

# Using SAS® Macros to simplify preparation of SDTM data, Annotated CRFs and Define.xml in a metadata driven environment.

Niels Both, S-Cubed, Copenhagen, Denmark

## ABSTRACT

Many SAS programmers working within the pharmaceutical industry are involved in the conversion of clinical data to the Study Data Tabulation Model (SDTM) and creation of annotated CRFs and define.xml files. Several of the programming tasks involved in the preparation of such a process are almost identical from one study to the next and higher efficiency can be gained by using a systematic approach. This paper presents a conceptual model of a system, to describe how to set up the process to maximize the efficiency and avoid tedious repetitive tasks, by creating a set of SDTM generation macros and providing all the required study specific information via an electronic mapping specification. It gives presents all the individual system components in detail, especially the generic SDTM macros to ease the creation of the SDTMs, including macros to produce the SDTM relational datasets SUPPQUAL, RELREC and CO and standardised --TEST and --TESTCD variables.

## INTRODUCTION

In connection with submission of clinical data to FDA the guidance (ref 1) suggests that the CDISC SDTM format should be used for the data and describes how to document the data with annotated CRF (Blankcrf.PDF) and define.xml (CRT DD). The details of the the data model (SDTM) are found in refs 2 and 3, the define.xml in ref 4. Further (draft) suggestions regarding the annotated CRF are described in ref 5. All those documents are available on [www.cdisc.org](http://www.cdisc.org). This paper describes how to automate the procedures associated with the creation of those deliverables, using a systematic approach. SAS programmers with experience in this area know that a lot of the work involved in the creation of those deliverables can be done using generic processes due to the fact that the format of the deliverables is always identical. The part of the work that is specific to the individual clinical trial can be separated from the generic parts, and the generic parts can be reused to increase efficiency when more study data are to be converted. The diagram below illustrates the process and the process components:

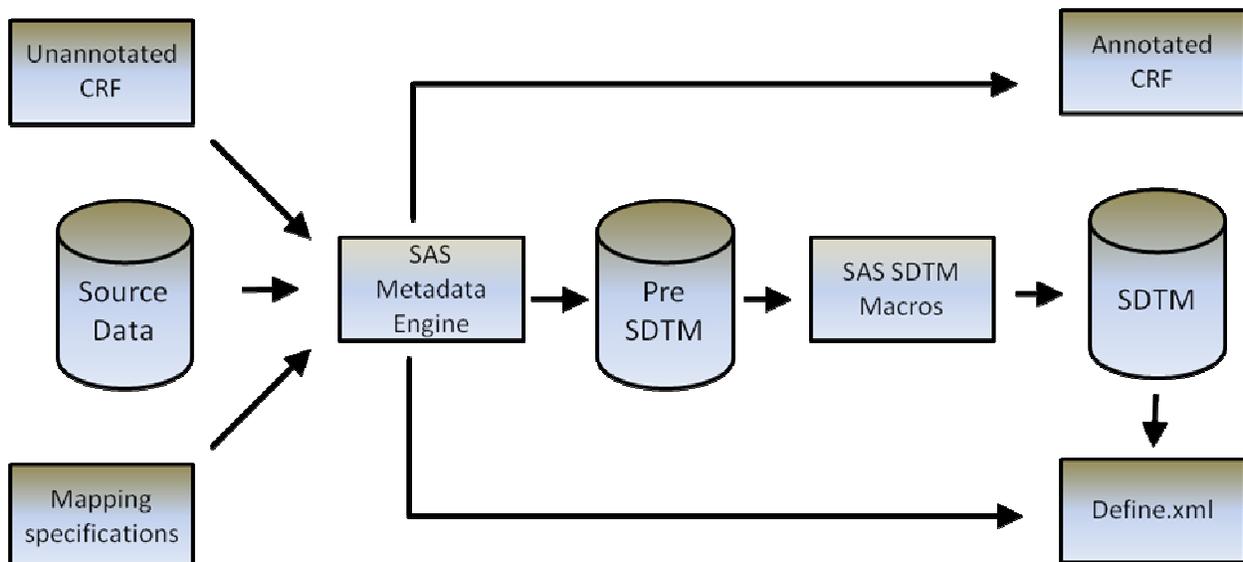


Diagram 1: Systematic overview of process for creation of submission deliverables

The individual components will all be described in the paper. In particular with a detailed focus on the SAS SDTM macros.

