Paper TS10

Technical application of SAS ETL Studio in the clinical reporting process

Dave Smith, SAS, Marlow, UK

ABSTRACT

The clinical reporting process has become an increasing area of focus as pharmaceutical companies seek to improve its efficiency. The main approach taken up to now has been to enforce standards of metadata and process to re-use as much as possible from earlier trials. This approach has largely failed to deliver the expected gains because scientific research, clinical investigators and patients consistently refuse to adhere to standards. In a growing number of companies, SAS® ETL Studio is being seen as the answer to this problem. In this presentation we will examine how SAS ETL Studio can be aligned with the existing clinical process, how an implementation team can use metadata scripting and extended attributes to automate that process, and how standards such as CDISC SDTM can be implemented.

INTRODUCTION

If you explained the clinical reporting process to a Martian they would have difficulty understanding why it takes so long - after all, the process consists of tabulating and graphing relatively small amounts of data in standardised formats. It's only when the detail of the data is examined, and the variability in the metadata is seen that the true complexity of the process becomes clear. The Martian might, reasonably, ask why the metadata can't be standardised at source to simplify the entire process, but the fact that the source data refers to real patients being managed by investigators doing scientific research precludes this rigid standardisation – after all, biological entities are inherently variable, and ethics prevents investigators proceeding exactly to the protocol where it is against the patients best interests! Given that clinical studies need to be combined for safety and efficacy summaries and that regulators are going to review them, it is also clear that standardisation of the output at the end of the trial is essential. In the following sections I will attempt to show how SAS technologies can solve the problem of having a variable process at one end and a rigid set of standards at the other.

WHY MANAGING CLINICAL PROGRAMMERS IS LIKE HERDING CATS

The task facing clinical programmers is a complex one; take a set of data that changes regularly and create a set of programs that produce a standard set of outputs for regulatory submission within a compliant framework. They are likely to be part of a global team and may not be in the same geography as their statistician. There is always too much work to do, and their programs are on the critical path for a drug submission.

Clinical programmers are usually focused and creative individuals – otherwise they would not be capable of doing the job successfully – and this leads to a desire to do things as efficiently as possible. When a typical SAS programmer programs efficiently, the following characteristics emerge

- Lots of Macro
- Few comments
- Loads of special "tricks" exploiting SAS' more obscure features that increase efficiency at the expense of readability
- A definitive programming style for each programmer

The more senior and experienced a programmer becomes, the more difficult their code becomes to understand by their more junior counterparts (and their managers), and validation of their code takes longer and is a more uncertain process. There is also a considerable overhead associated with maintaining and adapting code, and international collaboration becomes very difficult.

From the perspective of a manager this is a complete nightmare – where is the capacity to produce the submission code to the agreed deadlines? It's not much fun for the programmer either, because the majority of their time is spent either doing boring repetitive tasks or going through someone else's code to work out why it doesn't work or where it needs to be changed.

SO WHAT'S THE ANSWER?

IT'S THE METADATA STUPID

To paraphrase Bill Clinton's strategist James Carville, who used the phrase "It's the economy stupid" to focus the successful 1992 Democratic election campaign, the most important consideration in the management of clinical trials processes is metadata. This needs to be the one consistent thread running through a set of clinical data – if you can understand the source of each column in your submission table and which transformations have been applied, you can be sure that the data is accurate and complete. By the same token, if you know exactly where each item in your source database has been used in the submission dossier, when a data item has to change you are able to manage changes to the affected programs and minimise the associated validation effort.

A clear indication of the process flow that has been followed through the program structure leads to less need for written documentation and easier maintenance. Metadata can also be used to assess adherence to global metadata standards, and ensure that the combination of analysis datasets for integrated safety and efficacy summaries is possible. Finally, metadata can be used to add evidence to regulators that the clinical development process is in control.

SOUNDS GREAT – WHERE DO I BUY ONE?

A part of the answer could be provided by using SAS ETL Studio. ETL stands for Extract, Transform and Load, which is a good description of the process of creating analysis-ready datasets. The data (and metadata) are extracted from the source CDMS or EDC system and nearly always need to be transformed into a shape that is suitable for analysis. The data also need to be loaded into an area where clinical programmers and statisticians can create tabular and graphical analyses in a controlled fashion. The load component also prevents the current data being overwritten by erroneous data if there has been an error in an earlier part of the process. SAS ETL Studio is a piece of client software that enables clinical programmers to perform this task.

