

Best Practices for Working with CDISC Metadata in the SAS® Data Integration Server

Michael Kilhullen, SAS Institute, Cary, NC

ABSTRACT

Over the past few years, SAS has demonstrated how the SAS® Metadata Server can be used to implement and manage CDISC metadata and facilitate a metadata driven approach to standardizing clinical data. In this paper, we examine best practices for using SAS® Data Integration Studio to execute and manage key CDISC concepts such as controlled terminology, value level metadata, normalization of data, importing and exporting XML documents, and producing the CRT-DDS. Within the context of these topics, we will also examine considerations for setting up and managing study metadata, writing efficient transformation processes, leveraging metadata to answer key business questions, and effective use of change management.

INTRODUCTION

This paper will present best practices as a series of topics that contain basic information, considerations, and real world examples from SAS pharmaceutical solution developers and consultants, and through customer feedback. The topics presented are based on a common understanding about how the metadata is organized and used for a typical clinical data integration project. This does not imply that this is how you must implement a metadata server. When considering business objectives and workflow, we sometimes find that some of these recommendations are less helpful. For example, customers who start with data that is already standardized, or customers who only have a handful of SAS programmers, will look at these recommendations from a different perspective. What is more important is that you consider the reasoning behind these recommendations, decide what is best for your business processes, and then standardize the way you intend to use the metadata server. For example, the Clinical Data Integration Service provided by SAS Professional Services and Delivery is built on top of the “clinical metadata framework”. The framework, which implements and leverages the best practices presented in this paper, defines how clinical data and standards are represented in the metadata, how consultants can more easily access the metadata to build customizations, plug-ins and reports, and how metadata as a whole can satisfy common business needs such as change management, archiving, and integration with other solutions such as SAS® Drug Development.

ORGANIZE YOUR METADATA

Metadata describes your physical world in a way that makes sense for how users will interact with it to produce deliverables. It does not require you change how or where you store your physical data and can leverage your existing SAS environments and programs. The metadata server only stores enough information about where everything is, what it looks like, who can access it, etc., to make it easier for you to focus on what you need to do rather than where you need to go to do it. In order to efficiently integrate with business workflow and provide metadata capabilities across your organization, the metadata server maintains one or more metadata repositories.

A metadata repository is simply a collection of metadata. All metadata servers contain at least one repository- Foundation. This repository contains mostly IT level metadata, such as server configurations, user and group information, standard metadata templates, etc. As users begin to interact with the metadata server, they can optionally create other repositories. The two types of repositories supported by the SAS Metadata Server are Custom and Project repositories.

Custom repositories leverage all the information defined in Foundation, but allow you to keep your business level metadata physically separate and distinct. From an administrative perspective, it means that you can act on the repository independently to take advantage of Metadata Server capabilities such as replication, promotion, and backups. From a user perspective, it provides more focus. When you work with a custom repository, you only see content, libraries, and folders that are meaningful to that repository.

Project repositories are used when working under change management. They are typically assigned to an individual, and stored securely on the metadata server. As a user checks-out objects from a custom repository, the metadata is copied into the project repository. There, metadata can be added, edited, and/or deleted. Changes made by the user are not applied to the custom repository until the contents of the project repository are checked in.

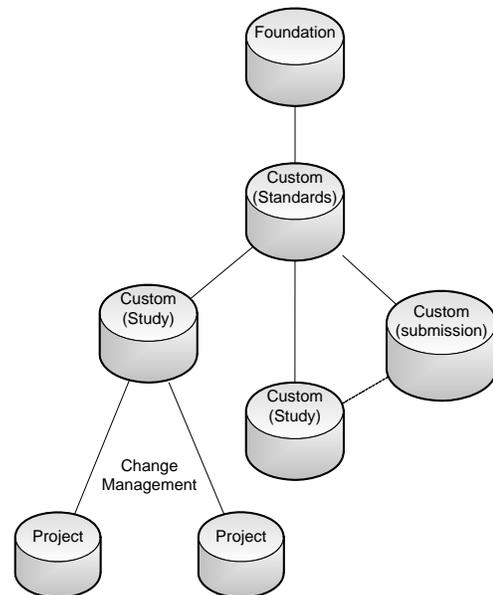


Figure 1-Typical Repository Configuration

CREATE A STANDARDS REPOSITORY

When working with clinical data in SAS Data Integration Studio, processes focus on creating well defined, standardized data. While this paper focuses on CDISC standards, specifically the Study Data Tabulation Model (SDTM), the concepts discussed can be applied to implementing any standard model. Most companies have a team assigned to managing the definition and deployment of standards. In addition to being able to easily define and manage the data and metadata associated with the standards, they would also benefit from centralized deployment, ongoing analysis of the use of the standards, and simplified version management and upgrade processes. These capabilities are available in SAS Data Integration Studio and are the primary considerations that lead to defining a single repository that focuses on standards (Figure 1).