## PROCESS COMPONENTS

The components involved in the systematic creation of the 3 submission deliverables Annotated CRF, SDTM and Define.xml are described below. As shown on diagram 1 there are 3 inputs to the process: The unannotated blank CRF in PDF format, the clinical source data in their native format and a mapping specification, usually as an Excel spreadsheet. The mapping specification contains metadata regarding all the individual mappings and derivations (on variable level) which take place when transforming the source data to Pre SDTM.

The SDTMs are created as a 2 step process. The first step is to transform the source data into a Pre SDTM format, which is an SDTM-like format, but the Pre SDTM format deviates from SDTM in that, for example, all SDTM variables that can be auto generated are left out, no SUPPQUAL or RELREC datasets are present. The variables to be added to the SUPPQUAL or RELREC datasets are left in the parent domains at the Pre SDTM stage. The second step is to transform the Pre SDTM into Submission SDTM, compliant with the description of the model in refs 2 and 3. The first step is dependent on the format of the source data, whereas the second step, which is where the most complicated transformations take place, is independent of the format of the source data.

In addition to the source-to-target mappings, the mapping specifications also holds information regarding what page numbers of the CRF a specific SDTM variable has to be annotated onto by the metadata engine. Furthermore the mapping specification contains the text comments regarding variables, which eventually will form part of the define.xml. The concept is that the mapping specification contains all the information that has to be manually entered to drive creation of the submission deliverables. The amount of information to enter manually is minimized by having as much of the programming as possible made generic. The creation of the 3 individual components is described further in the sections below.

## CREATION OF ANNOTATED CRF

As specified in ref 1, the annotated crf associated with the SDTM datasets has to be in PDF format. The Study Data specifications (ref 1) do not provide many specific details about how to make the annotations part of the PDF file, so there is room to do it in different ways. Most companies choose to produce the annotated CRFs so that they are somewhat similar to what is described in ref 6, using Adobe® Acrobat 'comments'. 'Comments' are text boxes that can be added on top of the PDF pages, and can contain the annotations. Adding the 'comments' manually using Adobe Acrobat Pro, is a tedious process. A better approach is to use the fact that metadata regarding position, contents and fonts of 'comments' can be exported or imported by Adobe Acrobat 9 Pro as fdf (form definition files). The fdf files are ASCII files, and can be created, read or modified using SAS. The process of annotating the CRF is therefore a process that can be automated. How to create annotated CRFs is very well described in ref 6 already, and will not be described further here. The conclusion is that creation of an annotated CRF in PDF format can be automated by adding sufficient information to the mapping sheet.

## CREATION OF DEFINE.XML

The structure of define.xml is described in ref 4. Define.xml contains metadata information about datasets, variables and formats, as well as links to pages in the annotated crf and text comments regarding variables in an xml structure that are machine readable. The define.xml file can be made human readable using style sheets. Most of the metadata required for define.xml are already present in the SDTM datasets themselves, and the remaining information (page numbers to link to, and text comments) can be retrieved from the mapping sheet. More details regarding SAS based creation of define.xml is supplied by ref 7. In the future it might be possible to create define.xml using SAS Institute provided software when SAS Clinical Data Toolkit is released. However conceptually there is no difference between using a homemade SAS program to create the XML file and using the toolkit. In both cases the process needs input from metadata present in the SDTM datasets plus additional information regarding page numbers in the CRF to link to and comments regarding variables to insert in the define file. Please note that in many cases the Food and Drug Administration also request the submission of the define file in PDF format. The define.pdf file contains the same information as define.xml and can be readily created based on the define.xml file. The diagram below illustrates the different sources of information required for creation of define.xml