THE ETL STUDIO PROCESS

The ETL Studio process differs from traditional SAS programming in that the source and target metadata are formally defined before the process is created. This adds rigour, assuring that correct standards are being applied to the target definition, and allows the majority of the simple one-to-one mappings to be created without further intervention by the programmer, thus reducing errors and increasing consistency.

Source definition

The source tables are defined in the metadata environment using an import wizard. The wizard will read the metadata directly from the source system, ensuring that the metadata impact analysis starts inside the source system; this also reduces assumptions about the structure of the input data sources.

Target definition

The metadata for the target is defined through a wizard. This can be imported from a variety of sources, including modelling tools (e.g. Erwin), existing SAS tables and metadata items, or external tables. It is likely that this structure will either be based upon the CDISC SDTM model or an existing company standard.

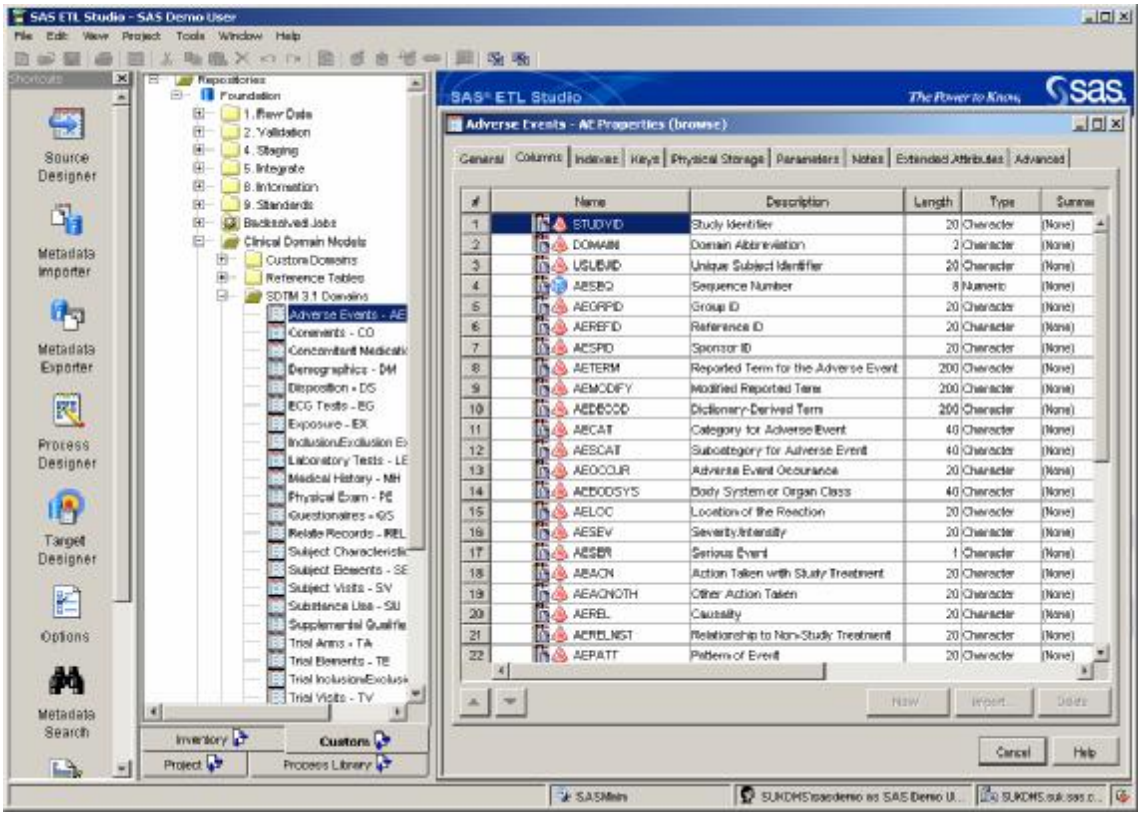


Figure 1 : The SDTM AE Domain defined in the metadata.

