PharmaSUG 2014 – DS08

Challenges of Processing Questionnaire Data from Collection to SDTM to ADaM and Solutions using SAS®

Karin LaPann, PRA International, Horsham, PA
Terek Peterson, MBA, PRA International, Horsham, PA

ABSTRACT

Often in a clinical trial, measures are needed to describe pain, discomfort, or physical constraints which are visible but not measurable through lab tests or other vital signs. In these cases, researchers turn to questionnaires to provide documentation of improvement or statistically meaningful change in support of safety and efficacy hypotheses. For example, in studies (i.e. Parkinson's) where pain or depression are serious non-motor symptoms of the disease, these questionnaires provide primary endpoints for analysis.

Questionnaire data presents unique challenges in both collection and analysis in the world of CDISC standards. The questions are usually aggregated into scale scores, as the underlying questions by themselves provide little additional usefulness. The SAS system is a powerful tool for extraction of the raw data from the collection databases and transposition of columns into a basic data structure in SDTM which is vertical. The data is then processed further as per the instructions in the Statistical Analysis Plan (SAP). This involves translation of the originally collected values into sums, and the values of some questions need to be reversed. Missing values can be computed as means of the remaining questions. These scores are then saved as new rows in the ADaM (analysis-ready) datasets. This paper describes the types of questionnaires, how data collection takes place, the basic CDISC rules for storing raw data in SDTM, and how to create analysis datasets with derived records using ADaM standards; while maintaining traceability to the original question.

INTRODUCTION

We are in a wonderful and fantastic era where standardized data accelerates the speed of discovery of great and novel compounds that will better the lives of patients. Sponsors, CROs, agencies, and standard's organizations must continue to push aggressively to that end or we as information miners will fail to deliver that elusive cure in time for that one disease that will improve a life in our family or someone else's family. Data standards sometimes seem like they add overhead, but they can speed the creation of that compound, device, or new therapy to improve or even save people's lives

The CDISC data standards are intended to organize very complex instruments into a vertical datamart type structure for ease of storage and analysis. In the case of questionnaires, there are hundreds of different types. Confounding this further is the clinical programmer must store this information in one or more tabulation datasets then again in one or more analysis dataset(s). There is a good reason for this as each type answers different data needs. First they are stored in the study data tabulation model (SDTM) which has been developed over the last 14 years to standardize the collection of study data so that it can be aggregated with other study data, and eventually submitted for approval of a drug to the FDA or other regulatory agency as part of the submission process. The SDTM should be representative of the originally collected data, and with minimal derivations. Analysis-ready datasets are the second way the clinical programmer stores the data by using the CDSIC ADaM standard that have been developed and are published in the ADaMIG [1]. The analysis-ready datasets are used to carry out analyses for efficacy of the drug, for safety tables and sometimes for patient profiles. The second structure allows any total scale scores and sub-scale scores to be computed and output as separate rows from the original data. This methodology allows efficiencies in creating tables that display the results and also provides traceability (being able to reproduce the same from original data) back to the original collection instruments.

SOME EXAMPLES OF QUESTIONNAIRE USAGE

In pharmaceutical drug research, questionnaires are often used to quantify feelings such as pain and depression which are not otherwise quantifiable with straight-forward readings such as laboratory, electrocardiograms, and other vital signs. These instruments are so important for research because they allow us to compare before and after treatment responses for people that suffer from these types of diseases. There are numerous questionnaires used to capture this type of data. Validated questionnaires carry more weight because they have been medically and statistically examined to verify that significant differences can be shown by change in the scale and subscale totals.

One paper that explains how to validate your own scale is mentioned in the references is from SUGI 29, describing the options of using PROC VARCLUS vs. Factor Analysis. [2] Needless to say, it is more meaningful to use questionnaire scales that have already been validated by psychologists, published in journals, or extensively used in other studies. There are two types of validated questionnaires, one is public domain, and the other is proprietary. Proprietary questionnaires may require some sort of fee to be able to use them. For the indication of Parkinson's disease for example, the following validated questionnaires were used in a study: the Parkinson's disease Questionnaire (PDQ-8), Hospital Anxiety and Depression Scale (HODS), the Unified Parkinson's Disease Rating Scale, Parts II, III and IV (UPDRS), Clinical Global Impression of Change (CGISC), Fatigue Severity Scale (FSS), and the Likert Pain Scale (LPS)

Another study for Multiple Sclerosis used a variety of questionnaires, including the Patient Health Questionnaire (PHQ-9), Fatigue Severity Scale (FSS), the SF-12, the Patient Determined Disease Steps (PDDS), the Work Productivity and Activities Impairment – General health (WPAI-GH), the MSIS-29, the Expanded Disability Status Scale Score (EDSS) and the TSQM-9. Of this set, only 3 have been defined in the CDISC set of standards (FSS, WPAI-SHP, PDDS). Therefore sponsors need to have ways to define these scales on their own. Further on in this paper will be presented tips for naming conventions and other details.