## PhUse 2009

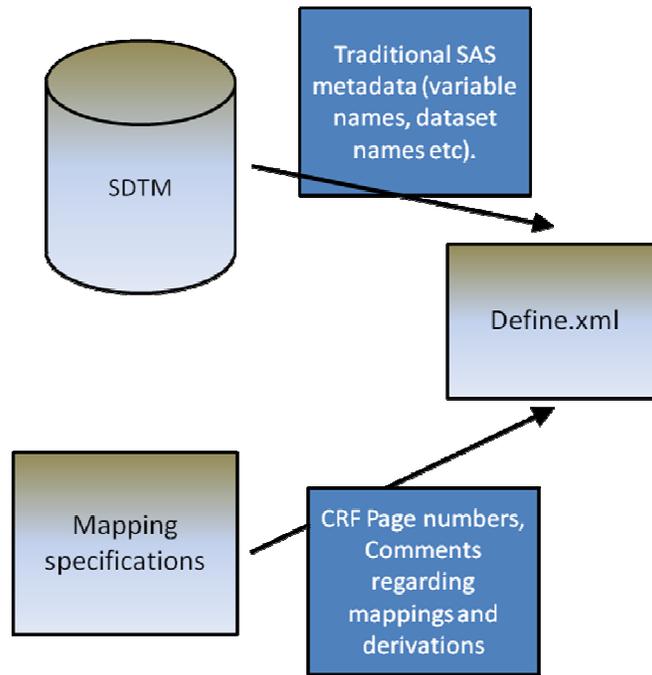


Diagram 2: Overview of sources for creation of define.xml

### CREATION OF SDTM STEP 1: FROM SOURCE DATA TO PRE SDTM

The creation of SDTM is a 2 step process. The first step is the transformation of data in their native source format to a Pre SDTM format. The second step is a fully automated generic step, to do all the transformations using SAS macros. The below diagram illustrates the 2 step process, where step one is driven by manual specifications.

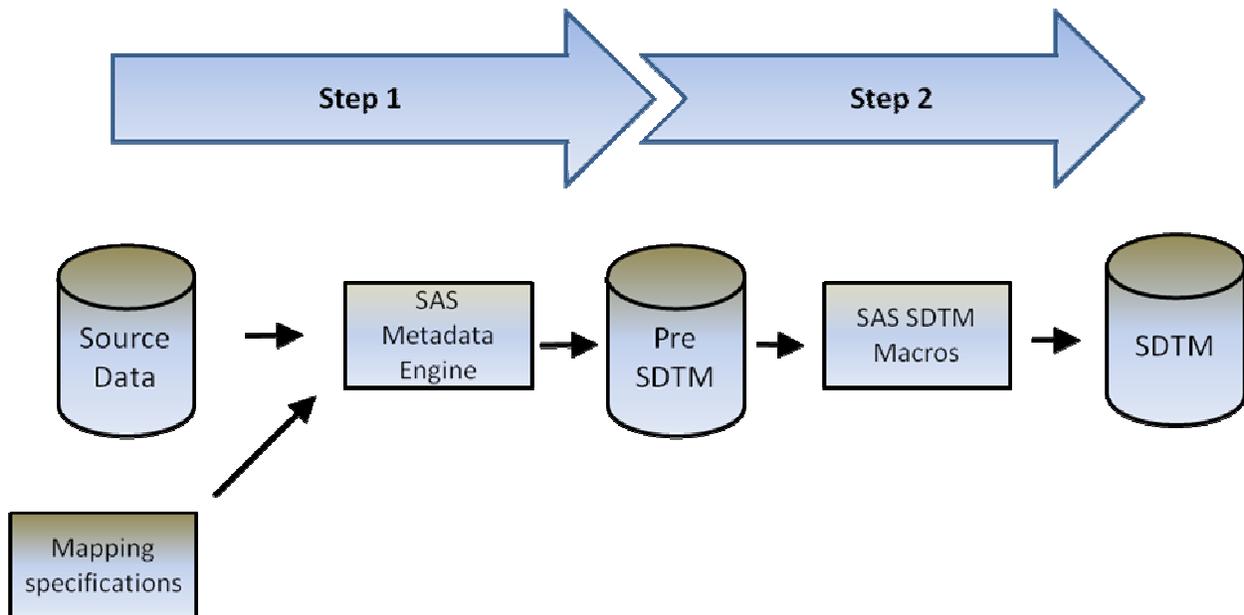


Diagram 3: Stepwise process to create SDTM, where the manual processes are in step one, and automated processes are in step two.

