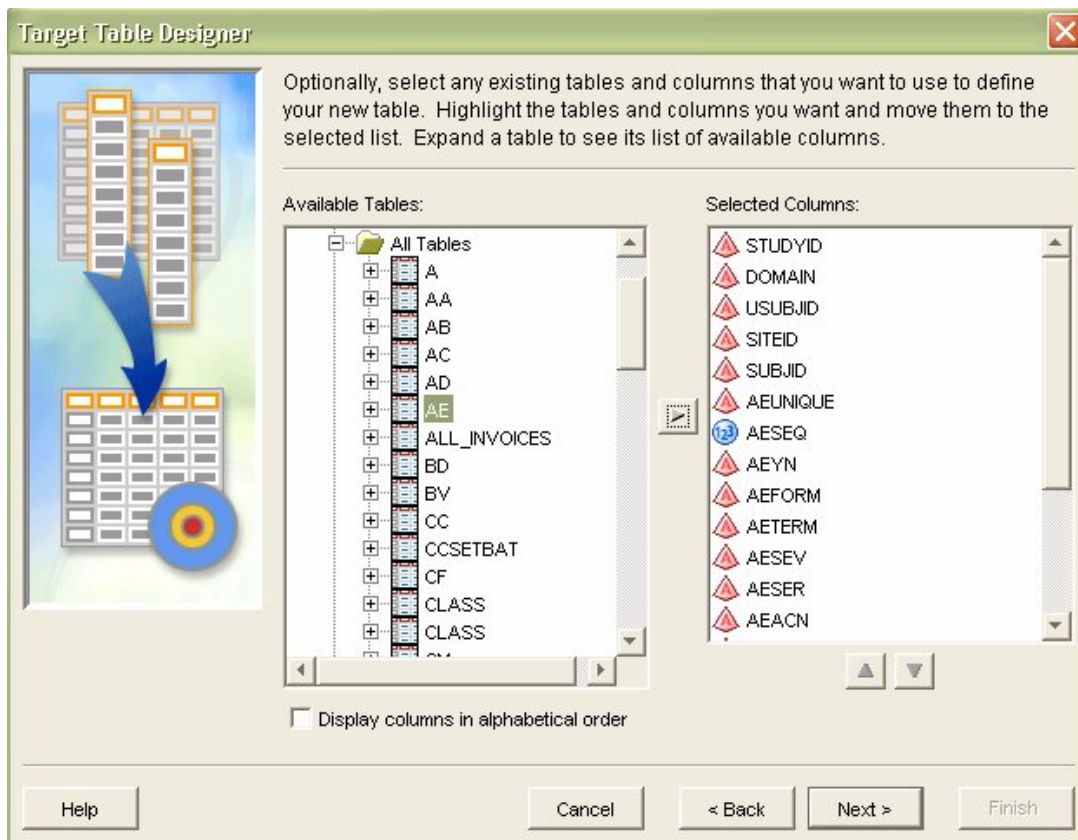


The SDTM model is a collection of domains (tables) and variables (columns) so creating it manually is simply a matter of defining all the tables and columns that you want to use within Data Integration Studio. SAS Data Integration Studio provides a series of point and click screens where you can define target tables by hand.

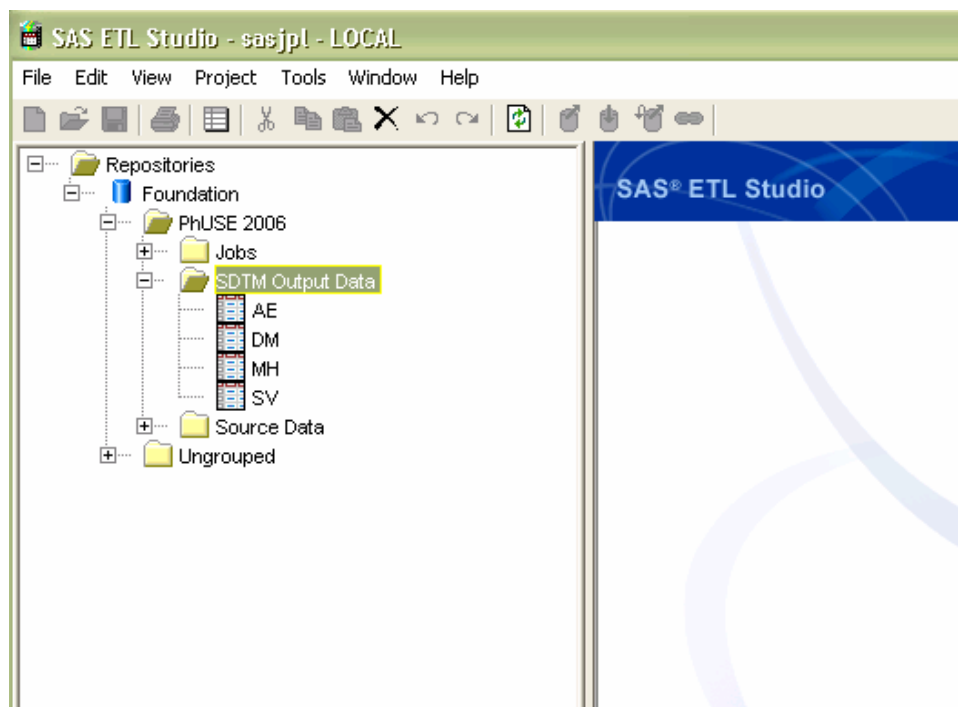
The process of manually creating metadata is quite time consuming. Though it is time consuming, it is not an unreasonable approach if you realize that the metadata you create can be copied and customized for other projects. It can be reused in many different jobs across your enterprise.

You will, however, save yourself a lot of typing if you have an existing set of SDTM data that is close to the data structure that you want. Data Integration Studio allows you to import this data structure and modify it to fit your needs.

The process is almost the same as manual metadata creation. As with manual metadata creation, you define new target table, and this invokes the Target Table Designer. Just before the screen where you manually enter columns, Data Integration Studio will prompt you to optionally select an existing table.



Selecting a table and clicking the arrow copies the column metadata from that table for use in your new target table. After you repeat this process for each table, you have built your own SDTM metadata model, and you are ready to begin processing data into those well-defined targets. The following screen shot shows a Data Integration Studio project after some target tables have been defined.



In typical SAS fashion, there are more options for creating SDTM metadata. SAS has been hard at work creating a new procedure called PROC CDISC. This procedure can be used to read data in CDISC XML format and create the metadata that SAS needs to operate on your data. This procedure is very powerful and is something that goes beyond the scope of this

paper. It is very important to mention it here because it can interoperate with the tools being described here to provide an alternate approach that is a good fit for organizations that are strongly aligned with CDISC standards for storing study data and metadata. SAS XML Libname technology, part of the underpinnings of PROC CDISC, is also worth mentioning for readers that anticipate the need to read custom XML data structures and utilize this metadata and data in a Data Integration Studio environment.

Another way of creating the CDISC metadata model is to copy it from an existing metadata repository. SAS Data Integration Studio has import and export functionality that makes it very easy for users within your enterprise to create and share metadata. In fact, the Health and Life Sciences division of SAS Consulting realizes that many customers will have need for the CDISC metadata, and they have defined this metadata and are offering it to customers as part of a service engagement. Also, SAS Consulting has gone beyond the basic CDISC metadata model. They have created custom wizards that extend the Data Integration Studio environment and allow you to define additional CDISC domains by simply pointing and clicking. The SDTM standard includes three domain classes: Interventions, Events, and Findings. Within these three classes are the standard domains: AE, DM, CM, etc. The wizards that SAS Consulting is offering allow you to define additional custom domains by selecting one of the three domain classes. When you select a domain class the wizard will utilize the SDTM metadata and any input options that you provide to create the metadata for the tables you need. Users can combine the standard SDTM metadata, these wizards for custom domains, and the stock transformations and quickly create a process for managing study data that is compliant with CDISC standards.

Whether you choose to use a manual approach, the new SAS CDISC procedure, or jump start with SAS Consulting, Data Integration Studio is a flexible, powerful focal point for your clinical data management activities.