Benefits of a standards repository:

- complements the responsibilities and workflow requirements for a standards management group
- easily deploy and manage standards
- track changes to standards using metadata server change management
- develop and deploy centralized libraries of processes, transformations, and programs that complement the standards
- easily propagate standards metadata through the promotion/replication processes provided by the SAS Metadata Server

CREATE A REPOSITORY FOR EACH STUDY

In clinical data integration, the primary workflow is built around a study, or trial. At this level, we find many common business rules concerning permissions, snapshots, archiving, database locks, etc. First and foremost, we want to ensure that the metadata can complement these processes since SAS Data Integration Server is metadata driven- if changes are needed, they are made in the metadata, not in a program. From a usability perspective, we want to make using SAS Data Integration Studio more efficient. When a user works on a study, the objects they see, the items available in selection lists, and where the things they create should be stored, should be related to only that study and automatic (Figure 1).

Benefits of a study centric repository:

- more easily archive or snapshot metadata
- change repository state to off-line or read-only when a study is finished
- more direct control over study metadata through repository level access permissions
- Better focus when developing processes- users only see what is relevant to the study
- easily add and manage custom study level metadata such as protocol information
- override metadata to reflect study specific scenarios without impacting standards or other studies
- easily propagate study metadata through the promotion/replication processes provided by the SAS Metadata Server

CREATE SEPARATE REPOSITORIES FOR AGGREGATED ANALYSIS

After studies are done, there are many ways in which the data can be used to contribute to other business needs. Most common to the clinical data integration process are IND updates and submissions. But many companies are also beginning to use clinical data to conduct further analysis to help optimize trial design, monitor enrollment and workflow, perform signal detection, etc. In these cases, they may look to use SAS Data Integration Server to build a clinical data warehouse, or specialized data marts to facilitate cross-study and ad-hoc analysis.

In all of these cases, the view and use of the metadata changes, and may even apply to different users and/or organizations within your company. Once again, creating a new repository will help refocus the use of SAS Data Integration Studio based on what is being done and who needs to do it (Figure 1).

Benefits of using a new repository for aggregation:

- all the benefits noted for a study repository are applicable to the topic of the aggregation
- allows read-only access to study repositories, thus maintaining the study locked state
- allows post study adjustments that will not impact study metadata. This would be the case if the repository represents a submission for studies conducted over a several year period- all studies might not be the same SDTM version.
- subset patient populations

WORK UNDER CHANGE MANAGEMENT

SAS Data Integration Studio fully supports metadata change management. In this configuration, users cannot directly add/edit/delete metadata content in a repository. Instead, they must “check-out” the metadata to their project repository. Once there, they can freely develop metadata content. While this is taking place, the metadata that is checked-out is locked. No other users can checkout the same object. The fact that metadata is checked out is reflected in the SAS Data Integration Studio Interface. Once the user is done, the contents of their project repository are “checked-in”. SAS Data Integration Studio prompts the user for information about changes made, updates the metadata based on the changes, and generates change metadata for every object affected as a permanent record of change.

Benefits of working under change management:

- prevents duplication of effort
- users have a failsafe- checkout can be “undone” to restore metadata to its original state
- generates metadata centric audit trail that is easily accessible from the Data Integration Studio Interface

BUILDING PROCESSES

Data standardization is achieved in SAS Data Integration Studio by building processes. A process is a visual flow that contains a series of transformation steps that define metadata about how to convert source data to standardized SDTM domains. Processes are metadata driven- they help users make better decisions about how to convert data by presenting information in graphical user interfaces, performing automatic mappings and propagations when similar data is found, collecting documentation from the user about decisions made at each step of the process, and interactively managing the SDTM metadata added to the SAS Metadata Server. SAS Data Integration Studio generates SAS code based on the metadata collected and provides workflow around publishing the code to scheduling servers, web services, or as stored processes. A sample process is shown in Figure 2.

USE A COMMON LIBREF FOR SOURCE DATA IN STUDY REPOSITORIES

Each study will have a number of standard library objects created to represent source and target locations at minimum. When dealing with source locations containing SAS datasets such as extracts from clinical database management systems, or a SAS transport file received from a CRO, the data is often accompanied by a SAS format catalog. A library object in SAS Data Integration Studio does not currently allow you to “activate” a format catalog when the library is assigned. That is, the code generated is only a LIBNAME statement. Users must therefore assign the FMTSEARCH option manually. There are several ways to approach this, but the preferred way is to use a common LIBREF for source libraries and alter the SAS configuration to assign this library to the FMTSEARCH option at startup. After doing this, the assignment of the format catalog is automatically applied when SAS Data Integration Studio assigns the LIBREF.