Once a suite of programs for creation of submission deliverables is in place, the definition of mappings from native format to Pre SDTM is the only step that requires a lot of manual work. Defining how to do the transformation from

## PhUse 2009

the study specific native format to Pre SDTM requires a thorough examination of the source data and solid knowledge of the SDTM model and CDISC controlled terminology, in order to map correctly. As mentioned above, the mapping can be defined in, for example, an Excel file, which is then read by the SAS Metadata engine, which creates the transformation programs as ascii files and then executes the created SAS code. Using this kind of approach means that the specification of mappings can be done by experts in clinical data, who are not necessarily SAS programmers. Another benefit of using a system to create the SAS code is that it eliminates the risk of the code having bugs. A disadvantage is that most often this kind of machine generated code is not as efficient as human created code. The process of turning the mappings into a program and executing the program can be done by ETL tools such as SAS Data Integration, but note that this can also be made from scratch by an experienced SAS programmer. If that solution is chosen it allows the company to choose a format for the mappings which is tailor made for that company, making the tedious mapping process as straight forward as possible. Also note that the tedious effort of mapping source data to Pre SDTM can be dramatically reduced and maybe even eliminated by using CDISC CDASH as the native source data format. For a definition of CDASH please refer to ref 8.

### CREATION OF SDTM STEP 2: FROM PRE SDTM TO SDTM

The second step in the creation of SDTM is to transform the Pre SDTM into compliant SDTM. This step can be performed by a suite consisting of 14 SAS Macros. Each of the macros performs a well defined action, to produce or change specific variables or to perform specific complex data transformations such as creating supplementary qualifier datasets (SUPPQUAL) (ref 3 section 8.4.1) or RELREC (ref 3 section 8.2). All of the macros take advantage of the strict SDTM variable naming convention and the consequent way of naming variables within a specific class (ref 2 section 2.2). Some of the macros are dependent on previous macros, it is thus essential that the macros are executed in the correct order as presented in table 1 below. The macros can and should be built in a way that they are 100% data driven, and do not require any study specific input what so ever.

#	Macro name	#	Macro name
1	%Usubjid	8	%Relrec
2	%Domain	9	%Studyid
3	%Dtc	10	%Bl
4	%Dy	11	%Stc
5	%Seq	12	%Res
6	%LongChars	13	%SDTMify
7	%SuppqualCo	14	%PhysicalTransform

Table 1: Overview of SAS macros to create submission SDTM datasets.

#### Macro 1: %Usubjid

The macro %Usubjid changes the contents of the the variable USUBJID. According to the definition of the variable in ref 3, the contents of USUBJID should uniquely identify a patient across several sets of SDTM datasets, so that 2 different subjects do not have the same USUBJID. In the vast majority of situations, this is achieved by prefixing the original subject ID, with the study and (sometimes) the centre ID, so that for example subject 0123, from center US001 in trial 'SCUB01', has USUBJID="SCUB01\_US001\_0123".

The purpose of the macro %Usubjid is to retrieve information about center and study from the demographics domain (DM), and add this to all domains that contain the variable USUBJID, by prefixing it with study and center IDs.

#### Macro 2: %Domain

All SDTM domains, except the Special-Purpose Relationship Domains RELREC and SUPPQUAL have a required variable 'DOMAIN' in their definition. The macro %Domain runs through all the datasets in the Pre SDTM and assigns a value to the variable 'DOMAIN'. The values are the uppercased dataset name, so that the dataset ae, for example, has DOMAIN="AE" for all its records.

#### Macro 3: %Dtc

All dates in SDTM should be recorded as ISO8601 format dates (see ref 3 section 4.1.4.1). The ISO8601 is a character format that can be used to store completely known date/time points as well as partly known date/time point. In the Pre SDTM all date and time variables are stored in a 100% consistent way that allows for storage of partly unknown date/times. This can be achieved for example by defining that any date or datetime value in Pre SDTM is divided into 5 components, one for the year, one for the month, one for the day (within the month), one for the hour and one for the second. For example an AE start date of 19<sup>th</sup> may 2007, is stored as AESTDTC\_YY=1970, AESTDTC\_MM=5, AESTDTC\_DD=19, AESTDTC\_HH=, and AESTDTC\_SS=. The hours and second values are blank as they are unknown or uncollected. If they had been known, they would be populated of course.

The macro %Dtc identifies all date/time variables in the Pre SDTM and loops through all relevant datasets, in order to produce the compliant SDTM date/time variable in ISO8601 format. The 5 original date/time variables in the Pre

## PhUse 2009

SDTM are dropped after the ISO8601 variable is created.

### Macro 4: %Dy

According to ref 3 section 4.1.4.4 the study day variables (--DY) describe the relative day of the observation starting with the reference date as Day 1. It is determined by comparing the date portion of any DTC variable to the date portion of the Subject Reference Date (RFSTDTC from the Demography domain). The subject reference date is designated as Study Day 1. The Study Day value is incremented by 1 for each date following RFSTDTC. Dates prior to RFSTDTC are decremented by 1, with the date preceding RFSTDTC designated as Study Day -1 (there is no Study Day 0).