### COMPARING INPUT STRUCTURE TO CDISC STRUCTURE

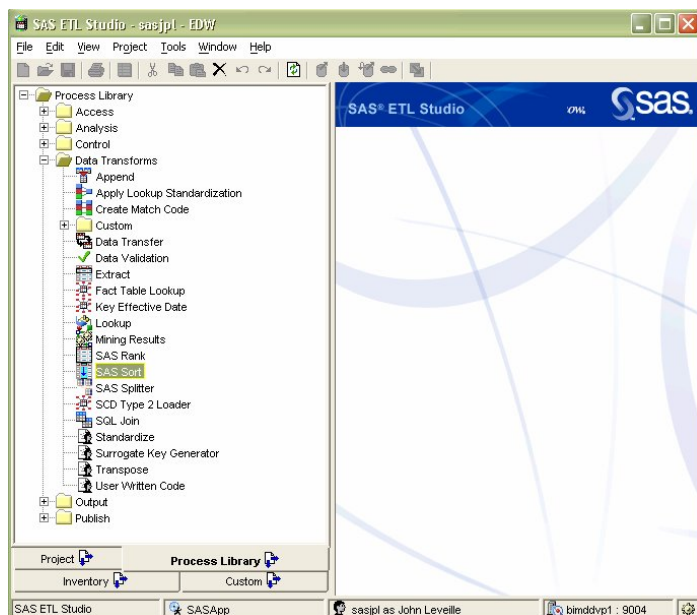
If you are loading input data from a SAS data library or relational database you can utilize the mapping features of Data Integration Studio to establish links between columns in source and target tables. When the job is invoked, the Data Integration framework will enforce the metadata mappings you have defined and will confirm, at each step in the job, that data structures inside the running job match the metadata as defined. So there is an implicit, continual comparison of data structures taking place at each step along the way. Finally, at the end of the job you will load the data into the final SDTM data format and the Data Integration framework will run a comparison check before it signals successful completion of the job.

If you are loading input data from external file sources you can utilize the SAS COMPARE procedure and reporting capabilities of SAS surfaced through Data Integration Studio to compare your input data to the defined CDISC structure. This is a good approach for each new project or study that you are processing with Data Integration Studio. Once you have aligned data structures and defined the mappings, Data Integration Studio will enforce the relationships you have defined each time your jobs run, and you can rest assured that the output data from your jobs is exactly as you have defined it.

Sometimes when you compare input structures to your CDISC model you will identify tables that require more than just a column to column mapping. There may be data that needs to be normalized, de-normalized, or transposed in order to fit your model. These operations are called transformations.

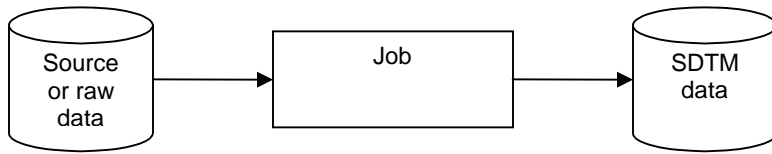
### TRANSFORMATION AND LOADING

The Data Integration Studio Process Library contains a number of transformations that you can apply to manipulate data for different desired end results.

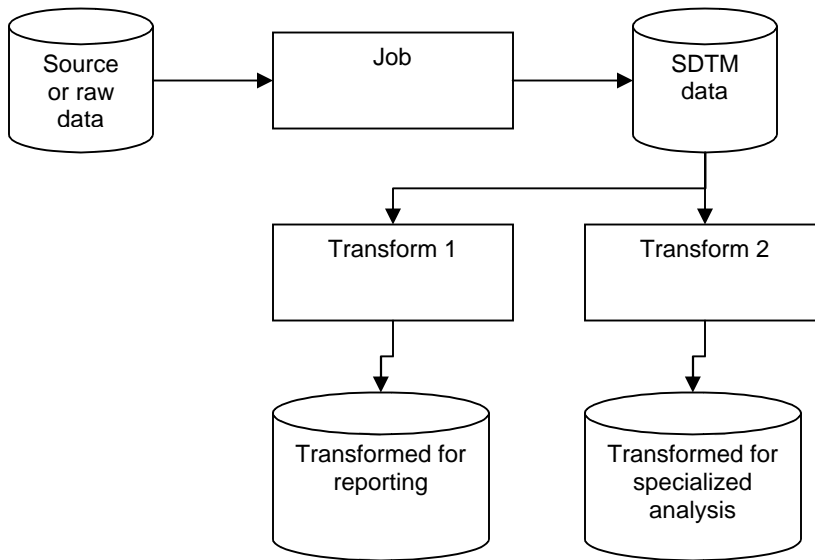


There are transformations for sorting, joining, splitting, and more. One common transformation example that you will see at this point in processing clinical data is a transpose or pivot. Many clinical data systems store lab domain data in a format that is different from the SDTM model. Your input data may define a record for a patient with laboratory test results stored in separate columns across that record. The SDTM model, however, contains a single column for the test name and requires a different record for each test performed. Therefore, you would define a job inside your Data Integration Studio project that contains a transpose step for the Laboratory Test Results domain (LB). Depending upon your input data you may also need additional steps in the lab transform job to clean the data before it can be loaded into your CDISC structured output table.

It is often the case that one standard set of data can be transformed into a variety of different related structures that are each well-suited for certain reporting or analysis activities. Up until this point we have been discussing a very simply logical flow that looks like this:



A more typical logical flow of job for processing study data might have multiple outputs. Like this:



This expanded flow illustrates how target table metadata can be used as source table metadata for other jobs. Just as Data Integration Studio allows you to reuse metadata that you have defined, it allows you to reuse transformations that you have defined. Suppose that you write some custom code for *Transform 1*. Later you realize that you need a new job that will do the same thing as *Transform 1* plus a little extra work. With Data Integration Studio you simply select the *Transform 1* node, copy it to your new job, and add a new code node for the extra work.

A Data Integration Studio job typically ends with a load step. This is where you define the final storage location for the data that you are creating, and you instruct Data Integration Studio how to load the data into that target. Again, Data Integration Studio assists you with a few different loaders that you can drop into your job. When you are loading data into SAS Drug Development you will use a loader in combination with a WebDAV library. The details for setting up a loader with a WebDAV library are discussed later in this paper.

### **AUTOMATION WITH JOB SCHEDULER**

The Platform Job Scheduler is an additional, optional component for SAS Data Integration. This component can be used to schedule jobs that you have developed in Data Integration Studio. If you think about the study life cycle you can imagine creating data integration jobs and running them on a regular schedule to pull fresh data into an analysis and reporting environment such as SAS Drug Development. The job scheduling mechanism is an important piece of the puzzle that helps bring you closer to real-time analysis and reporting for on-going clinical trials.

## **SAS DRUG DEVELOPMENT**

### **WHAT IS SAS DRUG DEVELOPMENT?**

SAS Drug Development (SDD) is a centralized, integrated system for managing, analyzing, reporting and reviewing clinical research information. SDD provides a web based environment for managing the data, analysis and results associated with the clinical development process.

SDD provides a 21CFR Part 11 compliant environment that supports version control and audit trails for every object stored in the system, ensuring traceability and reproducibility at all times. Point and click data exploration, intuitive organization and navigation, and searchable content provide researchers with direct access to the clinical research content in a collaborative

framework. SDD stores metadata for each object in the system and allows for the enforcing of metadata standards on incoming data. The system supports the use of views and reports on standardized data as well as the pooling of data sets or studies.

The SDD suite of applications consists of a data browsing tool (Data Explorer), SAS programming environment (Process Editor), an application to batch submit SAS code (Job Editor), the ability to see all the input and output from running a SAS program or batch job (Job Results Viewer), and an application to Schedule batch jobs (Scheduler). There are other SDD applications including an application to perform metadata comparison across multiple protocols to ensure consistency and an import/export application that can also convert SAS datasets to a variety of other formats.