To setup this approach, you must first standardize on the LIBREF that will be used for source data. In many of our implementations, we simply use RAWDATA. Next, the system administrator modifies the SAS configuration file to include the new LIBREF:

```
-FMTSEARCH =(RAWDATA WORK LIBRARY)
```

When the RAWDATA library statement is generated by SAS Data Integration Studio, the format catalog location in that library will be automatically applied. This solution does not work when more than one study is used as input into the same process. However, in the overall process flow of CDISC standards, by the time you are ready to aggregate, you should be using SDTM data that does not use SAS formats.

Benefits of a common source library reference:

- automatic access of format catalogs found with data
- less administrative overhead

BUILD PROCESSES WITH A DOMAIN FOCUS

The goal of domain focused processes is to have only one process defined for each primary domain. Secondary domains relevant to the primary domain should also be included. For example, if we build a process to convert our data to Demographics (DM), we will also include Supplemental Qualifiers (SUPPDM) and Subject Characteristics (SC) at minimum (Figure 2). Following this rule, each process will therefore have a minimum of two domains- a primary domain and its supplemental qualifier- and include all sources that contribute to the domain content.

The issue we are avoiding is determining how to load the data when it appears in several processes. If a domain was already loaded by one process, we must make sure we do not delete and/or overwrite data that already exists. Subsequently, we must also be cautious about the order in which the processes are executed. Rather than trouble users over figuring out how to load the data or where changes should be made, we establish a standard whereby domains are only loaded once during a study update.

By creating domain focused processes, you are immediately improving the usability and workflow of SAS Data Integration

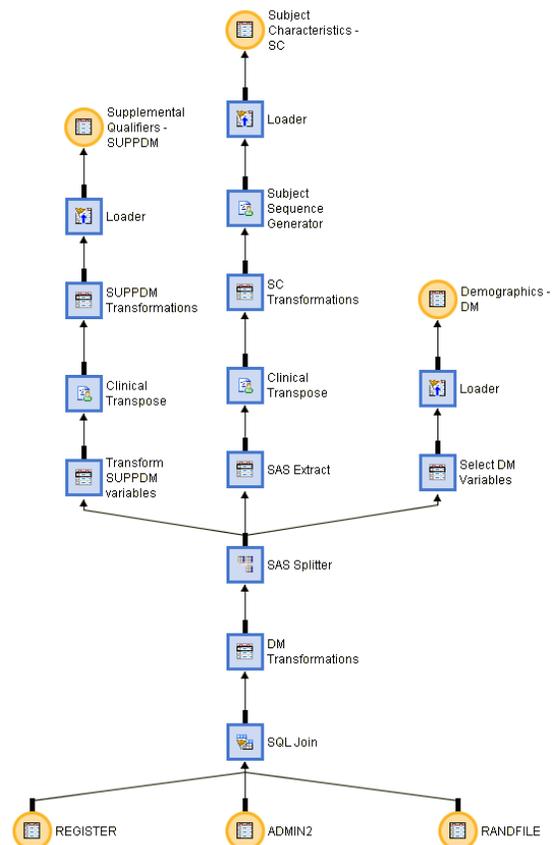


Figure 2- a Demographics(DM) Focused Process

Studio. Larger programming groups will delegate assignments based on domain. Therefore, the folder view in SAS Data Integration Studio will complement how they need to work. Furthermore, when working under change management, users will check out only what they really need for the domain minimizing cross process conflicts (as would be the case is one domain was loaded from several processes), improving performance (since they are not checking out gigantic processes), and creating more meaningful change record audits (which tend to reflect domains rather than program tweaks and source table changes).

Benefits of domain focused processes:

- maintains SDTM presence in the use of Data Integration Studio
- improves usability and focus while designing processes
- improves change management performance because the process size is more controlled
- results in more meaningful change audits since comments naturally tend to reflective the domain

AVOID CREATING DERIVED MAPPINGS IN TRANSFORMATION STEPS IMMEDIATELY AFTER A SOURCE TABLE

When data is mapped, each mapping is defined in the metadata as a FeatureMap with a role of either one-to-one or derived. If you subsequently remove a source table from the process, any FeatureMaps defined in the next step that use columns from the source table are also deleted. One-to-one mappings are easily recovered since SAS Data Integration Studio provides automapping and GUI based map definitions when you add a new table to the process. The expressions however, would need to be redefined. The same holds true for table loaders.

If you already defined derived mappings under these conditions, simply insert an Extract transformation after the source table before removing the table. By doing this, the FeatureMap becomes associated to the columns in the extract step rather than the actual dataset. The dataset can now be removed without losing the column mappings (Figure 3).

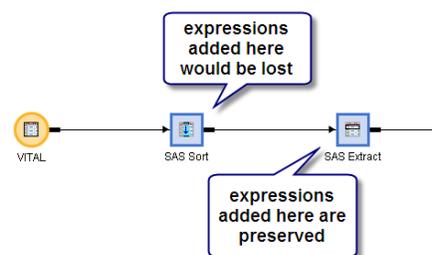


Figure 3- conditions for losing expressions

This technique can be used to create process “templates”. A process template is created from a running process.