These all were part of the primary or secondary efficacy endpoint analyses. Because all of these questionnaires included a variety of scoring methodologies they were broken up into multiple ADQS datasets. The MSIS-29 for example has so many different scores that it was designated its own analysis dataset of ADQSMSIS. Other instruments were combined in ADQSGEN (for General Questionnaires) because they were not primary endpoints. In ADaM it is allowable to name a dataset with additional descriptive information as long as the dataset name is 8 characters long. Also, in ADaM all datasets must have descriptive dataset labels that clearly describe the contents of said dataset.

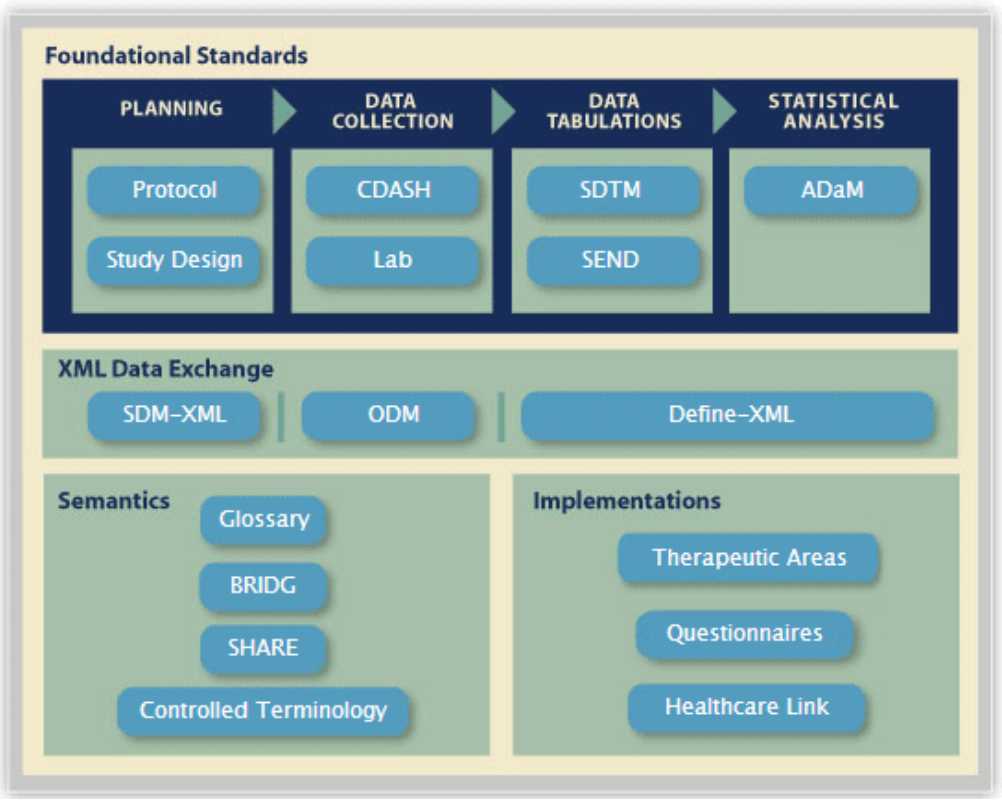
In addition to published and validated questionnaires, sponsors may create questionnaires that are in the form of yes/no responses that describe a patient feelings or unquantifiable characteristics. These may be stored in questionnaire datasets as well comingled with survey questions which are also then analyzed. As described, the combination of the many forms of questionnaire data can create quite complex data sets. However, by using the SDTM standards, they are easily identified by variables described as 'Identifier' variables.

CDISC INITIATIVES

The CDISC sub-team for questionnaire data has an ongoing project to annotate and create SDTM specifications for commonly used questionnaires. Each one includes a sample Annotated CRF form for CDASH and an SDTM specification. Thus far, as of Dec 2013, 54 questionnaire instruments have been catalogued and specified by CDISC volunteers, representing industry experts across pharmaceutical companies, CROs and FDA. This work includes 1) a sample annotated CRF, a .PDF document with some of the history and description of the scale, and also QS specifications with detailed variable names and labels. Added in December 20, 2013, is a full set of controlled terminology (CT) for the questionnaire data. This has been distilled from the specifications and code lists found in the questionnaires that are being mapped to SDTM and are now available at the following site in a variety of formats, including Excel, text, odm.xml, pdf, html and OWL/RDF formats. For those not familiar with controlled terminology, it is stored in the Cancer.gov website as part of the NCI Enterprise Vocabulary Services (EVS). The questionnaire data is currently separate from the other SDTM CT file which is also maintained by EVS. [3]

If a sponsor has not yet defined this instrument in their own standards, it is these authors' opinion to leverage the work that has taken place representing hundreds of hours, and adopt these for your use. The documents are on the CDISC website (www.cdisc.org) under the heading of STANDARDS & INNOVATIONS > Implementations > Questionnaires[3]. The questionnaire initiative for SDTM does cover annotating an eCRF. Although the collection forms are often not-CDASH compliant and in normalized form, these have been accepted as alternate acceptable eCRF structures and also have new CT associated with them.