#### **WHAT IS SAS DRUG DEVELOPMENT USED FOR?**

SDD provides easy and efficient access to data, documents and reports for a broad range of users across multiple locations and can serve as the central repository for all of the electronic information for a clinical research program. Information within SDD can be organized in a file hierarchy allowing authorized user access to the parts of the research content relevant to their work. The hierarchy can be customized at all levels based on the individual project requirements. Because the system is web-based and accessible through a standard web browser, users can access the research content regardless of where they are geographically located. This allows a collaborative environment for users in various departments, other geographies or other service providers.

Within clinical research organizations, physicians, medical writers and other scientific researchers typically have limited direct access to the clinical research data. Through SAS Drug Development, it is possible to create parameterized reports to answer common questions for these users as well as use the Data Explorer for ad hoc querying.

SAS Drug Development automatically provides the necessary controls and compliance for the data transformation and analysis process. A thorough history of each programming run (along with relevant inputs and outputs) is automatically captured in a job log and is available for rerunning with all of the original data or with the most recent version.

Data adhering to a standard metadata structures, such as the CDISC SDTM, can be loaded into SDD and then used with generic study definitions, clinical data views, and reports. The ability to reuse study definitions, views, and reports for standard data makes the data more immediately accessible to users and reduces validation time.

#### **THE TYPICAL SDD USER**

Several different user profiles access the data and applications in SDD. Technical users include both clinical and statistical programmers. The clinical programmer is responsible for writing SAS programs that extract data from a source data base and transform it to match an established structure. The statistical programmer is responsible for writing SAS programs, as well as organizing the programs to run in sequence via a batch mechanism. These programs produce tables, figures and listings that indicate the statistical outcomes of the trial. The statistical programmer constructs both non-interactive and interactive programs. Other non-technical users would include clinicians and medical writers.

The clinician is responsible for reviewing, summarizing, and interpreting reports that are produced. These reports will likely be included in a larger textual report summarizing the outcome of a clinical trial, or used as decision-making tools to further understand the safety and/or efficacy of a trial. In some cases, these reports may be used to plan future trials, or to identify clinical trends that are appearing within the patient population.

#### **MAKING YOUR CDISC DATA AVAILABLE TO SDD**

Recall that Data Integration Studio provides you with an excellent environment for controlled, deliberate creation of CDISC SDTM data. Now you want to make these data available to users within SAS Drug Development and allow them to analyze, report, and explore the results of your organization's clinical trials. The process of publishing output data from your Data Integration Studio job is quite simple. As the last step in your job, you define a loader to a target table for each output data set. These target tables reside in a SAS library on your SDD server. It is really that simple from the perspective of the user working in Data Integration Studio.

Since the SDD server could be located anywhere on the Internet, the SAS library definition is a little more complicated. The library must be defined with a remote protocol, namely something called WebDAV. You can think of WebDAV as a mechanism for reading and writing files to a web server, and the SAS data library will use WebDAV to read and write SAS data sets.

Diving further into the details, each file within SDD has a unique WebDAV URL. The typical SDD user is very familiar with URLs to access the SDD web interface. For example, suppose your SDD server has the name `sddmyserver.sas.com`. You will use the following URL to log in to the SDD web interface.

<https://sddmyserver.sas.com/p21>

Within the SDD web interface you will see a hierarchy of folders and files. One folder might have the path

/SDD/DataIntegration/

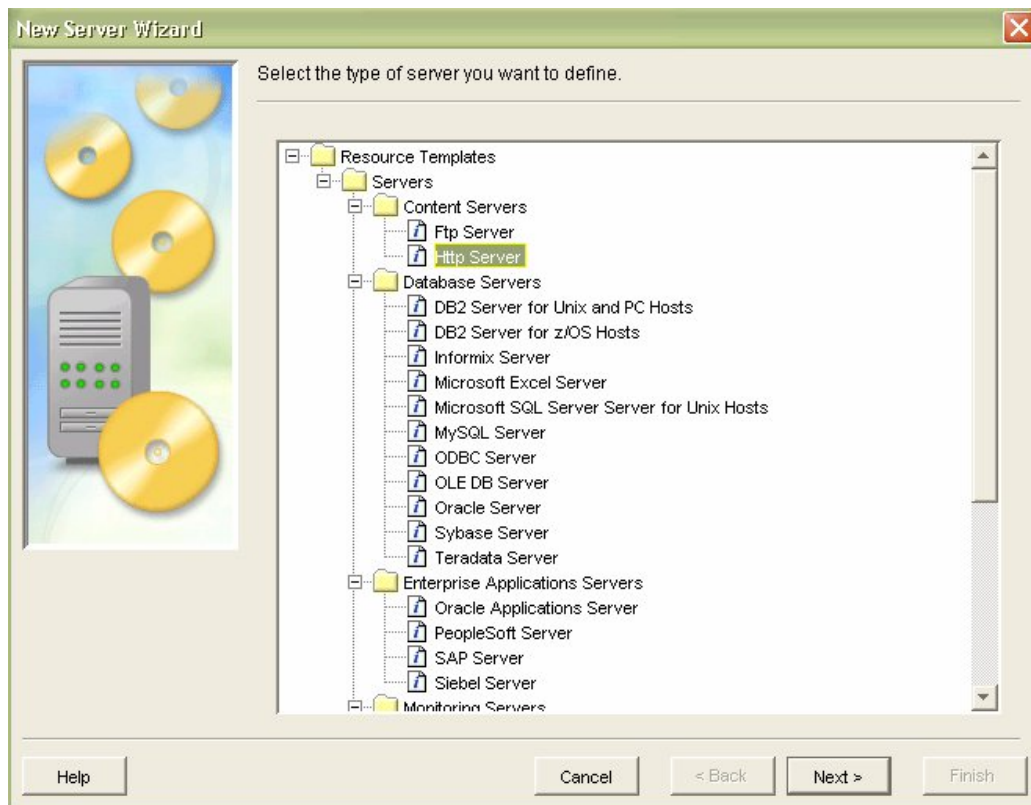
In this example, the complete WebDAV URL for this folder is

<https://sddmyserver.sas.com/webdav/SDD/DataIntegration>

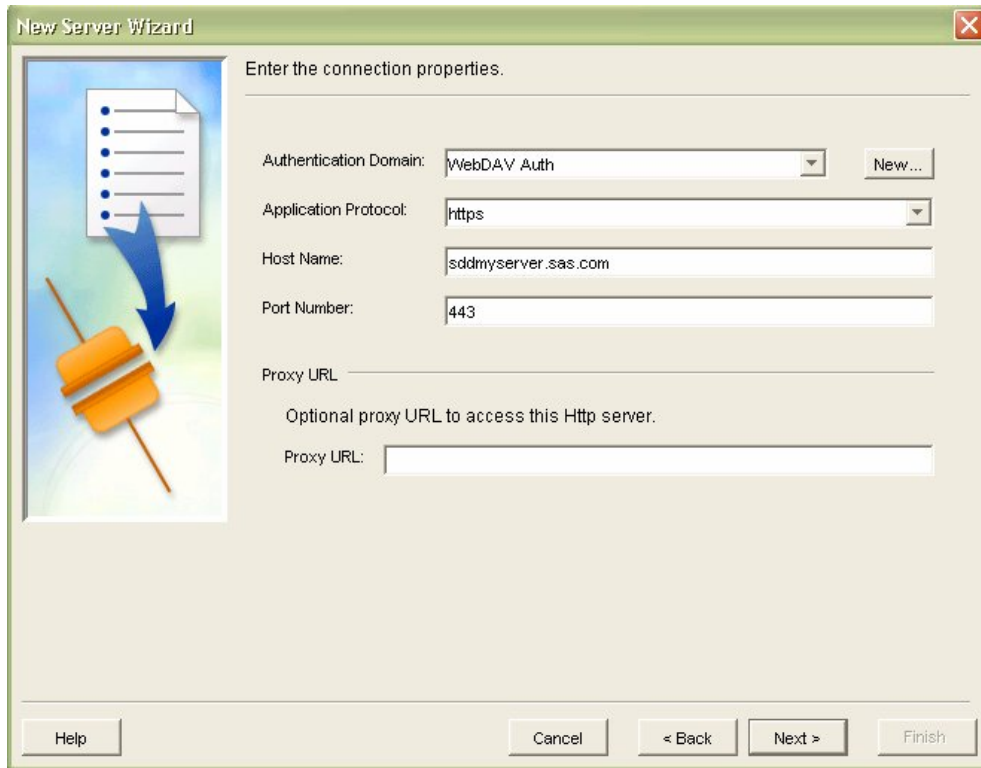
This URL breaks down into the following pieces:

Protocol	https
Host name	sddmyserver.sas.com
Port	443 (same as <a href="https://sddmyserver.sas.com:443">https://sddmyserver.sas.com:443</a> – the number 443 is assumed when https)
URI	/webdav/SDD/DataIntegration

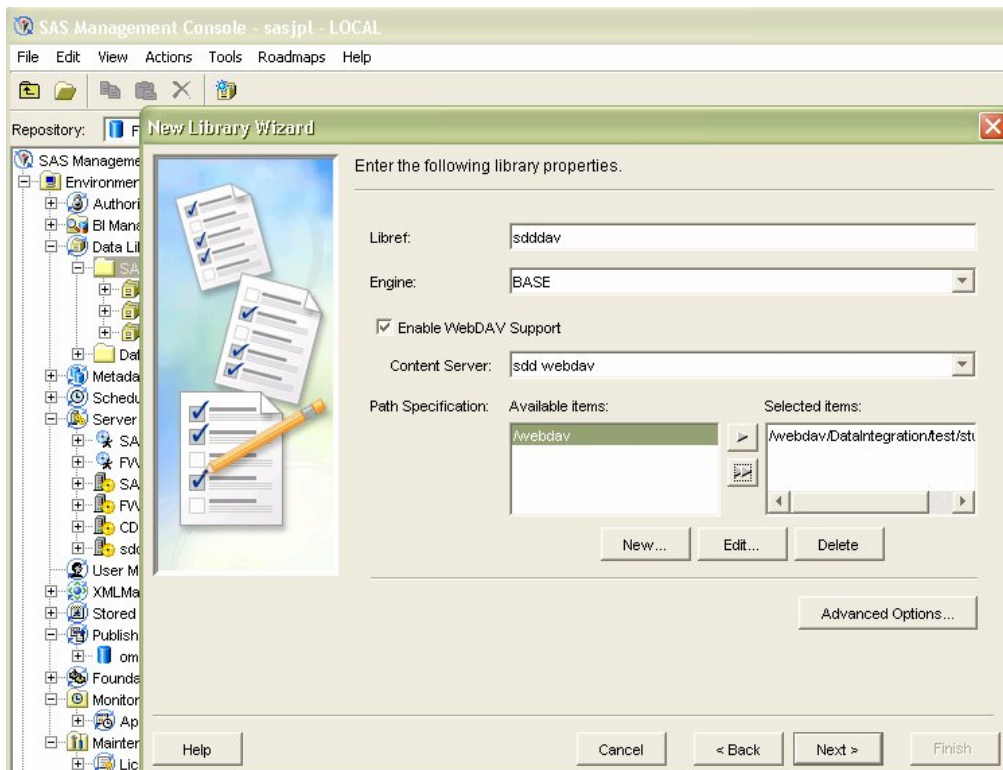
You need to know these pieces of information in order to set up the SAS library using WebDAV. The first step in setting up the SAS library on your SDD server is defining the SDD server in the SAS Management Console.



Right click on the Server Manager node and select New Server. Select Http Server from the list of possible servers. In the next screen define the authentication domain (this is where your username and password are stored in the metadata) as well as the protocol, host, and port. The protocol is always https when using SDD because this is an essential part of the security infrastructure which keeps your clinical data safe.



Once the server has been defined you can create a new SAS library using SAS Management Console. This is done through the same wizard that you used to create the SAS libraries used as input to your Data Integration Studio jobs. However, on the wizard screen where you provide the library details you check the box that says “Enable WebDAV Support.” Then use the drop down list box provided to select the server that you just defined. Under the section for Path Specification, supply the URI path to your folder within the SDD server. The URI is the part of the URL that comes after the host name and optional port number.



Once the SAS library with WebDAV support is defined in the SAS Management Console you can use it in your Data Integration Studio job. Just return to the Data Integration Studio environment, run your job, and you will see your CDISC data appear in the SDD server at the location you specified. With your study data inside the SDD server you are ready to create an SDFXML file and explore the data with Data Explorer.

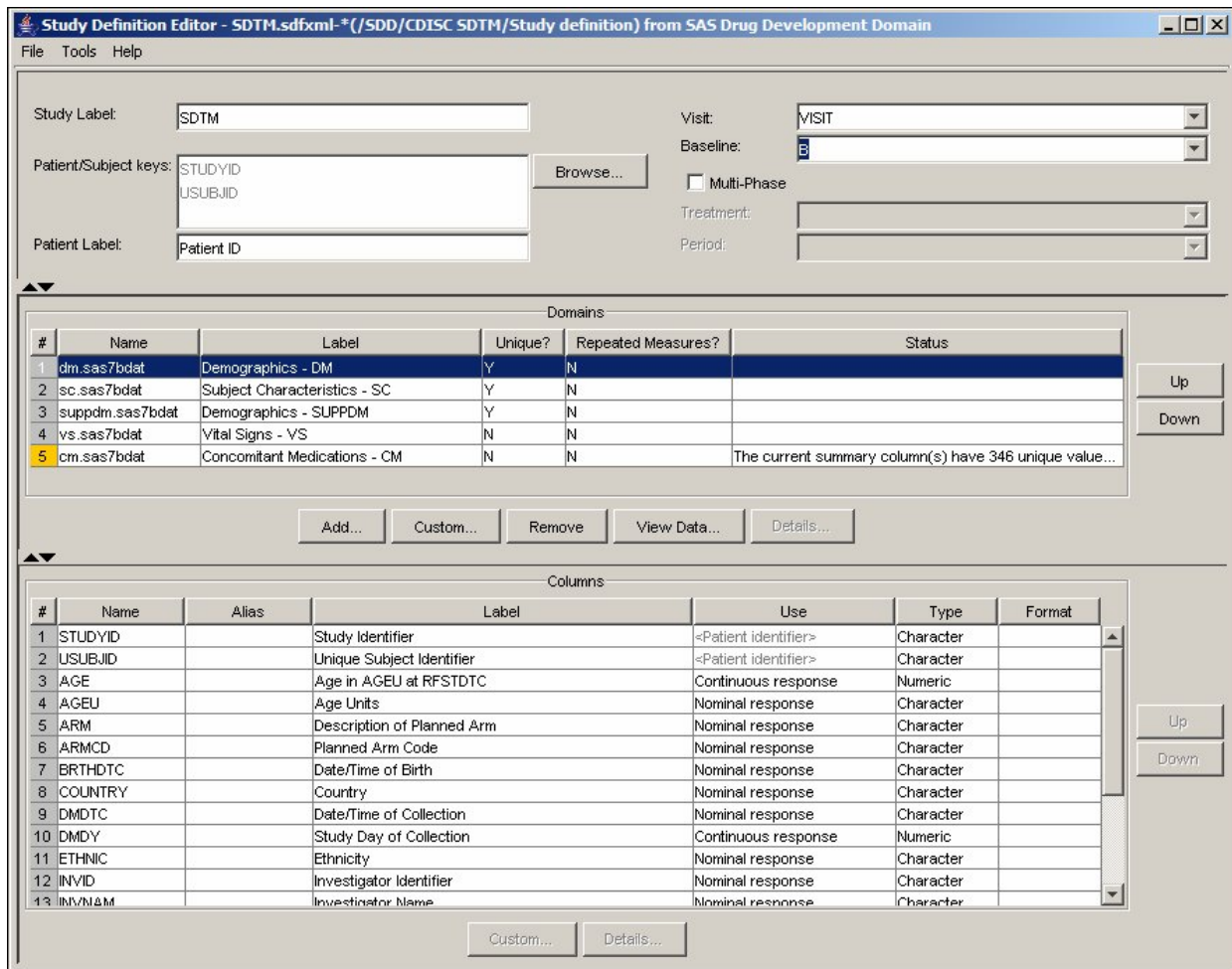
**WHAT IS THE SDFXML AND WHY IS ONE NEEDED?**

The Data Explorer is an application in SDD that enables browsing and exploration of study data. Users can view data in a row-and-column format or as summary graphs. Capabilities include dynamic sorting, filtering and color grouping to help identify specific clinical outcomes.