The premise is to strip the process of inputs and outputs so that all the logic and transformations remains captured in the process flow. Users can then copy the process template into a new study, then add and map a new data source (Figure 4).



Figure 4- process in figure 3 converted to a template

Benefits of preserving mapping expressions:

- less work when editing processes
- promotes creating process templates
- improved transformation metadata integrity

TURN AUTOMAP OFF PRIOR TO CHANGING SOURCE AND/OR TARGET TABLES IN A PROCESS

Whenever you add or change data sources and targets on a process, SAS Data Integration Studio will perform two useful functions to make quick work of building the process: quick map and quick propagate. Both functions apply to the mapping at each step. With quick map, SAS Data Integration Studio will automatically map columns that are the same in the source and target data. With quick propagate, SAS Data Integration Studio allows you to add source columns to the target table, and then automatically map them. Both of these functions are collectively controlled by the automap option that is available on every transformation step.

While automap makes quick work of creating a process, it often gets in the way when you need to add or remove data to an existing process. When a new table is added, SAS Data Integration Studio will automatically map and propagate variables throughout the process. If you took the time to validate your process, this will effectively invalidate it. Instead, prior to adding a new table, right click on the step immediately after where you will add the table and deselect automap (Figure 5). Then add your table. This prevents a bunch of columns from appearing throughout your process. However, you need to manually map the step for which automap was turned off. Luckily, you can choose to automap on the mapping tab.

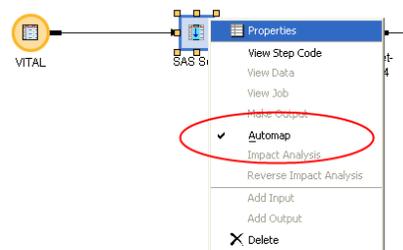


Figure 5 - automap option for a transformation step

Benefits of overriding automap technique:

- prevents columns from propagating throughout a finished process
- promotes metadata reusability

LEVERAGE TRANSPOSE TRANSFORMATIONS TO NORMALIZE DATA AND COLLECT VALUE LEVEL METADATA

A common requirement for transforming data to CDISC standards is to transpose, normalize, flip, or pivot the data. Arguably, PROC TRANSPOSE is the primary tool to do this. From a CDISC perspective however, once data is transposed, the meaning behind the original values that contributed is lost- the resulting data only has a variable name and some data converted to character values.

When we perform a transpose in SAS Data Integration Studio, we get the same result. However, by using the

TRANSPOSE transformation, we are capturing metadata about the variables that contributed to the final data and can retrieve any information about what data looked like before it was transposed. By combining the output of a PROC TRANSPOSE and the metadata collected about how it was transposed, we can generate the level of documentation necessary to provide value level metadata for the CDISC models. How users can extract this information in general and use it is discussed later in this paper.

Benefits of the Transpose transformation step:

- transposes data
- collects metadata about what was transposed to contribute to value level metadata documentation needs

CONTROLLED TERMINOLOGY

Controlled terminology is implemented as physical tables, or a link to a database in the metadata server. The primary advantage of this approach is that the data content can be leveraged within a SAS Data Integration Studio process to contribute data to the final domains and to automate management of the controlled terminology tables (Figure 6).

DEPLOY CONTROLLED TERMINOLOGY TABLES IN THE STANDARDS REPOSITORY

Most organizations will implement centralized controlled terminology, either as a relational database in Oracle or simple SAS tables. Considering the metadata server configuration discussed previously, the standards repository is naturally the best place to store metadata about the tables to facilitate use, management, and deployment within the SAS Metadata Server. Some companies extend this configuration by copying the controlled terminology tables into a study folder, and then register them as part of the study repository. This way, study specific modifications can be made and the controlled terminology tables can be more easily archived with the completed study or snapshot. Regardless, the “master” controlled terminology tables are still managed centrally, and customizations are added to SAS Data Integration Studio to automate copying the tables to a study area.

Benefits of deploying controlled terminology in the standards repository:

- easy deployment
- maintains relationships to the standard models
- directly accessible in processes

USE THE LOOKUP TRANSFORMATION TO ACCESS CONTROLLED TERMINOLOGY

The Lookup transformation allows you to load a table with columns taken from another table. The properties of the Lookup transformation define how the data is matched, what values get mapped through, and what to do if values are not found in the lookup table (Figure 7). This is pretty straight forward when it comes to controlled terminology. Any process can tap into the content of a controlled terminology dataset to add and standardize content in the final domain. The added value here is the error handling.