The picture below is from the CDISC website [4], showing areas that are being addressed by the various standard sub-teams in the Standards & Implementations section.



source: http://www.cdisc.org/standards-and-implementations

1. STANDARDS INFORMATION FOR SDTM

SDTM standards have been developed since the year 2000 and are now in version v3.2, which was released on Nov 26, 2013. [5]

In SDTM, most of the questionnaire data, especially validated versions are mapped to QS. The QS dataset follows the guidelines first presented in the SDTMIG v3.1.2 guide. The QS dataset is part of the Findings Observation Class. This class captures the observations resulting from planned evaluations. It is vertical in structure, so any questions collected on one row with individual variable names need to be transposed. Here is a summary of the common fields expected in the SDTM specifications. The core columns defines the variables as required (always part of the dataset and never null), Permissible (Optional, not required) and Expected (required to be in dataset but can contain null values). Further details are provided in the CDISC notes column of the source document. Although lengths are not listed in the document, most are allowed to be up to 200 characters, except for Domain, length = 2, QSTESTCD length = 8 and QSTEST length = 40. Flag variables are expected to be length = 1.

The SDTM datasets should representative of the originally collected data, and without derivations. Usually the questionnaire and survey data are stored in SDTM QS, but questionnaires can also be found in FA and in sponsor defined domains starting with the letter X, such as XS or XD. The primary purpose for SDTM is to store collected research data in a logical way in order to create listings, and to use it later in creation of analysis datasets. The complex derivations should be reserved for the ADaM dataset programming based on the details specified in the Statistical Analysis Plan (SAP) and in these authors' opinion, not be attempted in the original tabulation datasets.

Table 1 SDTM Specifications

Seq. For Order	Variable Name	Variable Label	Type	Role	Core
1	STUDYID	Study Identifier	Char	Identifier	Req
2	DOMAIN	Domain Abbreviation	Char	Identifier	Req
3	USUBJID	Unique Subject Identifier	Char	Identifier	Req
4	QSSEQ	Sequence Number	Num	Identifier	Req
5	QSGRPID	Group ID	Char	Identifier	Perm
6	QSSPID	Sponsor-Defined Identifier	Char	Identifier	Perm
7	QSTESTCD	Question Short Name	Char	Topic	Req
8	QSTEST	Question Name	Char	Synonym Qualifier	Req
9	QSCAT	Category of Question	Char	Grouping Qualifier	Req
10	QSSCAT	Subcategory for Question	Char	Grouping Qualifier	Perm
11	QSORRES	Finding in Original Units	Char	Result Qualifier	Exp
12	QSORRESU	Original Units	Char	Variable Qualifier	Perm
13	QSSTRESC	Character Result/Finding in Std Format	Char	Result Qualifier	Exp
14	QSSTRESN	Numeric Finding in Standard Units	Num	Result Qualifier	Perm
15	QSSTRESU	Standard Units	Char	Variable Qualifier	Perm
16	QSSTAT	Completion Status	Char	Record Qualifier	Perm
17	QSREASND	Reason Not Performed	Char	Record Qualifier	Perm
18	QSBLFL	Baseline Flag	Char	Record Qualifier	Exp
19	QSDRVFL	Derived Flag	Char	Record Qualifier	Perm
20	VISITNUM	Visit Number	Num	Timing	Exp
21	VISIT	Visit Name	Char	Timing	Perm
22	VISITDY	Planned Study Day of Visit	Num	Timing	Perm
23	QSDTC	Date/Time of Finding	Char	Timing	Exp
24	QSDY	Study Day of Finding	Num	Timing	Perm
25	QSTPT	Planned Time Point Name	Char	Timing	Perm
26	QSTPTNUM	Planned Time Point Number	Num	Timing	Perm
27	QSELTM	Planned Elapsed Time from Time Point Ref	Char	Timing	Perm
28	QSTPTREF	Time Point Reference	Char	Timing	Perm
29	QSRFTDTC	Date/Time of Reference Time Point	Char	Timing	Perm
30	QSEVLINT	Evaluation Interval	Char	Timing	Perm

Source: SDTMIG v3.1.2 [6]

The SDTMIG v3.1.3 section 6.3.5.1 page 147 states at the bottom of the QS section the following regarding whether data really belongs in QS or another domain: [7]

> *..Questionnaire data may include, but are not limited to subject reported outcomes and validated or non-validated questionnaires. The QS domain is not intended for use in submitting a set of questions grouped on the CRF for convenience of data capture. Some diaries are vehicles for collecting data*

for a validated questionnaire while others may simply facilitate capture of routine study data. When objective numeric data with result Qualifiers are collected in a questionnaire or diary format, the sponsor should consider whether this data actually belongs in a separate (new or existing) domain. For example, if the subject records the number of caffeinated beverages consumed each day in a diary, this information might be more appropriate for the Substance Use domain. The names of the questionnaires should be described under the variable QSCAT in the questionnaire domain. These could be either abbreviations or longer names, at the sponsor's discretion until controlled terminology is developed. For example, Alzheimer's Disease Assessment Scale (ADAS), SF-36 Health Survey (SF36), Positive and Negative Syndrome Scale (PANSS).