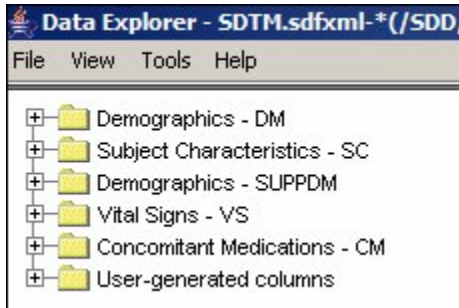
Users can view individual data tables or multiple data tables by using study definitions.

A study definition in SDD defines data sources, groupings and columns to include in a view of study data. The data is always transformed into a view of one row per patient across all data sources.

A study definition consists of XML tags that describe the basic metadata model of a study. The tags for a study definition are stored in a text file with the extension SDFXML. XML tags define parameters such as the common patient variable, data set names, and variable attributes. SDD provides a Study Definition Editor application for creating and modifying study definitions. The following screen shot shows the study definition editor being used to define a study consisting of five SDTM domains (Demographics, Subject Characteristics, Supplemental Demographics, Vital Signs, and Concomitant Medications.) The data domains have been defined along with the patient keys, visit variable and baseline visit value.



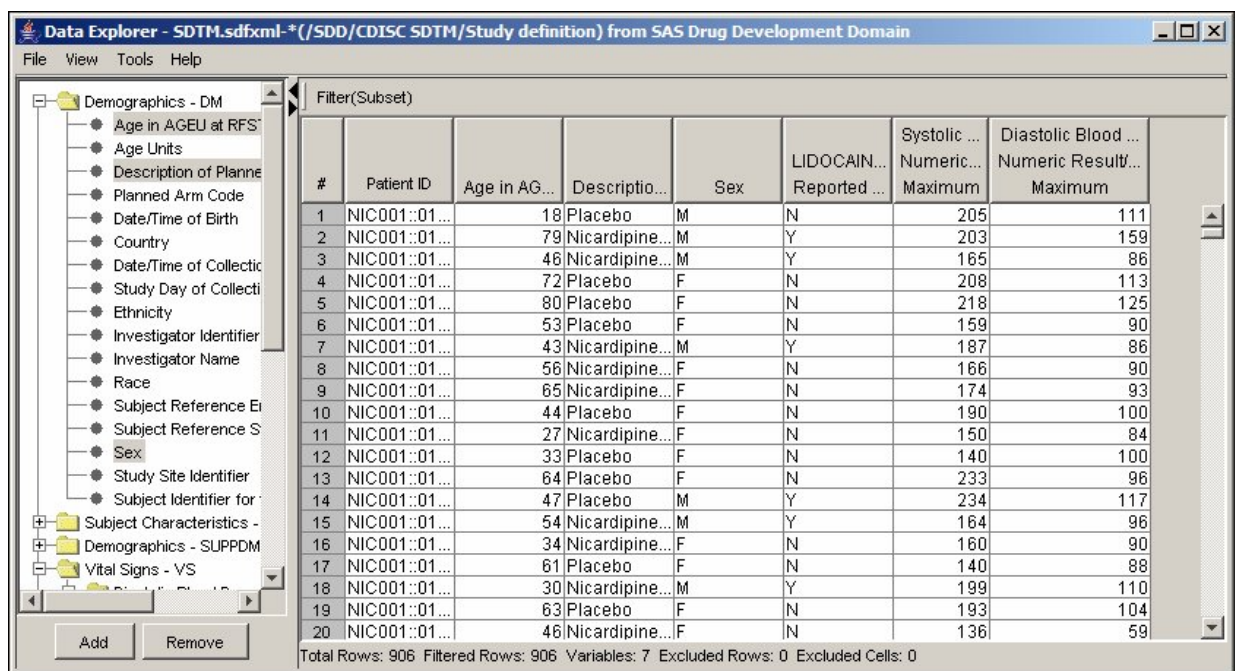
The Data Explorer can be used to view the various domains included in SDTM data as illustrated below. The data domains defined to the study definition appear as top-level folders in the Data Selector panel. Expanding a folder in the Data Selector panel displays columns in that data domain. The Data Explorer can open multiple study definitions at one time, providing a way to view patient data across multiple studies.



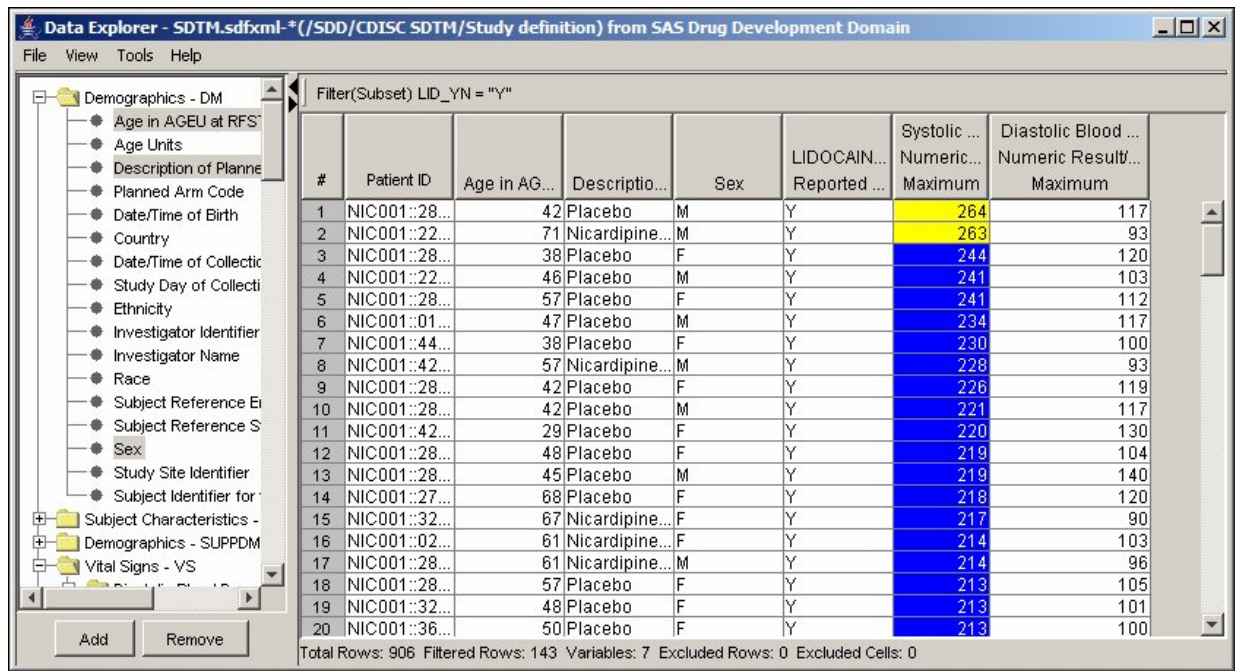
Study definitions can be customized to present the same data in many different ways. For example, clinicians may be more interested in browsing patient data, so a study definition would be set up specifically to include those data sources and variables. A biostatistician may have a completely different requirement for looking at the same protocol. Through well-designed study definitions both groups of users can be accommodated

When using a study definition, The Data Explorer automatically performs the necessary transformations to create a view of one row per patient. The following screen shot shows the Data Explorer displaying variables from the Demographic, Concomitant Medication and Vital Signs domain. The data is presented one record per patient with the status bar indicating that there are 906 patients in this study.

For visit level data, such as Vital Signs blood pressure, the data is summarized over the visit (e.g. maximum or minimum systolic blood pressure). Individual visit values are also available as separate columns. For event level data, such as the Concomitant Medication data, there is a separate column indicating whether or not the event occurred as well as a column giving the actual number of occurrences. In this example a column indicates whether or not the patient took the medication Lidocaine.