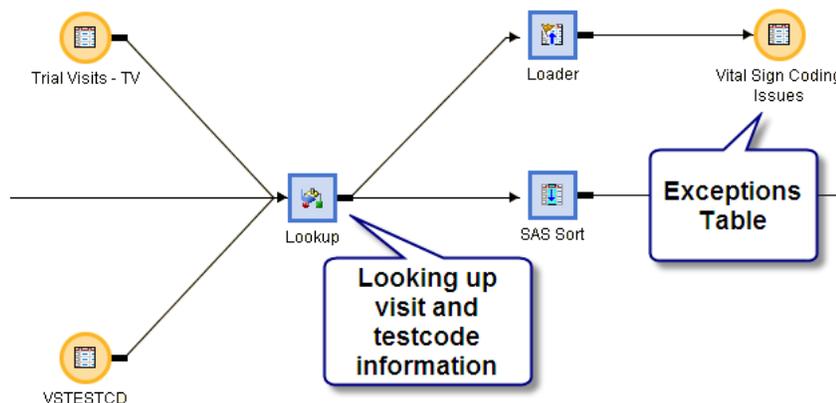


Figure 7- Example of lookup Transformation

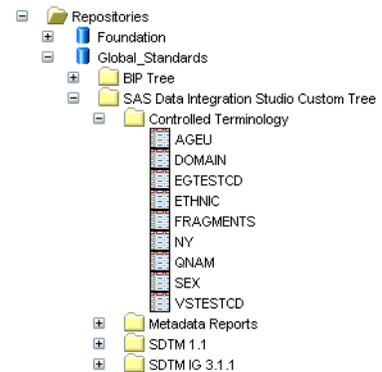


Figure 6 - Controlled Terminology in Standards repository

Suppose that you receive daily incremental data for a study. One day, a value comes in for LBTESTCD that is not defined in the controlled terminology table. Rather than running quality assurance programs after data is loaded, the lookup transformation performs checks automatically when processes are executed. If it finds any issues, information about the record causing the conflict (and any other info you choose to add) is posted to an exception table.

When a study is defined in SAS Data Integration Studio, we also define a standard “exceptions” folder that contains one or more exceptions tables used by Lookup transformations regardless of what process they are defined in. Therefore, exception tables have a metadata identity, and can be used by other processes, pushed out to a portal, and/or interactively browsed by a user. For example, after a study is done updating, an exception notifier process can be run that scans these tables and emails a list of discrepancies to the team responsible for managing controlled terminology content.

Benefits of using the lookup transformation:

- integrated validation and consistency checks whenever a controlled terminology table is used
- better error handling, including missing value and missing table support
- persisted issue storage

LEVERAGING THE SAS METADATA SERVER

Most programmers who evaluate SAS Data Integration Studio tend to focus only on the ability to produce standard data. In this case, they are missing the value that the metadata server brings to the clinical data integration process. Even at a higher level, the Business Intelligence Architecture, which is driven by the metadata server, provides a framework for improved workflow, management, communication, data integrity, data quality, reusability, etc.

Metadata is becoming an integral part of our data stream as well. In the past, a simple PROC CONTENTS could generate table descriptions to aide in standardization or macro call automation. But today, we have specific models to deliver which define metadata beyond just the data structure. There is also a much greater emphasis on data content standardization and quality through the implementation of controlled terminology. In short, we need to know a lot more about what our programs are doing and be able to analyze and document how we interact with clinical data.

When working with SAS Data Integration Studio, much of this value is gained just by using the product. However, it is designed to accommodate the data standardization needs of many industries and may not have all the content expected for working with pharmaceutical data. However, the SAS Business Intelligence Architecture is designed to be extensible, and provides tools to allow SAS programmers to easily customize and enhance their SAS Data Integration Studio user experience, or for Java programmers to add completely new functionality, develop code generators, and embed tighter connections with 3rd party systems and tools.

USE PROCESS TRANSFORMATIONS OR CUSTOM TRANSFORMATIONS BEFORE RESORTING TO USER WRITTEN CODE

One of the goals of using SAS Data Integration Studio is to gain a detailed level of understanding about what you are doing to your data. This understanding is initially gained just in designing a process. By linking tables through transformations and loaders, you are defining data lineage- how you got to a standardized dataset. In addition, each transformation step is collecting more detailed metadata about how data is acted upon when it passes through the step. SAS provides several tools to navigate the detailed metadata of a process, some of which are directly available in SAS Data Integration Studio. For example, the Impact Analysis Tool allows you to trace a table or column throughout the metadata server, describing not only how the data was created, but how that data was consumed by other processes and tools such as SAS® Web Report Studio or SAS® Enterprise Guide®.