So something like diary data might look like questionnaire data but should not be stored as such. Other times, you might have a sponsor that requests the collection of a standard pain scale at every visit rather than a visit designated for questionnaires in their table of assessments from the Protocol. In that case they might request to store it in a custom domain such as XD or XS. However, if it then appears in the SAP or table mocks along with questionnaire data or within that section of questionnaire data, then that domain will be mapped in the analysis ready questionnaires to ADQSxx (in ADaM the analysis dataset names are extensible to 8 characters).

Per the ADaMIG here is a list of qualifiers not generally used in the QS domain, and also any domain that is used for questionnaire type of data: --POS, --BODSYS, --ORNRLO, --ORNRHI, --STNRLO, --STNRHI, --STRNC, --NRIND, --RESCAT, --XFN, --LOINC, --SPEC, --SPCCND, --LOC, --METHOD, --FAST, --TOX, --TOXGR, --SEV.

Since the QS dataset is often very large, one might want to split it logically by questionnaire, or groups of questionnaires. In that case, the datasets must conform to the splitting conventions. For each new dataset, the QSCAT must be unique across the datasets. The QS datasets may also have SUPPQS for non-standard variables describing the QSORRES. It is this author's opinion to try not to create the SUPPQS unless essential, as it will be very large. Alternately you can add meaningful keys such as QSGRPID and QSDTC to facilitate the merge back.

There were no updates to QS in version the SDTMIG v3.1.3. In the latest version just released, SDTMIG v3.2 additional guidance are provided for the SDTM QS datasets, such as "Please check the CDISC website for the published questionnaire specifications if you haven't already developed your own standards, and to put meaningful question text in the Comments section of the questionnaire". Also in the comments section the version of the questionnaire instrument should be identified. All these comments will make it into the define documentation.

Additional changes in SDTM are provided in SDTMIG v3.2 (page 10) as follows:

Degree of change	Type of change	Details
Major	Removal	Deleted "References" column from Domain specification table.
Major	Update	"Updated QSCAT in Example 2 to map to the QS CT. Updated QSTESTCD to map to the QS CT."
Minor	Format	"Section 4.1.4.10" reference changed to "Section 4: 4.1.10, Representing Time Points".
Minor	Format	"Section 4.1.5.1"" reference changed to ""Section 4: 4.1.5.1, Original And Standardized Results Of Findings And Test Not Done".
Minor	Format	"Section 4.1.5.3.1" reference changed to "Section 4: 4.1.5.3.1, Test Name (--TEST) Greater Than 40 Characters".
Minor	Format	"Section 8.4" reference changed to "Section 8: 8.4, Relating Non-Standard Variables Values To A Parent Domain".
Minor	Update	Added (QSCAT) as controlled terminology for QSCAT in Specification table.
Minor	Update	Added (QSTESTCD) as controlled terminology for QSTESTCD in Specification table.
Minor	Update	Added (QSTEST) as controlled terminology for QSTEST in Specification table.
Minor	Update	Updated CDISC Notes for QSTESTCD so that the examples provided now read "Examples: ADCCMD01, BPR0103.".

Minor	Update	Updated CDISC Notes for QSTEST so that the examples provided now read "Example: Fist, BPR01 - Emotional Withdrawal.".
Minor	Update	Updated CDISC Notes for QSCAT so that the examples provided now read "Examples: ADAS-COG, MDS-UPDRS.".

2. STANDARDS INFORMATION FOR ADAM

Individual questions with a questionnaire might appear to be interesting or of interest, but they do not carry any weight unless added together or converted into a scale of some sort. The source for all this information on scoring questionnaires is first described in the study protocol. It should then be elaborated in the SAP and the algorithms for scoring each scale should be described in detail, even if it is a published metric scale score. Then it is just a matter for the programmer to create the scoring logic in the program that is defined in the SAP. Additional considerations will be described in the SAP as to what to do with missing data, some imputation rules and maximum number of missing values allowed for computation of each score. All this information needs to be repeated by the programmer in the mapping document's Comments or Computational Algorithms section for use in the define.xml.

The ADaM team is also updating the standards for ADQS, changes are mentioned later in the paper. The analysis-ready datasets (ADaM) require any total scale scores and sub-scale scores to be computed and output as separate rows from the original data. Usually the original data is kept as well for traceability (being able to reproduce the same from original data). Like SDTM, the ADaM questionnaire datasets are in vertical structure, with one row per subject, timepoint and question. The analysis datasets reuse variable names in a vertical structure, so additional descriptors need to be added such as flags explaining whether the entire record is derived and also the type of derivation using ADaM-specific variables (PARAMTYP, DTYPE and ANL01FL) which will be discussed in more detail below.

The ADQS dataset is a BDS structure in the ADaMIG. That means it is vertical in nature just as the incoming source domain QS. The ADQS name can be extended to 8 characters when multiple ones are created to capture different questionnaires. For example ADQSMSIS is specific to the MSIS-29 which has complex derivations. In the BDS structure, the value of interest is stored in AVAL if numeric or AVALC if character. All the rest of the variables on that row describe or identify attributes of the record for AVAL. In the case of scale scores it is always stored in AVAL, which requires building a new row or record. The rest of the variables or fields on that row or record are there purely to support and further describe AVAL. That is, USUBJID, AVISIT, APERIOD, all the ADSL variables copied to ADQS and all the variables shown below are purely describing, defining and identifying AVAL for that subject, time point and questionnaire instrument.