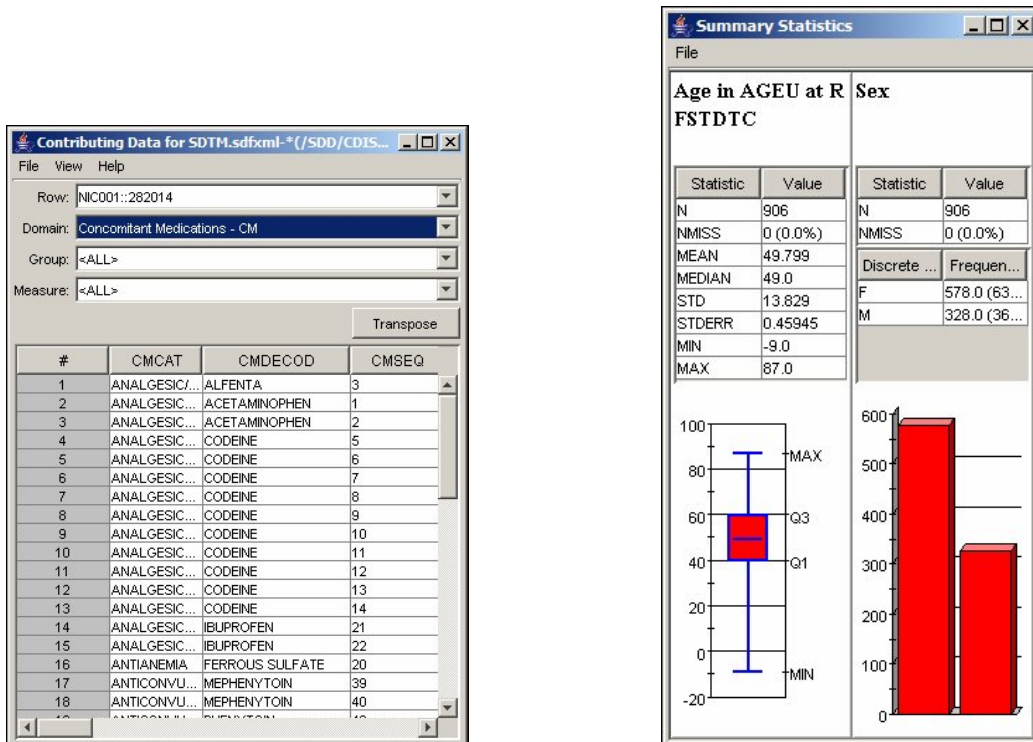


The Data Explorer allows the user to apply a filter to the data creating a new subset. Users can also create color groups which are a way to color code groups of data values. The following screen shot shows a filter applied to the data, so that the only patients appearing in the table are the 143 patients taking Lidocaine. Color coding was also applied to Maximum Systolic Blood Pressure and the data table was sorted by this variable. This resulting filtered data can be saved as another data set or as a clinical data view.



The Data Explorer also allows the user to drill down to view the contributing or underlying data for a patient or group of patients. The following screen shot shows all of the Concomitant Medication data for one patient who was in the “yellow” range for Maximum Systolic Blood Pressure.

Summary statistics can be generated on columns of data in the Data Explorer. A vertical bar chart is produced for nominal/ordinal variables and descriptive statistics and a box-and-whisker plot is generated for continuous variables. The analysis types of nominal, ordinal or continuous are assigned to each variable in the study definition.

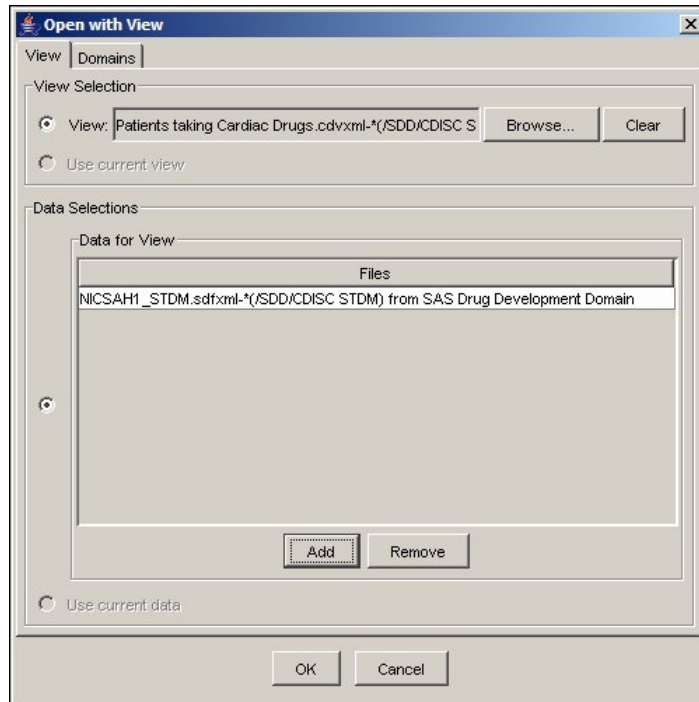


When viewing data in the Data Explorer, users can choose to explore a specific domain in the study, changing to a view of the actual data set. In the following screen shot, a new Data Explorer window contains the Concomitant Medication data and a filter is applied so that only Cardiac Drugs appear in the table. This domain-level filter can also be applied to the patient-level view of the data, so that the resulting data display contains patients who have taken a cardiac drug.

#	Study Identifier	Unique Subject Identifier	Sequence Number	Standardized Medication Name	Category for Medication
1	NIC001	011001	31	LABETALOL HCL	CARDIAC DRUGS
2	NIC001	011001	52	PROPRANLOL	CARDIAC DRUGS
3	NIC001	011002	13	LABETALOL HCL	CARDIAC DRUGS
4	NIC001	011002	14	LIDOCAINE (CARDIAC)	CARDIAC DRUGS
5	NIC001	011003	39	LIDOCAINE (CARDIAC)	CARDIAC DRUGS
6	NIC001	011003	40	LIDOCAINE (CARDIAC)	CARDIAC DRUGS
7	NIC001	011003	41	LIDOCAINE (CARDIAC)	CARDIAC DRUGS
8	NIC001	011003	42	LIDOCAINE (CARDIAC)	CARDIAC DRUGS
9	NIC001	011003	43	LIDOCAINE (CARDIAC)	CARDIAC DRUGS
10	NIC001	011003	44	LIDOCAINE (CARDIAC)	CARDIAC DRUGS
11	NIC001	011004	30	LABETALOL HCL	CARDIAC DRUGS
12	NIC001	011005	8	CAPTOPRIL	CARDIAC DRUGS
13	NIC001	011005	21	DIGOXIN	CARDIAC DRUGS
14	NIC001	011006	36	LABETALOL HCL	CARDIAC DRUGS
15	NIC001	011007	32	LABETALOL HCL	CARDIAC DRUGS
16	NIC001	011007	33	LIDOCAINE (CARDIAC)	CARDIAC DRUGS
17	NIC001	011008	7	CAPTOPRIL	CARDIAC DRUGS
18	NIC001	011008	16	DIGOXIN	CARDIAC DRUGS

Total Rows: 34589 Filtered Rows: 1270 Variables: 13 Excluded Rows: 0 Excluded Cells: 0