Thus, the algorithm to calculate Study Day is:

```
--DY = (date portion of --DTC) - (date portion of RFSTDTC) + 1 if --DTC is on or after RFSTDTC  
--DY = (date portion of --DTC) - (date portion of RFSTDTC) if --DTC precedes RFSTDTC
```

The macro to calculate study day is %Dy. It uses RFSTDTC from DM, and runs through all date/time variables in all domains, and calculates the corresponding --DY variable (except for RFSTDTC and RFENDTC). It is not required by the SDTM model to calculate all the --DY variables, however the --DY variables are always at least permissible when a date/time variable is present, and it is recommended to always add the --DY variable in those cases. Calculating them all offers a systematic solution.

### Macro 5: %Seq

On all the domains in the general observation classes a counter variable (--SEQ) is required, which together with USUBJID uniquely identifies a record within a dataset. In most cases the --SEQ variable counts from 1 and then increments with one for all the records for a given USUBJID, for a new USUBJID the counter starts again from 1. The macro %Seq initially sorts the datasets in a meaningful way for example by USUBJID and date. After that the macro goes through the datasets and assigns values to the variable --SEQ for each new USUBJID. Please note that the sequence number variables is only in very rare cases (if ever) a CRF variable. If counters exist in the CRF source data - for example line numbers on a Medical History (MH) or Adverse Event (AE) page, they are to be stored in the --SPID variables within SDTM.

### Macro 6: %LongChars

SDTM datasets for submission are to be stored as SAS version 5 xpt files. One of the restrictions that requirement puts on the SDTM model, is that no character variable can have a length of more than 200 characters, however the native source data may well contain values longer than 200 characters which have then been transferred to the Pre SDTM. The macro %LongChars is part of the solution to that problem. In section 4.1.5.3 of ref 3, it is presented how to store the long text strings in SDTM. The specified solution is to separate the part of the text that is beyond 200 characters and store it as separate records in supplementary qualifier datasets. Macro 6 does not create supplementary qualifier datasets; however it does solve part of the problem with the long texts. Macro 6 %LongChars, identifies any text string in the data that is beyond 200 characters long and then splits it up and creates new variables on the datasets, containing the extra text that cannot fit into the required 200 characters.

If the variable that is too long is for example MHTERM, then the new variable created is named SQ\_MHTERM1, if one new variable is not enough, another one called SQ\_MHTERM2 is created and so forth. These variables are NOT allowed on the domain as part of the SDTM model. Macro 7 described below will identify them as being supplementary qualifier variables and move them accordingly to SUPPQUAL where they belong. The rules for long texts in the domains TS and CO is slightly different, so the macro handles those domains following the rules laid out in ref 3 sections 5.1.2 and 7.10.

### Macro 7: %SuppqualCo

In SDTM datasets it is not allowed to add any variables which are not defined as part of the model. Such additional information is to be stored as individual records in supplementary qualifier datasets – see sections 4.1.5.2 and 8.4 of ref 3. Creating the Pre SDTM datasets is far easier without the need to also create supplementary qualifier datasets. Macro 7 allows just that. During generation of the Pre SDTM all the variables that belong in a supplementary qualifier dataset are left on the Pre SDTM, but prefixed 'SQ\_' to indicate a supplementary qualifier variable. The macro %suppqualCo identifies the SQ variables, runs through all the individual Pre SDTM datasets cutting off the SQ variables and storing them as individual records in the appropriate supplementary qualifier datasets, linked back to the parent domain by USUBJID, RDOMAIN, IDVAR and IDVARVAL. The process is illustrated in the diagram below:

## PhUse 2009

AE (Pre SDTM)

STUDYID	DOMAIN	USUBJID	AESEQ	AETERM	AESER	...	SQ_AESOTHC
SCUB01	AE	123	1	TRAUMA	Y	...	Neck Injury

SUPPAE

STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	QNAM	QLABEL	QVAL
SCUB01	AE	123	AESEQ	1	AESOTHC	Desc of Other Event	Neck Injury

AE (SDTM)

STUDYID	DOMAIN	USUBJID	AESEQ	AETERM	AESER	...
SCUB01	AE	123	1	TRAUMA	Y	...