Table 4 ADaM Specs, BDS format (NOTE: this is just an excerpt, for full specs see the ADaMIG [1]

Variable Name	Variable Label	Type	Core	Comments
STUDYID	Study Identifier	Char	Req	Identifier
USUBJID	Unique Subject Identifier	Char	Req	SDTM DM.USUBJID
TRTP	Planned Treatment	Char	Req	Record-level Planned Treatment
PARAM	Parameter	Char	Req	The description of the analysis parameter.
PARAMCD	Parameter Code	Char	Req	The short name of the analysis parameter in PARAM.
PARAMN	Parameter (N)	Num	Perm	Must have one-to-one mapping with PARAM and be an integer.
PARAMTYP	Parameter Type	Char	Perm	Whether derived or a function of one or more other parameters.
PARCAT1	Parameter Category 1	Char	Perm	First Categorization for PARAM. Use to identify questionnaire instrument name. Maps from QSCAT
PARCAT1N	Parameter Category 1 (N)	Num	Perm	A numeric representation of PARCAT1.
PARCAT2	Parameter Category 2	Char	Perm	Second categorization for PARAM. Maps from QSSCAT.
PARCAT2N	Parameter Category 2 (N)	Num	Perm	A numeric representation of PARCAT2.

Variable Name	Variable Label	Type	Core	Comments
AVAL	Analysis Value	Num	Req	Numeric analysis value described by PARAM
AVALC	Analysis Value (C)	Char	Req	Character analysis value described by PARAM
ABLFL	Baseline Record Flag	Char	Cond	Character indicator to identify the baseline record for each parameter.
BASE	Baseline Value	Num	Cond	Baseline analysis value.
CHG	Change from Baseline	Num	Perm	Change from baseline Analysis value. Equal to AVAL-BASE.
PCHG	Percent Change from Baseline	Num	Perm	Percent change from baseline analysis value. Equal to ((AVAL-BASE)/BASE)*100.
DTYPE	Derivation Type	Char	Cond	Analysis value derivation method.
PARAMTYP	Parameter Type	Char	Perm	A categorization of PARAM.
ANL01FL	Analysis Flag 01	Char	Perm	Used to identify a record selected in an analysis.
ANL02FL	Analysis Flag 02	Char	Perm	Used to identify a record selected in a separate analysis.

For the derived rows the following variables are of utmost importance: DTYPE, PARAMTYP, ANL01FL, ABLFL, BASE, and CHG. Also timing variables not listed above: AVISIT, AVISITN, APERIOD, and APERIODC.

In addition to the Statistical Analysis Plan (SAP), information about the versions used for each validated questionnaire should be included in the metadata, preferably in the comments column of the define.xml. A logical place to put this comment is attached to the value-level metadata for QSCAT. If more than one version of the questionnaire is used in a study, that is, the questionnaire is updated in the middle of the study; a record can be created in the Supplemental Qualifiers dataset to flag those subjects with the update. Often this occurs to only a few of the questions within the questionnaire, so that QSTESTCD will be flagged.

What to do with 'NOT DONE' or missing data: anything marked 'NOT DONE' is mapped to QSSTAT = 'NOT DONE' and then QSORRES is left blank. QSREASND is then used to store any reasons why the questionnaire was not administered. Further decisions need to be made by the researchers whether these values provide any important information for data management purposes. In the ADAM Datasets these can be removed as they do not add to the analysis. Sometimes the missing measures are imputed with rules such as last observation carried forward (LOCF) or worst observation carried forward (WOCF). These details are captured in the SAP by the study statistician(s).

Questionnaire data is sometimes totaled, averaged, or summed directly on the eCRF in the event that the clinician requires seeing this for their diagnosis. The ADaMIG states:

> *Derived information such as total scores and sub scores, etc., may be stored in the QS domain as derived records with appropriate category/subcategory names (QSSCAT), item names (QSTEST), and results (QSSTRESC, QSSTRESN). Derived records should be flagged by QSDRVFL. Single score measurements or results may go into questionnaire (e.g., APACHE Score, ECOG), but the sponsor should consider if the results should go into a more appropriate domain.*

For example, in one study the Likert Pain scale was shown as an average for the past 7 days purely for the clinician's view. In the case that the scale has been processed in the eCRF, then it is mapped to the QS dataset, but given a different QSTESTCD, such as QSTESTCD = 'MEANLPS'. Other times, the sponsor might want to compute simple scales in SDTM for some quick displays in the patient profile. It could also have been left out from the mapping as it is re-calculated within the ADaM program. It is this author's opinion that this does not replace computing the scale again in the derived dataset, as the SAP usually adds imputation rules and other potential weighting rules that create different value for AVAL than just simple the sum scale. Also, the derivation of a scale score from individual questionnaire items requires creating a new row. Computations can become quite complex, such as reversing the values of questions, multiplying each question by a weight prior to aggregating, and other complex formulas, which do not lend themselves well to the general mapping tasks required for SDTM conversion. Adding to this complexity, if more than one dataset needs to be accessed to create a score, which is a clear indication to leave the derivation for the ADaM datasets.