When you are building your processes, try to use transformations from the Process Library as much as possible. If SAS Data Integration Studio does not provide one that meets your needs, consider creating a custom transformation. As a last resort, add a User Written Code transformation. This hierarchy is proportional to the amount of detailed metadata that you are collecting. Transformations provided in the Process Library generate a rich amount of metadata. Defining a custom transformation allows you to wrap metadata around the code you wish to execute. Using User Written Code gets the job done, but is effectively a “black box” approach when it comes to understanding what is happening in the transformation step

Benefits of Using Process Library and custom transformation:

- more meaningful metadata is captured
- adds value to metadata exploration tools
- leverage graphical user interfaces to collect metadata

MIGRATE CODE TO CUSTOM TRANSFORMATIONS AS MUCH AS POSSIBLE

A large part of a clinical data integration service is deciding how to best leverage existing programs and macros. Deciding whether a program should be migrated into a transformation step or retained in a macro library accessible to the workspace server depends largely on what the program does, how it is executed, and how it can be best leveraged by SAS Data Integration Studio. I want to be clear- SAS Data Integration Studio should complement your existing workflow and programs, not replace it. In some cases, you will find that a macro that creates transport files, compares data structures, or calculates a value can be done easily in SAS Data Integration Studio. But the standard SAS server configuration does allow you to more easily deploy standard programs, macros, and formats for use in any SAS client.

The advantage of migrating some of your programs to user written transformations is that you can leverage SAS Data Integration Studio to provide a graphical front end to your program, which in turn is promoting metadata collection about how your program is being used. For example, you might have a macro that accepts two dates and calculates the duration. In a programming approach, you would have to read code to find out what variables were passed in and which variable holds the result. Alternatively, you can add more overhead to your macros to publish this information when the macro is executed. In SAS Data Integration Studio, users can select the columns from a graphical interface that stores the selections in metadata. This is a design time process- there is no need to passively generate documentation about selections when the program is run.

Migrating your code to custom transformations will also allow you to leverage the change management workflow available in SAS Data Integration Studio to develop, manage, and analyze the use of your programs. When you create a custom transformation, you are creating a metadata object. Beginning in SAS Data Integration Studio version 3.4, the object is available in your Custom tree in addition to the Process Library. This enables you to enforce change management on the maintenance of your custom transformation. Users must now check-out the transformation in order to make changes, and when checked back in you get change audit objects added to the metadata.

In addition to change management, you also gain two useful capabilities in managing the use of your transformation. First, changes made to your user transformation are propagated to all processes that use it. When the user opens the process, SAS Data Integration Studio will notify the user about the change and prompt them to update the transformation. This ensures that a change to the custom transformation does not break deployed processes. Second, custom transformations are impact analysis enabled. This means that with a single mouse-click, a developer can see every process known to the metadata server that uses the custom transformation.

Benefits of using custom transformations:

- manage code under change management
- easily propagate change
- leverage impact analysis to monitor use of programs

EXTRACT METADATA FOR VALIDATION AND REPORTING

In SAS Data Integration Server implementations, metadata drives the process of standardizing your data. In the end, you create SDTM data. In the process, you collect a wealth of new metadata that can be leveraged for validation and reporting. Metadata stored in the SAS Metadata Server can be accessed through the SAS® Open Metadata Interface. This interface enables users to read and write metadata from SAS, Java™, and .NET™ applications.

One of the more challenging aspects of working with metadata, is understanding how to request and process data through the API. In order to do this effectively, you must first understand how metadata is stored in the SAS Metadata Model. Then, you must understand how to structure XML requests and parse the information returned. These topics are beyond the scope of this paper, but SAS provides a fair amount of documentation about the SAS Open Metadata Interface and how to define and use XML maps on <http://support.sas.com>. For the everyday SAS programmer, most common metadata interactions can be achieved through metadata DATA step functions or simple XML calls through PROC METADATA, both available in Base SAS.

In the domain centric world of the SDTM, most metadata queries begin with an extract of table and column metadata. To retrieve metadata for a table, you write a XMLSELECT statement to filter results for a table, and then define the associations and attributes that you would like to include in the result set in a TEMPLATE definition. An example metadata query is shown in Listing 1. Executing this metadata request would return results similar to Listing 2. Once we have the output XML, we can define and apply a XML Map to convert it to SAS datasets that will serve as input to our metadata reporting needs.

Benefits of extracting metadata:

- leverage all the information that is already collected while using SAS products
- collect and share metadata with other CDISC models
- add reports and specialized validation not included in SAS Data Integration Studio

IMPLEMENT METADATA REPORTS IN DATA INTEGRATION STUDIO PROCESSES

After you created your metadata extraction programs, the ideal place to execute them is in a SAS Data Integration Studio Process. This will allow you to submit reports in batch or schedule them for regular execution as part of your data standardization process. To make this a manageable and repeatable process, implement the extraction programs as custom transformations. In doing this, metadata extraction becomes a simple drag and drop operation for your users.