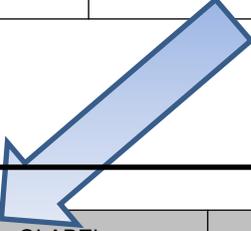


Diagram 4: The %supqualCo macro cuts out a supplementary qualifier variable from Pre SDTM, and stores it in SUPPAE.

This concept and the SAS code for it, is partly presented already in ref 9. The concept presented in ref 9 however can be improved by using a naming convention for the supplementary qualifier variables, and making the process 100% data driven, by using advanced SAS macro code.

Please note that the macro works on the texts that have been cut into smaller pieces by macro 6, to make the storage of character variables longer than 200 characters meet the SDTM model requirements.

The macro also handles creation of records in the SDTM dataset CO. CO can contain two fundamentally different types of comments. It can contain either free text comments collected on a CRF or other comments directly related to another SDTM domain (ie if a comment is made regarding a particular blood sample in the lab data). In the latter case, the information should be stored in the CO domain, using variables RDOMAIN, IDVAR, IDVARVAL and USUBJID, to link it back to its parent domain in a fashion which is identical to the SUPPQUAL mechanism. If the textual comments to be stored in CO are prefixed 'CO\_', the macro stores them appropriately in the CO domain.

### Macro 8: %Relrec

According to ref 3 section 8.2, the Related Records (RELREC) dataset is used to describe relationships between records in two (or more) datasets, such as an Event record and an Intervention record, or a Finding record and an Event record. The relationship is described as records in the RELREC dataset, where the same mechanism (USUBJID, RDOMAIN, IDVAR and IDVARVAL) as used in the supplementary qualifiers is used to point at records in other domains. The variable RELID is used to indicate the relation between records. The example below illustrates how a relationship between an adverse event and two concomitant medications should be expressed as 3 records in RELREC.

## PhUse 2009

### RELREC

STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID
SCUB01	AE	123	AESEQ	3		AECM1
SCUB01	CM	123	CMSEQ	11		AECM1
SCUB01	CM	123	CMSEQ	12		AECM1

*Example 1: RELREC dataset with 3 records.*

In the example above, the adverse event with AESEQ=3 is associated with the two Concomitant medications with CMSEQ=11 and CMSEQ=12. This information about the relationship could have been collected on the Concomitant medication page, where it could have been asked "Is the CM taken in relationship with an adverse event? If yes, please specify the adverse event number". The specified number would be a record number on the adverse event. That record number would be stored in AESPID both in Pre SDTM and SDTM. Turning that type of information about a relationship into records in a RELREC dataset is usually a significant challenge. The macro %Relrec can do it in a generic fashion, provided that strict naming convention and syntax rules are followed when creating the Pre SDTM datasets.

For the macro to function, the information regarding the relationship has to be stored in a variable prefixed 'RELREC\_' in order for the macro to recognize it as something that belongs in RELREC. In this example the variable could be named 'RELREC\_AE'.

The contents of the variable RELREC\_AE collected on the CM page should not only contain the collected record number, but in addition also information about what variable is the key for the relation on the AE page. In this example it is AESPID and if the related record on the adverse event page has AESPID=3 (Please note that it is not necessarily identical to AESEQ) then both of the concomitant medication records should have the RELREC\_AE set to "AESPID=3", to indicate the relation.

Due to the fact that a strict naming convention and syntax is used, the macro %Relrec can identify the appropriate variables to be used in RELREC and can identify the appropriate relationships using the contents of the variables.

First the macro identifies all variable names starting with 'RELREC\_'. Then the macro loops through the datasets to identify which relations exist using the fact that the 2 first characters of the string is the domain name the relationship points to (in this case 'AE'). The macro then runs through the AE dataset, to identify the AESEQ, which has AESPID=3 and USUBJID=123. This record – together with the two from the CM domain - forms the 3 records in RELREC which are then created by the macro. The RELID has to be the same for all 3 records; in this case it is chosen to use the names of the involved domains with an incremental number to be used if there is more than one of the same type of relationship for the same USUBJID.

Please note that the concept is expandable to handle relationships between more than 2 domains and situations where the same record in a dataset is involved in more than one relationship.

The %Relrec macro is build with advanced macro code, in order to provide a generic solution to this challenging issue of creating RELREC records.