THEIR DATA, OUR DATA

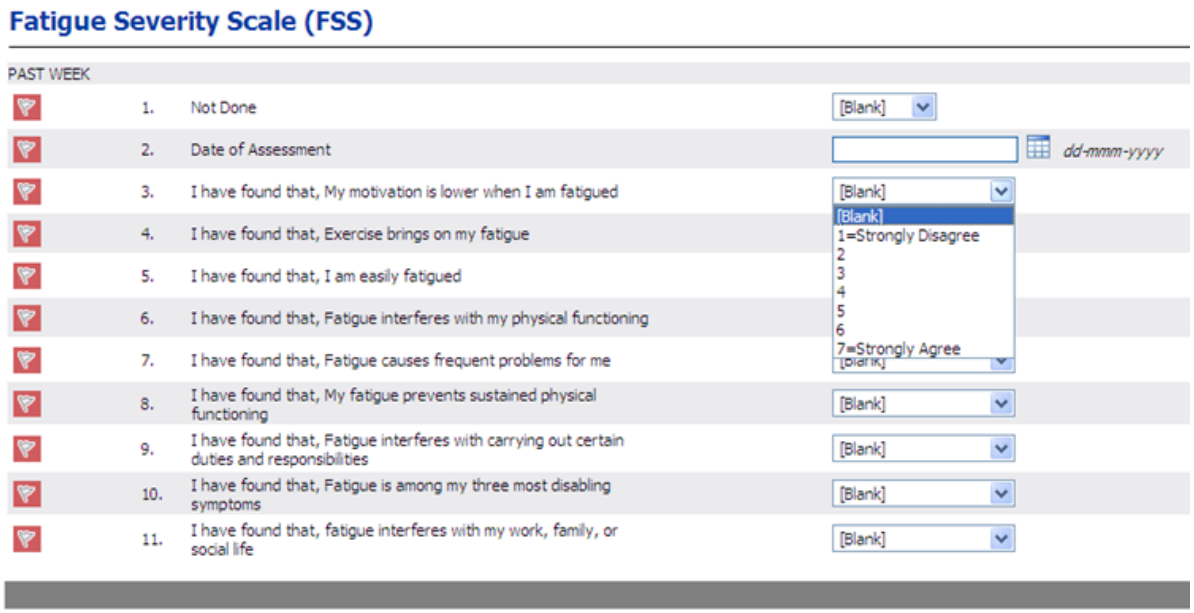
As mentioned above, there are several steps in the conversion process, and often this is broken up into different groups within a company. Let's get beyond Their data, The data, Not our data, to Our data. As we blend traditional departments during the adoption of standards, there is still the resistance to work together. Traditionally, the SDTM and ADaM efforts have been split into two departments. This leads to a 'throwing it over the wall' mentality. By making this one data stream, flowing from the data management and collection departments (using CDASH) to mapping by one group to SDTM and then to ADaM then finally to TFL creation we break down these walls. Below we will describe the entire process using the questionnaire data as the primary example. This will show how the data flows through the process and the efficiencies that are built in by sharing some basic standard concepts, such as shared controlled terminology, and keeping SDTM variables unchanged in ADaM datasets as needed for traceability. Some questionnaires that the authors have come across in various studies and will be used to illustrate the complex issues that are involved in mapping the data from collection to SDTM and then to analysis-ready ADaM datasets, and will be described later in the paper. We will use in our examples the Likert Pain Scale (LPS), the MSIS-29 (MSIS), the TSQM-9 and the Fatigue Severity Scale (FSS). These have a good cross-section of complexity to illustrate SAS code.

COLLECTION OF THE QUESTIONNAIRE DATA

Questionnaire data is generally collected at the study site by an electronic data capture EDC system. Paper Case Report Form are now less common, but if paper is used it also must be entered into a data entry system. It will be assumed that the data has been collected in a variety of ways, whether using Datalabs, Medidata Rave, Oracle Clinical or some other EDC system. According to the CDISC Clinical Data Acquisition Standards Harmonization (CDASH) 18 January 2011 documentation, questionnaire data is currently not mapped to a CDASH standard and no standards are defined. Instead the document states on page 2 that "Proprietary questionnaires and other copyrighted data collection instruments: In order to maintain the validation of these collection instruments, studies that include these questionnaires should present the question and response choices in the manner that these were validated", and "In some cases, this may result in CRF panels that are not conformant with the above items, or with best practices; however, restructuring these questionnaires could invalidate them." However, a new effort is ongoing to produce sample collection forms and standard eCRF recommended annotations one by one for well known and validated questionnaire instruments; future additions to the terminology set is handled via the terminology change request and maintenance process.