In a SAS clinical data integration implementation, metadata focused custom transformations are categorized into two types: Metadata Extractors and Metadata Reporters (Figure 8). The idea behind this approach is to minimize the number of extractions in order to keep metadata server load to a minimum, and then generate as many reports as possible from each extraction. In addition, developers gain all the advantages previously discussed to help with the deployment and management of the programs.

```

<GetMetadataObjects>
<Type>PhysicalTable</Type>
<Options>
<XMLSelect
  search="PhysicalTable[@Name = 'Adverse Events - AE']/TablePackage/SASLibrary[@Libref
    = 'S_NIC1']"/>
<Templates>
<PhysicalTable Id="" Name="" Desc="" SASTableName="" MetadataCreated=""
  MetadataUpdated="">
<Changes/>
<Columns/>
<Extensions/>
</PhysicalTable>
<Column ID="" Name="" Desc="" SASColumnLength="" SASColumnType="" SASColumnName=""
  SASFormat="" SASInformat="" MetadataCreated="" MetadataUpdated="">
<Extensions/>
</Column>
<Extension ID="" Name="" Desc="" Value=""/>
</Templates>
</Options>
</GetMetadataObjects>

```

Listing 1- XML Metadata Request

```

<Objects>
<PhysicalTable Id="A5YJEXUX.AD000001" Name="Adverse Events - AE" Desc=""
  SASTableName="AE" MetadataCreated="09Mar2007:22:30:52"
  MetadataUpdated="09Mar2007:22:30:52">
<Columns>
<Column Id="A5YJEXUX.AE000001" Name="STUDYID" Desc="Study Identifier"
  SASColumnLength="20" SASColumnType="C" SASColumnName="STUDYID" SASFormat=""
  SASInformat="" MetadataCreated="09Mar2007:22:30:52" MetadataUpdated="09Mar2007:22:31:34">
<Extensions>
<Extension Id="A5YJEXUX.AF000001" Name="TERM" Desc="Controlled Term or Format" Value=""/>
<Extension Id="A5YJEXUX.AF000002" Name="ORIGIN" Desc="Origin" Value="CRF"/>
<Extension Id="A5YJEXUX.AF000003" Name="ROLE" Desc="Role" Value="Identifier"/>
<Extension Id="A5YJEXUX.AF000004" Name="CORE" Desc="Core" Value="Req"/>
<Extension Id="A5YJEXUX.AF000005" Name="References" Desc="References" Value="SDTM
  2.2.4"/>
</Extensions>
</Column>
<Changes>
<Change Id="A5YJEXUX.B6000001"/>
</Changes>
<Extensions>
<Extension Id="A5YJEXUX.AF00005G" Name="DOMAIN" Desc="Domain Abbreviation" Value="AE"/>
<Extension Id="A5YJEXUX.AF00005H" Name="CLASS" Desc="Domain Type" Value="Events"/>
<Extension Id="A5YJEXUX.AF00005I" Name="XPORT" Desc="Transport Filename" Value="AE.xpt"/>
<Extension Id="A5YJEXUX.AF00005J" Name="STRUCTURE" Desc="Domain Structure" Value="One
  record per event per subject"/>
<Extension Id="A5YJEXUX.AF00005K" Name="PURPOSE" Desc="Purpose" Value="Tabulation"/>
<Extension Id="A5YJEXUX.AF00005M" Name="VERSION" Desc="CDISC Version Number"
  Value="3.1.1"/>
<Extension Id="A5YJEXUX.AF00005N" Name="DATE" Desc="Release Date" Value="August 26,
  2005"/>
</Extensions>
</PhysicalTable>
</Objects>

```

Listing 2- XML Metadata Results

After custom transformations are defined, simply create processes that use them. An example process that leverages metadata extractors and reporters is shown in Figure 10. This process will extract data and run several reports for all data found in the SDTM library. Before you run the process, optional macros that define your metadata server connection must be enabled. To do this, open the process properties, click the options tab, then change “enable optional macro variables” to yes (Figure 9). When SAS Data Integration Studio generates code, it will now include macro parameters that define the metadata server connection based on where the process is executing.

Benefits of implementing metadata extraction as a process:

- drag and drop metadata reporting
- schedule reports or run in batch
- add as part of the data load to automatically generate documents when data is loaded

USE PROC CDISC TO WORK WITH ODM AND CRT-DDS

PROC CDISC enables you to read and write ODM xml, verify SDTM domain conformance, and generate CRT-DDS (define.xml). It is available in Base SAS and can therefore be used in any SAS Data Integration process. While full support for the XML libname engine already exists in SAS Data Integration Studio, PROC CDISC provides a nice packaged solution with features that can be used at multiple points within a process.

While PROC CDISC provides XML generation, it does not take care of additional needs such as collecting metadata, creating transport files or extracting documents and data to a publishing area.