#### Macro 9: %Studyid

The macro %Studyid is a very simple macro that finds the appropriate study id in the DM domain, and then applies it to all the other domains, thus allowing the person mapping the Pre SDTMs to only worry about applying the study id variable once, namely on the DM domain.

#### Macro 10: %BI

The SDTM Findings domains have an expected variable named --BL, to be used to flag baseline assessments. The existence of these baseline flags is under debate in the CDISC teams, as they are normally regarded as statistical flags, which is part of the CDISC analysis dataset model (ADaM) and does not really fit into the SDTM model at all.

## PhUse 2009

Nevertheless the flags need to be set in SDTM, both according to the previous version of the model (1.1), and the latest production version (1.2). If a sponsor chooses to follow the SDTM model and set the flags in the SDTM datasets, then it can be done using the %BI macro.

The macro uses the reference start date (RFSTDTC) in the demographics domain (DM) to define baseline. The records are sorted by 'type of assessment', then by the assessment date. The latest record prior to RFSTDTC is flagged as the baseline. It is important to consider how to define 'type of assessment'. On the vital signs domain the variable VSPOS for example is used to distinguish between sitting and standing systolic blood pressure. On the lab domain the variable LBSPEC is used to distinguish between what specimen type the sample was. A number of variables like --POS, --SPEC and --EVAL, thus need to be taken into account when defining the baseline flag, and not only the obvious --TESTCD variable.

Please note that the use of baseline flags in the SDTM is debatable, the optimal solution would be to transfer the baseline flags from the ADaM datasets to the SDTM, but in many cases it is not possible as the definition of a baseline flag from a statistical point of view is more advanced than what the SDTM model can contain. It is a good solution to use a simple and systematic approach to set the SDTM baseline flags and at the same time make comments in the define.xml stating how the flags are set and commenting that the flags are not used in the statistical analysis.

### Macro 11: %Stc

On the SDTM findings domains the results are stored using in a number of different variables. According to ref 2, the rules are that --ORRES is used to store the originally collected result regardless if it is a numeric or a character based result. --ORRESU is used to hold the associated unit. In addition to the --ORRES variables, the model has variables to hold the results in a standardised fashion as well. The variable --STRESC is used to hold the standardised result, regardless if its character or numeric, whereas the variable --STRESN is used to hold standardised numeric results only. --STRESU is the standardised unit. --ORNRLO, --ORNRHI, --STNRLO, --STNRHI and --STNRC is used to store normal range information. It is described further in ref 2 section 2.2.3.

In Pre SDTM the result variables to be populated are --ORRES and --ORRESU (and potentially --ORNRLO and --ORNRHI), as they hold the raw collected data. The macro %Stc and the following macro %Res standardise the results and populate all the appropriate variables. The first of the 2 macros %Stc is only concerned with the conversion of results to new units and loops through all the findings datasets then determines the conversion algorithm based on the value in the test code variable (--TESTCD) and the unit (--ORRESU). The macro has a wide range of combinations of test codes and units hardcoded in, so that all assessments which are normally encountered within clinical trials can be converted to standard units. If a value in --ORRES is a character result; it is not converted but just copied straight into --STRESC. A numeric result gets converted if needed. Results that start with the characters '>', '<', '<=', '>=' are also converted if the remaining part of the variable is numeric. The normal range variable is converted following the same rules as the result variables.

The macro %Stc consists of a long list of conversion algorithms which ideally cover all test types that can be encountered. However, this approach leaves open the possibility that when a new conversion needs to be added, the program itself would need to be amended, which could entail extra validation effort. Another way to approach this would be to create a central and separate resource containing the conversion information, which the program could refer to. This way the programs do not need to be changed when a new test is added.

### Macro 12: %Res

The macro %Res is connected to the previous macro %Stc. %Res is a more simple macro that is used to populate the field --STRESN in case the value of --STRESC is numeric. It simply checks if the --STREC variable has a value which is a number and, if so, transfers it to --STRESN.

### Macro 13: %SDTMify

This macro is used to streamline the SDTM datasets by making final adjustments to them. It performs a number of operations on the datasets. First of all it assigns labels to the variables, so that the person doing the work to define the Pre SDTM does not need to spend time on assigning any labels. All the labels are defined by the electronic SDTM definition which is available on the member's area of the CDISC website ([www.cdisc.org](http://www.cdisc.org)). In addition the macro sorts the records in the datasets and puts the dataset columns in the order which is specified in the mapping sheet. Finally the macro checks if there are any required or expected variables completely missing from the datasets and produce a warning if that is the case.