Process definition

The sources and targets are dragged and dropped onto the process flow diagram along with transformation nodes, such as SQL join, Sort, Transpose and Append. The metadata for each of these nodes is defined through the GUI, and the code is generated in the background. The process can be tested and then scheduled for batch running when it has been approved. Where the existing nodes do not accomplish any complex programming tasks that might be required, a user transformation can be employed which permits the programmer to write SAS code for a particular section of the process. In this case the metadata items will need to be linked up manually for the node; this is a small overhead compared to the benefit of an end-to end impact analysis. If this is a task that is commonly deployed the programmer can create a node that can be shared easily with other users; more advanced programmers can also build a custom GUI using Java.

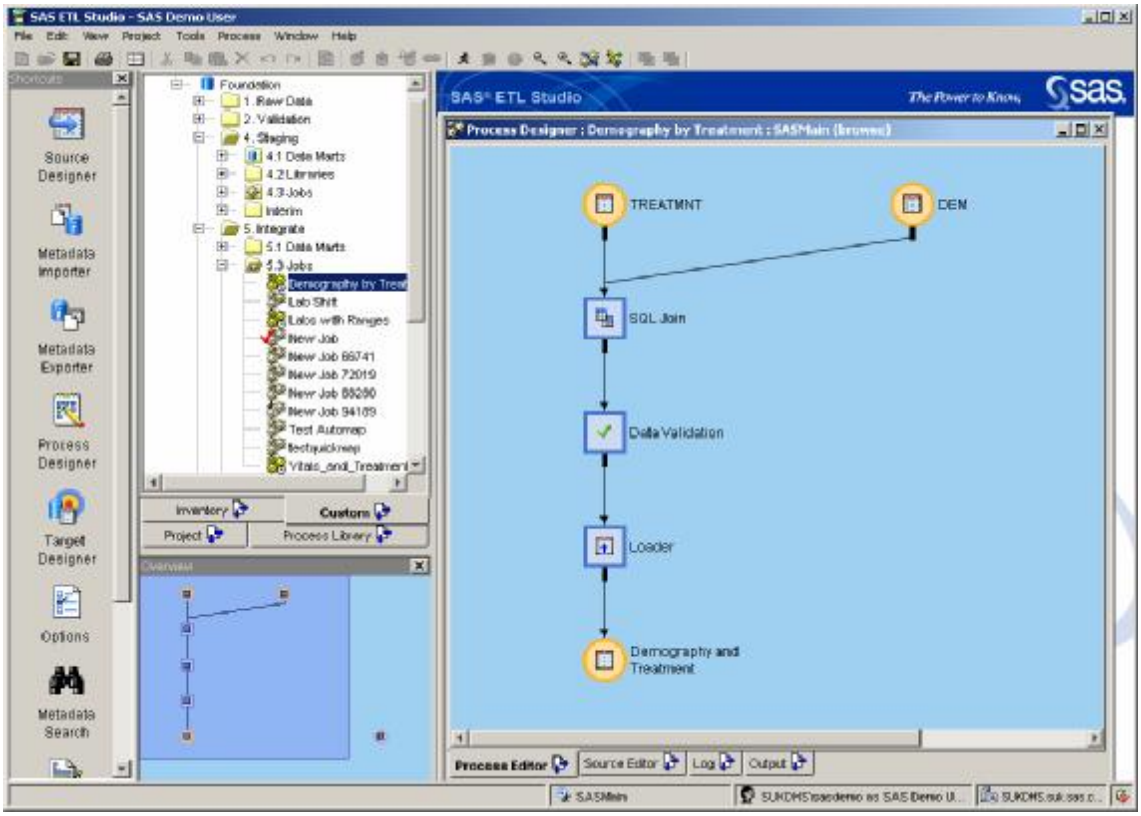


Figure 2 : A simple process to combine two datasets

Deployment

Once the process has been created, the process can be scheduled as a job within the SAS 9 environment. This allows the management of job dependencies, automatic and conditional scheduling, and the creation of alerts such as e-mails on completion or failure.

Maintenance and Updates

When changes are required to the process, either to the source data or the targets, the first step is to identify the impact of the planned changes. This is expedited by the impact analysis feature which works both forwards (demonstrate where a column or table is being used) and backwards (show the inputs that have been used to create a column or table).

Some examples are shown in figures 3 and 4; in figure 3 the uses of the LAB_RANGE table are shown with its mapping to the Labs with Ranges table and beyond. In Figure 4 the uses of the HI_NORM column are shown, including input into the LOMEDHI column, which is marked as Derived.