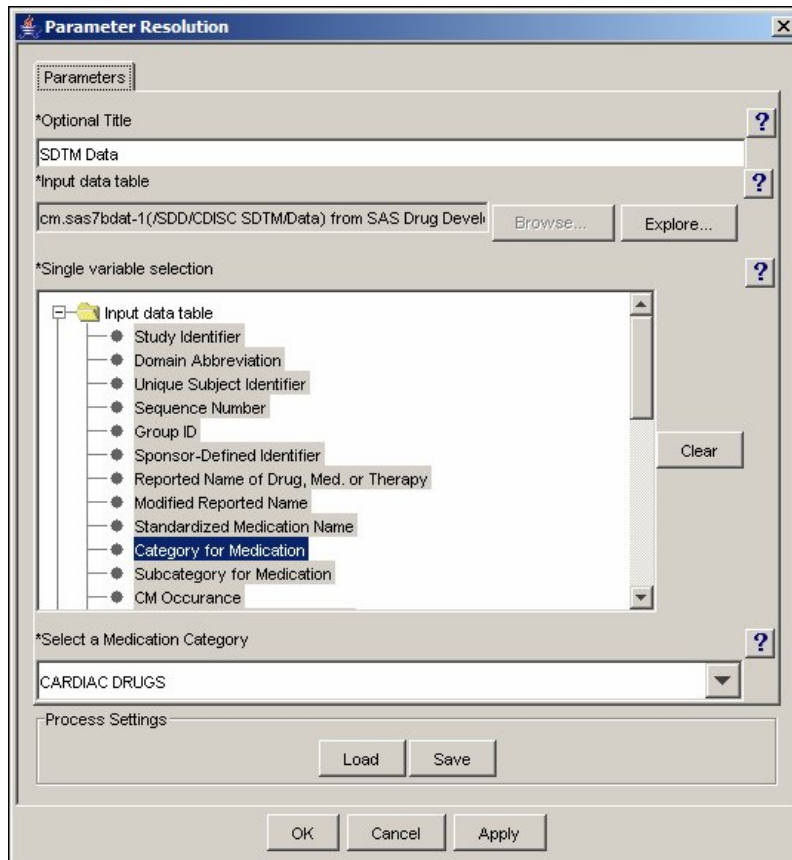
This clinical data view can then be saved to SDD and reused on any study or groups of studies that have data in SDTM format. In this example the clinical data view “Patients taking Cardiac Drugs” is saved with the NICSAH1\_SDTM study as the default input to the view. Multiple studies can be used as input to a view. A view can also be used as input to a report.



### RUNNING REPORTS IN SDD ON THE DESIRED CDISC STRUCTURE

Once the desired CDISC structure is loaded into SDD, users can also run reports on this data. The Process Editor is used to develop standard reports with automatically constructed user interfaces that prompt the end-user to specify query and reporting parameters. In this way, libraries of standard reports can be created by technical users and then run as is, or with new values for the substitution parameters

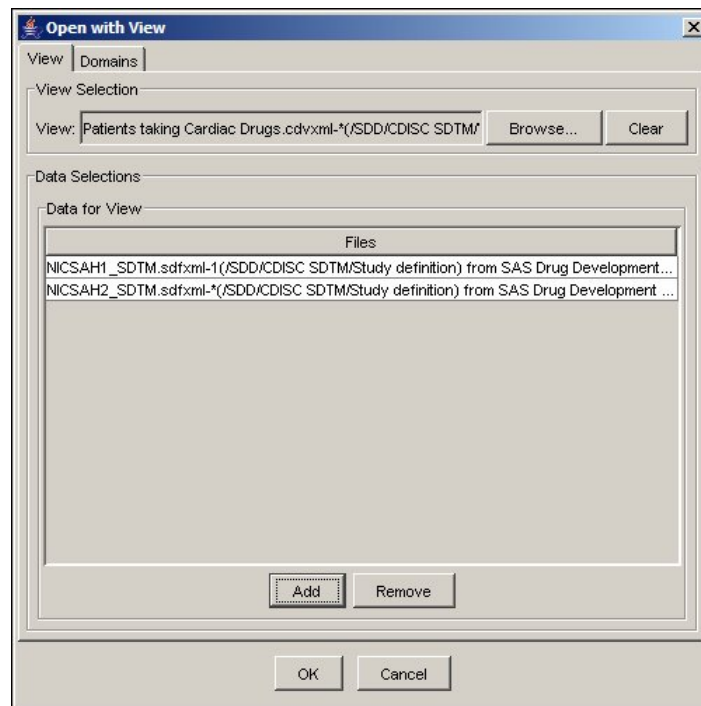
The following screen shot shows the Parameter Resolution window for a report that generates a frequency of concomitant medications for a selected drug category. In this example the report will produce a frequency report of Concomitant Medications in the Cardiac Drug category. The input data to this report is the SDTM CM data set. The programmer can also allow the user to create a view of this data set interactively by selecting the “Explore” button next to the input data source.



This report uses ODS to produce an RTF output file, which is saved in SDD.

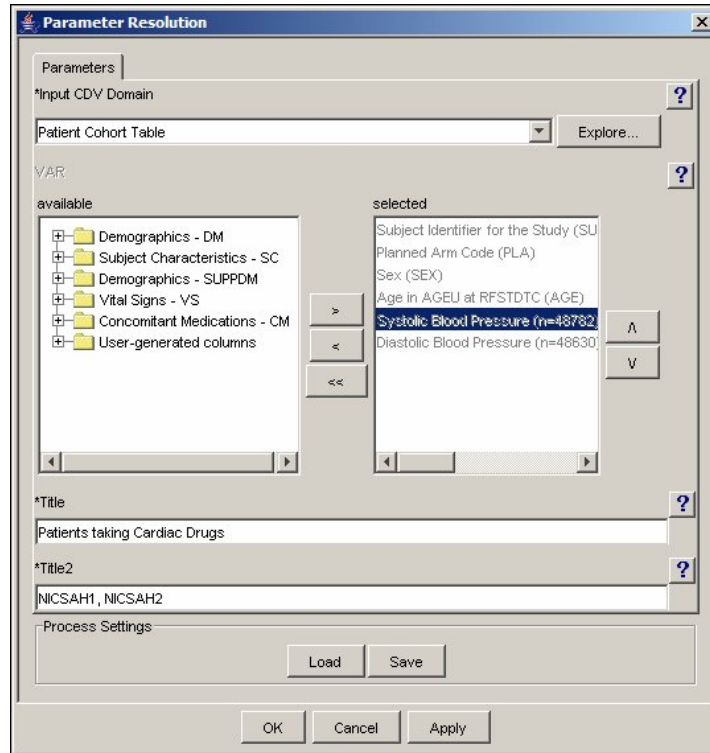
CM Frequency Report				
CARDIAC DRUGS				
SDTM Data				
01SEP06				
<i>The FREQ Procedure</i>				
Standardized Medication Name				
CMDECOD	Frequency	Percent	Cumulative Frequency	Cumulative Percent
ATENOLOL	23	1.81	23	1.81
BRETYLIUM	3	0.24	26	2.05
CAPTOPRIL	78	6.14	104	8.19
DIGOXIN	173	13.62	277	21.81
DILTIAZEM	3	0.24	280	22.05
DISOPYRAMIDE	2	0.16	282	22.20
ESMOLOL	29	2.28	311	24.49
LABETALOL HCL	578	45.51	889	70.00
LIDOCAINE (CARDIAC)	223	17.56	1112	87.56
PROCAINAMIDE	18	1.42	1130	88.98
PROPRANCLOL	89	7.01	1219	95.98
QUINIDINE	10	0.79	1229	96.77
VERAPAMIL	41	3.23	1270	100.00

The following example shows a clinical data view being used as input to a report that allows a user to select variables for a patient listing. The clinical data view includes only the patients who have taken cardiac drugs. The NISAH1 and NISAH2 study definitions are used as input to this view. When a clinical data view or study definitions are used as input to a report, the input data can be pooled across studies.



This report lets the user select variables from any of the domains in the study. In this example, variables from the demography domain and vitals signs domain will be printed. The Patient Cohort table is used as input so the data will be presented one

record per patient. The Vital Signs data is summarized over the visits for each patient.



The PDF output file shows the output listing from the report. The data is presented as one record per patient. The maximum values of systolic and diastolic blood pressure are given for each patient.