However, when used in combination with SAS Data Integration Server, these capabilities can be provided as part of a CRT-DDS publishing process. While determining how and where metadata is stored in a standard clinical data integration solution, SAS consultants also considered the needs of the CRT-DDS. Because of this, metadata extraction and reporting techniques discussed in this paper also include define.xml metadata. Therefore, we can automate and package the CRT-DDS in a process flow using a metadata reporter transformation based on the use of PROC CDISC (Figure 10).

Benefits of using PROC CDISC in SAS Data Integration Studio:

- XML gain without the pain
- leverage metadata that you already collected
- add other packaging requirements as part of a process flow

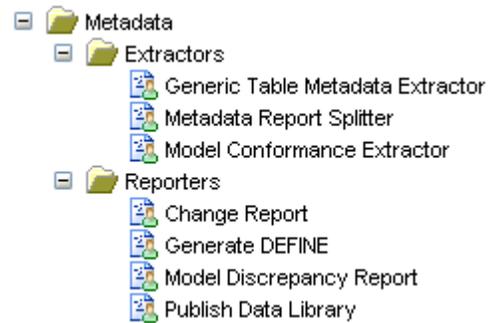


Figure 8- Metadata Folder in Transformation Library

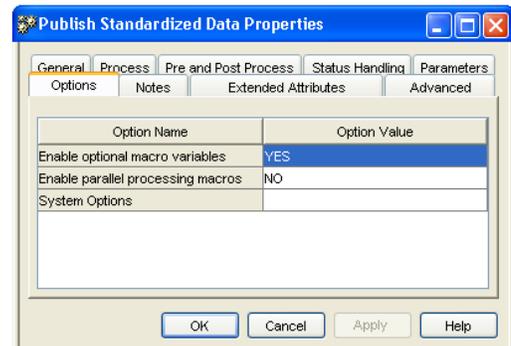


Figure 9- Enable Optional Macro Variables

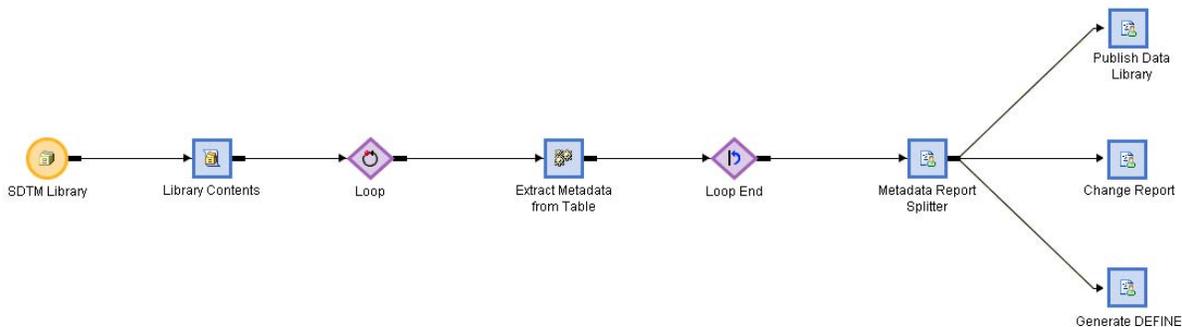


Figure 10- flow implementing metadata extractors and reporters

CONCLUSION

This paper presented very common best practices to consider if you are using SAS Data Integration Studio to convert your data to SDTM format. While the usefulness of these best practices depends greatly on how your business goals can be best obtained while using the software, this will hopefully give you some ideas about how others are using and customizing SAS Data Integration Studio to improve the SDTM standardization process. As you begin to travel down this road, remember that starting with a common understanding about how your organization will collect, manage, and use metadata is the key to implementing an efficient, reusable, and scalable process. Whether you are focused on SDTM or your own custom standard:

1. Understand the metadata you need to capture
2. Perform a mapping exercise against the SAS Metadata model
3. Decide how to implement custom metadata
4. Formalize your decisions by creating a mapping plan
5. Have fun making SAS Data Integration Studio your own

RECOMMENDED READING

The Value of Integrated Metadata: SAS® Open Metadata Architecture
A SAS Whitepaper available at <http://www.sas.com/whitepapers/>

The SAS Enterprise Intelligence Platform: SAS Data Integration
A SAS Whitepaper available at <http://www.sas.com/whitepapers/>

Kilhullen, Michael. "Implementing CDISC Data Models in the SAS® Metadata Server"
Proceedings for the Pharmaceutical Industry SAS® Users Group Conference, 2006 & 2007

ETL Performance Tuning Tips
A SAS Technical Paper available at <http://support.sas.com/documentation/whitepaper/technical>

CONTACT INFORMATION

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

Michael Kilhullen
SAS Institute Inc.
62 Edison Road
Stewartsville, NJ 08886
(908) 760-6528
michael.kilhullen@sas.com
www.sas.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.