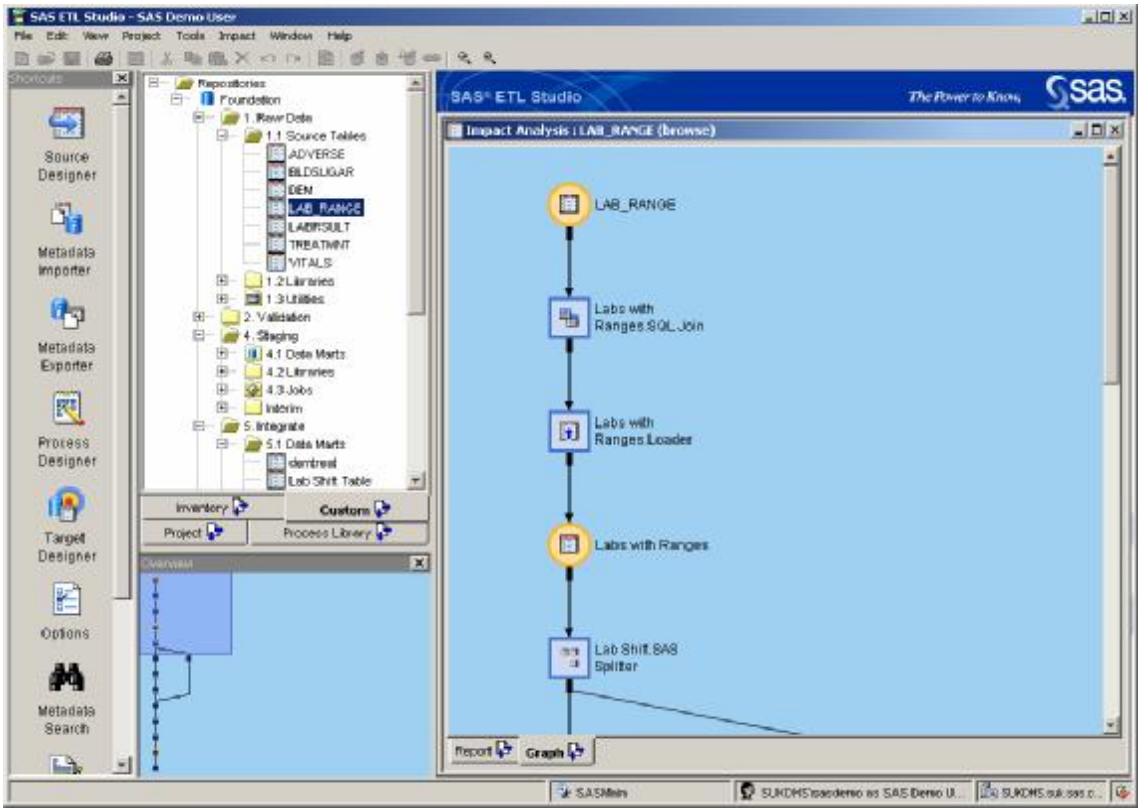


Figure 3 : Impact analysis on a dataset

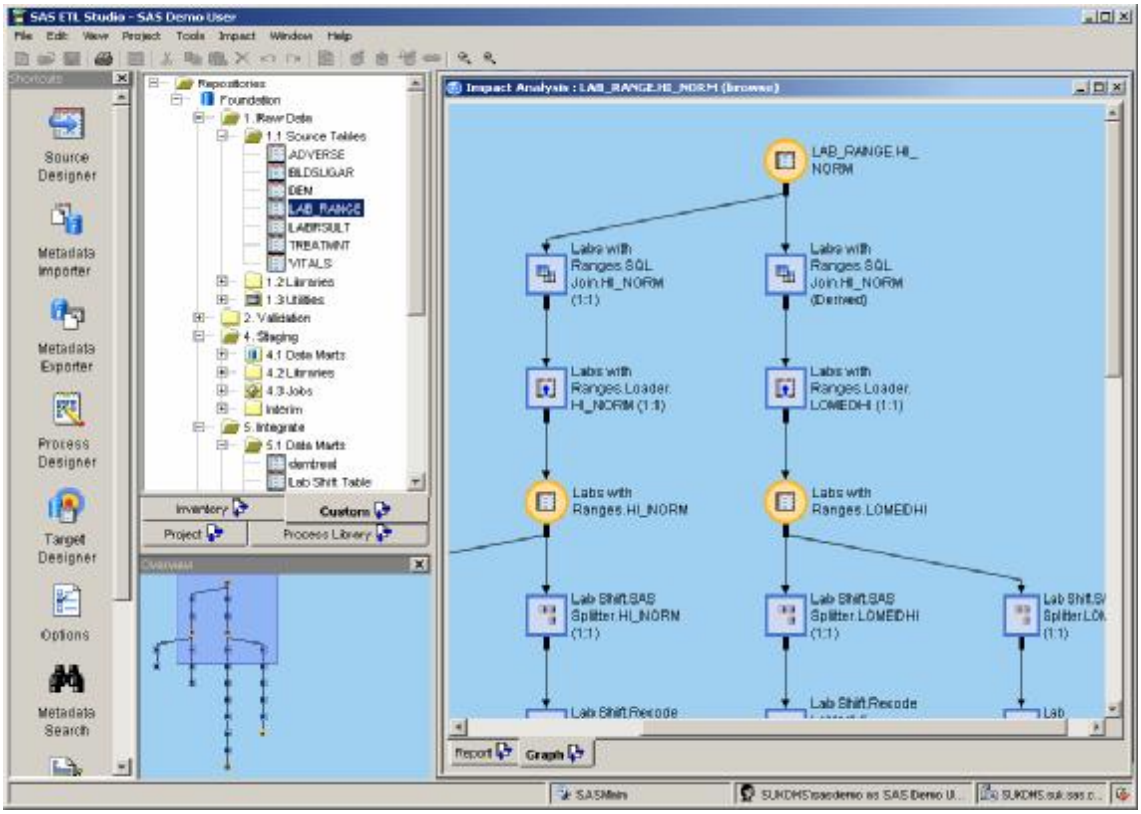


Figure 4 : Impact analysis on a column

Once the impact of changes has been determined, and a decision to proceed has been made, the mappings can be updated and, crucially, only those parts of the process that are affected need to be re-validated.

WHAT'S THE COST?

This section examines the costs of embarking on an implementation of a clinical programming environment based upon ETL Studio; not the direct costs of software and hardware but the changes that will need to be made to effect the new environment.

WHAT TO I NEED TO CHANGE?

The first and most obvious cost is in changing the processes that underpin the clinical reporting process. These changes include designing and altering the Standard Operating Procedures and ensuring that metadata standards are key to the entire process; these may need to be defined or enhanced.

The second area involves alterations to the infrastructure. Unless SAS 9 has already been implemented, it is unlikely that a separate metadata server will have been set up; this is recommended as an architectural standard for SAS 9 to ensure that the user interface is sufficiently responsive.

WHAT DO I NEED TO LEARN?

There are a number of new clients for the programmers and statisticians, plus those who will be managing the system infrastructure. There is also a new programming philosophy; create the infrastructure and design before any code is cut. This leads to better code with less errors and more maintainability; however there will be a period where it is less efficient as the new methodology is adopted.

WHAT TOOLS WILL I NEED TO CREATE?