### Macro 14: %PhysicalTransform

The final step in the process to create submittable SDTM datasets is to physically transform the SDTM datasets from SAS version 9 files, to SAS version 5 xpt files. This is handled by the macro %PhysicalTransform.

## PhUse 2009

### ADDITIONAL COMMENTS REGARDING THE SDTM CREATING MACROS

The macros which are listed above constitute a generic way to post process a sponsors Pre SDTM datasets. It was chosen to split them up into many separate macros to make the process transparent. It could potentially have resulted in better performance if some of the macros were integrated, however good performance was not a priority when defining the macros. Exactly how to apply such a series of generic macros and exactly which ones to apply, depends to some extent on the different company they are used in. However in the majority of cases, a number of the above listed macros or some which are very similar need to be applied. The macros above do not consider the creation of trial design domains and in the above described setting the trial design domains have to be added as part of the Pre SDTM. If the SDTM datasets are to be created in a setting where some of the information regarding trial design is available in electronic format (for example, if the sponsor is using the CDISC Protocol Representation model), then it is feasible to create the trial design domains using a sas macro as well.

### CONCLUSION

The process to create annotated CRFs, Define.xml and SDTM datasets can to a high degree be automated by using data driven, advanced SAS Macros., therefore minimizing the study specific work as much as possible. Generic, data driven, SAS Macros, can also be used to create advanced datasets such as RELREC and the supplementary qualifier datasets.

### REFERENCES

- 1: eCTD Study Data Specifications  
([www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM163561.PDF](http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM163561.PDF))
- 2: Study Data Tabulation Model version 1.1 Study Data Tabulation Model (SDTM) Final Version 1.1, May 4, 2005, prepared by the CDISC Submission Data Standards Team  
(Available from [www.cdisc.org/models/sdtm/v1.1/index.html](http://www.cdisc.org/models/sdtm/v1.1/index.html))
- 3: Study Data Tabulation Model (SDTM) Implementation Guide Final Version 3.1.1, September 8, 2005, prepared by the CDISC Submission Data Standards Team  
(Available from [www.cdisc.org/models/sdtm/v1.1/index.html](http://www.cdisc.org/models/sdtm/v1.1/index.html))
- 4: Case Report Tabulation Data Definition Specification (define.xml) Prepared by the CDISC define.xml Team  
([www.cdisc.org/models/def/v1.0/CRT\\_DDSpecification1\\_0\\_0.PDF](http://www.cdisc.org/models/def/v1.0/CRT_DDSpecification1_0_0.PDF))
- 5: Metadata Submission Guidelines, Appendix to the Study Data Tabulation Model Implementation Guide 3.1.1 prepared by the CDISC Submission Data Standards Team  
([www.cdisc.org/standards/index.html](http://www.cdisc.org/standards/index.html))
- 6: Using SAS to Speed up Annotating Case Report Forms in PDF Format prepared by Dirk Spruck and Monika Kawohl (PharmaSuc 2004 paper CC02)  
([www.lexjansen.com/pharmasug/2004/CodersCorner/CC02.PDF](http://www.lexjansen.com/pharmasug/2004/CodersCorner/CC02.PDF))
- 7: A SAS based solution for define.xml prepared by Monika Kawohl (PhUSE 2007 paper RA02)  
([www.lexjansen.com/phuse/2007/ra/ra02.PDF](http://www.lexjansen.com/phuse/2007/ra/ra02.PDF))
- 8: Clinical Data Acquisition Standards Harmonization (CDASH), Prepared by: CDISC CDASH Core and Domain Teams  
([www.cdisc.org/standards/cdash/downloads/CDASH\\_STD-1\\_0\\_2008-10-01.PDF](http://www.cdisc.org/standards/cdash/downloads/CDASH_STD-1_0_2008-10-01.PDF))
- 9: A Toolkit for CDISC Implementation prepared by Robert T. Stemplinger (PharmaSuc 2008 PO02-2008)  
(<http://www.lexjansen.com/pharmasug/2008/po/po02.PDF>)

## PhUse 2009

### CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the author at:

Niels Both, S-cubed ApS,  
Email: [nb@s-cubed.dk](mailto:nb@s-cubed.dk),  
Telephone; +45 31267677,  
Web: [www.s-cubed.dk](http://www.s-cubed.dk)

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 the respective companies.