*Patients taking Cardiac Drugs  
NISCAH1, NISCAH2*

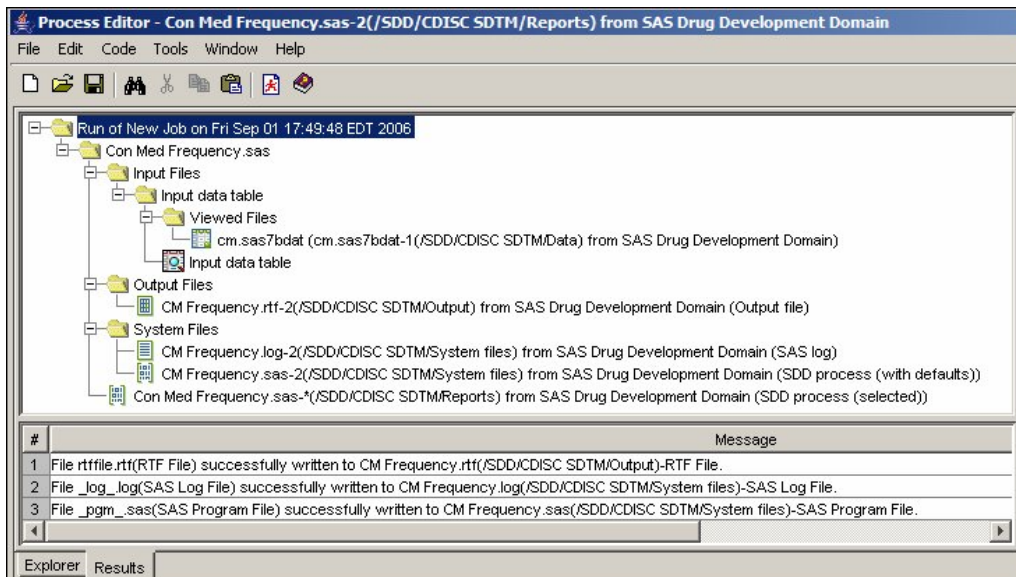
1

Obs	subject	Treatment Group	Sex	Age	Maximum Diastolic Blood Pressure	Maximum Systolic Blood Pressure
1	001	PLA	M	18	111	205
2	002	NIC15	M	79	159	203
3	003	NIC15	M	46	86	165
4	004	PLA	F	72	113	208
5	005	PLA	F	80	125	218
6	006	PLA	F	53	90	159
7	007	NIC15	M	43	86	187
8	008	NIC15	F	56	90	166
9	009	NIC15	F	65	93	174
10	010	PLA	F	44	100	190
11	011	NIC15	F	27	84	150
12	012	PLA	F	33	100	140
13	013	PLA	F	64	96	233
14	014	PLA	M	47	117	234
15	015	NIC15	M	54	96	164
16	016	NIC15	F	34	90	160
17	017	PLA	F	61	88	140
18	018	NIC15	M	30	110	199
19	021	NIC15	F	63	119	221
20	023	NIC15	M	45	100	188

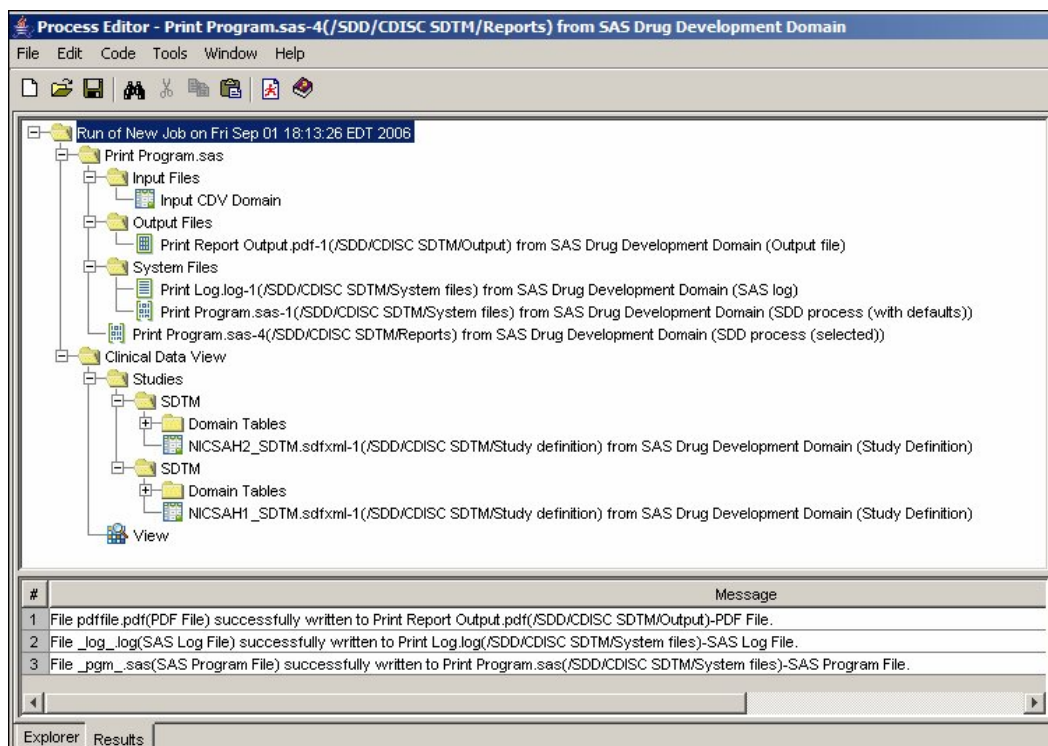
**SAVING AND PUBLISHING THE RESULTS**

The job log, a standard feature of SDD, allows for reproducibility. A job log is created when reports are run interactively or as part of a scheduled job. A job log typically contains links to any input and output data sources or views, SAS system files, and analysis and reports created by the report. At any point in time, the contents of a job log can be re-opened for review/reproducing, or for execution on a new version of the data.

These traceability and documentation capabilities are valuable for providing ongoing quality control, supporting quality assurance activities, and when addressing questions from regulatory agencies. The following screen shot shows the job log from running the Concomitant Medication frequency report program on the SDTM concomitant medication data. The log contains links to the specific version of the input, output and system files used in this report.



The following job log is from the patient listing report. The job log shows the NICSAH1\_SDTM and NICSAH2\_SDTM study definitions that were used as input to the report allowing the data to be pooled.



## CONCLUSION

As always, there are a number of different ways to implement a process for handling clinical data. The example methods presented in this paper is not as important as the understanding the basic guidelines that were followed:

- Industry standards, when commonly adopted, are good and should be followed wherever possible
- Data should make it's way from source to analysis and reporting as quickly as possible
- custom integrations should be eliminated wherever possible
- the process should require as few human interactions as possible along the way
- the process needs to be flexible and allow for changes over time

CDISC is most likely past the "emerging standard" phase and is quickly approaching "the standard". EDC is taking off, and together with CDISC, can radically change the way data is moved to an analysis and reporting environment and when it is moved. And SAS, as the traditional choice for analysis and reporting, has worked to integrate well with these new technologies in order to promote more efficient data analysis.

## REFERENCES

Clinical Data Interchange Standards Consortium (CDISC), <http://www.cdisc.org>

Kilhullen, Michael, "Implementing CDISC Data Models in the SAS Metadata Server", Paper SA01, PharmaSUG 2006 Conference Proceedings, <http://www.lexjansen.com/pharmasug/2006/sasinstitute/sa01.pdf>

Olinger, Chris and Weeks, Tim, "The Ins and Outs of SAS ETL Studio", Paper ET02, SUGI30 Conference Proceedings, <http://www2.sas.com/proceedings/sugi31/260-31.pdf>

## ACKNOWLEDGEMENTS

Special thanks to Nancy Cole and Michael Kilhullen, both from SAS Institute, for their contributions to this paper. Nancy

provided SDD study definition and reporting expertise and Michael is the pioneer of CDISC SDTM in SAS Data Integration Studio.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author(s) at:

Andrew Fagan  
SAS Institute, Inc.  
SAS Campus Drive  
Cary NC 27513  
Work Phone: 919-677-8000  
Fax: 919-677-4444  
E-mail: [andrew.fagan@sas.com](mailto:andrew.fagan@sas.com)  
Web: <http://www.sas.com>

John Leveille  
d-Wise Technologies, Inc.  
3115 Belspring Ln  
Raleigh NC 27603  
Work Phone: 1-888-563-0931  
Fax: 1-888-563-0931  
E-mail: [jleveille@d-wise.com](mailto:jleveille@d-wise.com)  
Web: <http://www.d-wise.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.