SAS ETL Studio is a generic ETL tool that is designed to work in all business sectors; there are likely to be a few techniques that are not covered by the standard nodes but are common in clinical reporting. These will either need to be created as individual SAS code nodes or more likely as generic transformations.

It is also likely that custom metadata scripts will be required to support the integration of the tool into existing processes, for example to set up standard directory structures or to validate that the correct standards had been applied to a particular study.

WHAT ABOUT MY LEGACY CODE?

The good news is that SAS code will run in ETL studio unchanged, give or take a couple of library references and format catalogs. The bad news is that the metadata trail will not be automatically updated, although it would be possible to read some of this from the code.

For studies that have already completed it is probably better to leave the code as it is, but to register the metadata from the output of analysis dataset creation programs, allowing their exploitation in future integration studies and bringing them into the metadata search environment. For studies that are part way through, it may be worth considering migrating existing code to the new environment in order to feed integrated summaries later.

ARE MY PROGRAMMERS STILL PROGRAMMING?

There are several areas in which programmers would still need to use their skills in writing SAS code. Some of the code might look quite different where it is directly interfacing with the metadata environment, and the structures around the code that they do write will be quite different from those they are used to, but it is still SAS code.

It is also important to recognise that at least 50% of the skills base of a clinical programmer is in the comprehension of the clinical study data and the requirements that a statistician has for analysis-ready data, and those aspects of the job will remain unchanged.