Below is an example of the FSS instrument being collected on an electronic case report form (eCRF)

1) DATALABS EDC FSS COLLECTION SCREEN



2) MATCHING CONTROL TERMINOLOGY

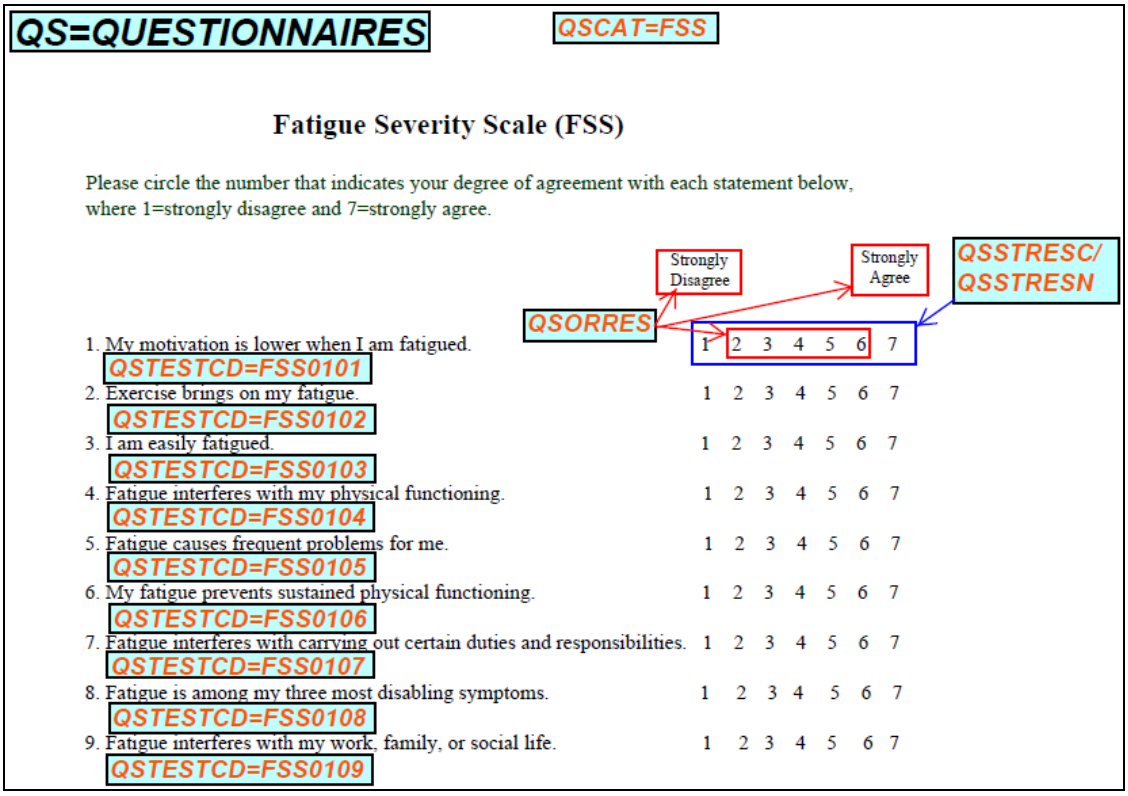
The below data entry screen also has controlled terminology (CT) behind the scenes to populate the answers to the questions. This example is non-CDASH compliant, but defined in CDISC standards document. In the CDISC initiative, this has been documented in the .pdf version of the documentation for the FSS questionnaire [9]

All QSTESTCDs

QSORRES	QSSTRESC	QSSTRESN
Strongly Disagree	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
Strongly Agree	7	7

3) ANNOTATED CRF

Here is the annotated CRF (aCRF), with the annotations for SDTM. This aCRF maps to the SDTM specifications used by the programmer to create conversion programming.



SOME CHALLENGES IN CONVERSION TO SDTM

Compared to other data collection, some unusual steps are required in capturing the QS data. First of all the questions can be very verbose. The full question text string does not usually fit in the standard QSTEST field which is 40 characters long. Therefore it must be truncated or paraphrased to fit. The latest SDTMIG v3.2 (Section 6.3 QS Domain) states that you should use meaningful text in QSTEST and then do one of two things. Section 4 Assumptions for Domain Models, subsection 4.1.5.3.1 gives general rules for populating - -TEST variables:

1) If the full text is available in the eCRF, then link to that page

2) Create a .pdf document with the full text and link it to the define.xml comments section for that variable

For populating QSTESTCD it becomes very important to use good judgment for a naming convention as it is the primary variable to identify that question. Again, if the instrument has been defined by the CDISC committee, then it is advantageous to make use of that work.

Another difference for questionnaires is regarding the rule that QSSTRESC and QSSTRESN must have a 1:1 match. QSORRES is where we store the original answer for the questionnaire, in full alpha-numeric decode. However, we are combining multiple questionnaires into QS, so we cannot have QSSTRESN = 1 and QSSTRESC = 'A little' for one instrument, and QSSTRESN = 1 and QSSTRESC = 'Seldom' for another. Therefore, the way to handle it is to store the full decode in QSORRES. Then for QSSTRESC and QSSTRESN we store the character '1' and the numeric 1 respectively. So the decodes are stored in both QSSTRESC and QSSTRESN; '1'=1 and '2'=2 so there is no 1:1 conflict across measures and even questionnaires. Other answers might be Yes/No. These are mapped with the control terminology of Y or N in QSSTRESC only (see example below). Other required fields are not shown. The CGI, CSSRS-B and CSSRS- x examples are taken from the SDTM website QUESTIONNAIRES standards.