WHAT ABOUT VALIDATION?

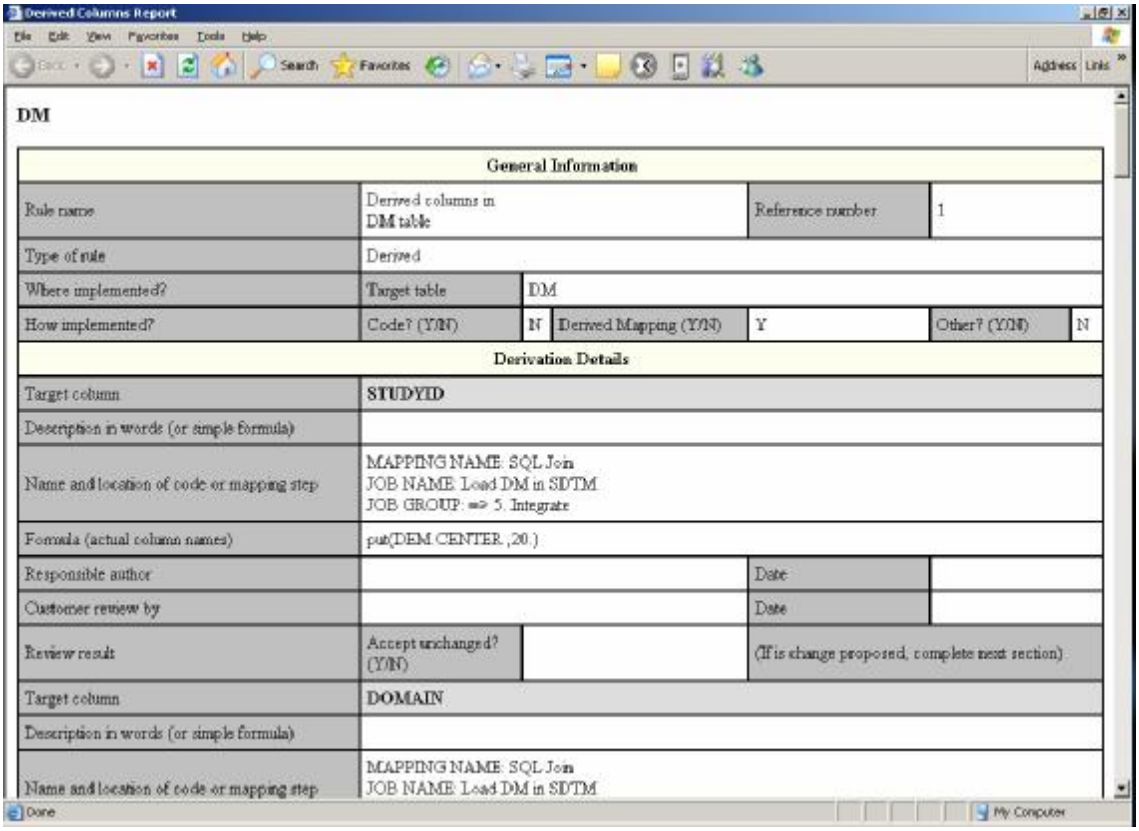
There will be an initial validation requirement, although this is to ensure that the code that is being generated is fit for purpose. Once that has been completed the validation would be of the derivations that have been applied to individual columns and any new transformations that are written. The underlying SAS 9 system is also a lot easier to qualify, as it ships with Installation and Operational Qualification tools to reduce the installation burden.

CUSTOMISING ETL STUDIO FOR THE CLINICAL ENVIRONMENT

A combination of custom scripts and extended metadata can be used to align the technology to clinical business processes. This ensures that the technology is used in an efficient manner and that there is less likelihood that it will be used incorrectly.

CUSTOM SCRIPTS

Custom scripts are able to directly read and write metadata and can therefore be used to integrate the metadata environment with the clinical process. A good example would be the validation of column derivations within a process; a script has been written which extracts all the column mappings that involve a derivation and then prints them out with a signature box for approval (see output in Figure 5).



Derived Columns Report

DM

General Information						
Rule name	Derived columns in DM table			Reference number	1	
Type of rule	Derived					
Where implemented?	Target table	DM				
How implemented?	Code? (Y/N)	N	Derived Mapping (Y/N)	Y	Other? (Y/N)	N
Derivation Details						
Target column	**STUDYID**					
Description in words (or simple formula)						
Name and location of code or mapping step	MAPPING NAME: SQL Join JOB NAME: Load DM in SDTM JOB GROUP: => 5. Integrate					
Formula (actual column names)	put(DEM.CENTER,20.)					
Responsible author				Date		
Customer review by				Date		
Review result	Accept unchanged? (Y/N)		(If is change proposed, complete next section)			
Target column	**DOMAIN**					
Description in words (or simple formula)						
Name and location of code or mapping step	MAPPING NAME: SQL Join JOB NAME: Load DM in SDTM					

Figure 5 : Derived Columns Report

Similarly reports could be written to summarise a clinical study reporting process (datasets created, datasets used, standards applied etc) or to check that metadata standards have been applied correctly.

Beyond the simple reporting side, custom scripts can be used to perform tasks where metadata is created, for example copying processes between studies. It is also possible to automatically deploy the target structures from a metadata hierarchy, essentially creating the output tables ready for loading from the source systems.

EXTENDED METADATA

Extended metadata can be used to link metadata standards from external sources (for example a data modelling tool or an external database of metadata source items) to ETL Studio metadata, thus providing metadata genealogy. This also ties in well with custom scripts, which can read the extended metadata items and perform different actions depending upon the contents. An example of this might be creating date mappings between a date-time source item and separate date and time items in the target dataset; the correct items would be identified by having a common column grouping (e.g. adverse event start date) and a type designation (date, time or date-time).

Extended metadata items are also required to fully implement industry standards such as the CDISC SDTM, which requires that each column has attributes such as Origin, Code, Role and References, and that each table has attributes such as Domain Abbreviation, Domain Type, Transport Filename, Version and Release Date.

HOW DOES ETL STUDIO FIT WITH SAS DRUG DEVELOPMENT?

SAS Drug Development (SDD) is a specific solution for the clinical development process, strongly enabling regulatory compliance. The main features are detailed auditing and versioning of all objects in the environment, a data browser that allows a consolidated view of all datasets in the study, a program development environment, archiving, scheduling and output bundling, all available through a web browser.

The most common configuration of ETL Studio with SDD would be to use ETL Studio as the development environment and promote to SDD for production. Data stored in SDD can be accessed by the rest of the SAS 9 toolset using a WebDAV library, thus opening it up to developing processes in ETL studio without compromising the SDD security model. Once testing of the processes is complete the code can be promoted into the SDD environment and the entire process comes under tight regulatory control.

HANG ON - I COULD WRITE ONE OF THOSE MYSELF!

It has been successfully proved in at least two companies that it is possible to write systems that manage metadata and/or the programming environment. SAS programmers are adventurous and like nothing more than a large system to build that will change the way that their company does business forever. However, such a system needs to be supported and change control needs to be properly implemented to maintain the validated state; in any event building software systems in a regulatory environment is not to be undertaken lightly! The experience of some companies has been positive; others less so. The ones that have succeeded tend to have experience of managing large IT projects and have the systems and processes in place to support them, and have particular good reasons why the commercially available alternatives do not fit their environment. Even where this is the case, there is anecdotal evidence that failures can still occur, and most systems have a relatively short life span compared to the effort involved in creating them.

CONCLUSION

As a result of the complexity of the clinical reporting process, the control of metadata is paramount for the efficient and accurate reporting of clinical studies. With its extensible metadata model and familiar language for extending its functionality, SAS ETL Studio is an ideal environment to assist clinical programmers with the production of analysis ready datasets and reports while still ensuring that the SAS code that they are familiar with is always ready for inspection and validation should the need arise. It can enhance the repeatability of studies, ensure that they can be efficiently and easily combined, and reduce the amount of revalidation when changes are made to source metadata or report specifications.

RECOMMENDED READING

SAS® 9.1.3 Intelligence Platform: Overview
http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_913/intell_overview_9036.pdf
SAS 9.1.3 ETL Studio: User's Guide
http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_913/etl_ug_8238.pdf

CONTACT INFORMATION

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

Dave Smith
SAS Institute
Wittington House
Henley Road
Medmenham
Marlow
Bucks
SL7 2EB

Work Phone: 01628 404379
Fax: 01628 490550
Email: david.smith@suk.sas.com
Web: http://www.sas.com