Table 3 SDTM Example of QSORRES, QSSTRESC and QSSTRESN taken from various mapped questionnaires [2]

USUBJID	QSCAT	QSSCAT	QSTESTCD	QSTEST	QSORRES	QSSTRESC	QSSTRESN
ALPHA-001-003	CGI		CGI0101	CGI01-Severity of illness	Severely ill	6	6
ALPHA-001-003	CGI		CGI0102	CGI01-Global improvement	Much worse	6	6
ALPHA-001-003	CGI		CGI0103	CGI01-Efficacy index	Unchanged or worse – None	13	13
ALPHA-001-003	C-SSRS BASELINE	INTENSITY OF IDEATION	CSS0106	CSS01-Most Severe Ideation	2	2	2
ALPHA-001-003	C-SSRS BASELINE	INTENSITY OF IDEATION	CSS0109	CSS01-Most Severe Ideation, Control	Can control thoughts with a lot of difficulty	4	4
ALPHA-001-003	C-SSRS BASELINE	SUICIDAL BEHAVIOUR	CSS0120	CSS01-Suicidal Behavior	Yes	Y	
ALPHA-001-003	C-SSRS BASELINE	SUICIDAL BEHAVIOUR	CSS0121A	CSS01-Most Recent Attempt Date	2010-11-09	2010-11-09	
ALPHA-001-003	UPDRS	II: Activities of Daily Living (for both "on" and "off")	UPD111	UPDRS-Activities: Hygiene	Needs help to shower or bathe; or very slow in hygienic care	2	2

The example above shows the flexibility of a vertical structure (similar to the ADaM BDS structure) to handle all kinds of data. Note several additional types of data not previously discussed. Yes/No response is coded to CT of Y for Yes in QSORRES. The most recent attempt date is stored as a text string. Since it is a date, it is also first converted to ISO8601 standard format from whatever the original collection was. Also note that the first questionnaire, CGI did not require a QSSCAT as the questions were sufficiently identified by QSCAT and QSTESTCD. However, the C-SSRS instrument has sub-categories known as INTENSITY OF IDEATION and SUICIDAL BEHAVIOUR. QSSCAT allows us to further group questions within this instrument into logical groupings. Also, UPDRS has 5 different sub-categories. The QSTESTCD is unique across all of UPDRS, but QSSCAT shows logical grouping. Furthermore, QSTESTCD should be unique across all questionnaire instruments if possible.

QSCAT should always be mapped to the instrument name. QSSCAT can be a subset of that, or a time point. QSTESTCD should always be the question short name, and should be unique across all questionnaire data. QSBLFL can be used, and will be mapped to ABLFL in ADaM but only for the derived rows. QSTEST needs to

capture as much as possible to identify the question, but is usually not used for outputs. If you need the full question captured then it can go into SUPPQS in a custom variable name or names identified by IDVAR and can be up to 200 characters long, or 200 * n in the case of multiple variables.

Baselines for QS data pose unique problems in SDTM domains. The baseline observations should indeed be flagged with the variable QSBLFL for the record closest to dosing. However, because most individual questions are of little meaning until aggregated into a scale or score it is best not to add baseline or change from baseline to any of the rows as it will not be used for the individual questions. Most derived variables are not useful in this domain as they will need to be derived again in analysis ready ADaM datasets, and will usually reside on an entirely new derived record or row. Even in SDTM+ type scenarios (SDTM plus some analysis variables in SUPP domains) it is usually not meaningful to compute these scores and carry a baseline. This is further discussed in the ADaM section, why a straightforward score is not meaningful in SDTM.

IMPLEMENTATION TIPS FOR CREATING SDTM USING SAS®

1) Use of format catalogs in SDTM conversion

As mentioned above, the data is usually collected with a format catalog associated with it in the original eCRF. If that is the case ensure appropriate format is used for the decoding process. The first step is to use PROC CONTENTS in SAS so that we can see what the format names are. Rather than re-typing which would potentially cause a lot of errors, a good programming practice is to use codes and decodes from the original format catalog provided with the data.

This again only affects the QSORRES field as the decodes are stripped out for the QSSTRESC and QSSTRESN fields. These catalogs are delivered along with the raw data during a data extraction from the EDC system. The original data is usually collected in a record or multiple records which represent an eCRF page. These fields then need to be renamed and formatted or de-formatted and finally transposed into a vertical data structure. A handy macro for this process is attached in Appendix A and will be discussed in detail below.

2) Use of lookup tables for QSTEST, QSCAT, QSSCAT

How to map the QSTEST also is a dilemma. Currently QSTEST is limited to 40 characters for QS domain. This text is not important for reporting purposes, other than to serve as identification that you have the correct question. Therefore, if the text is readily available in the raw data it is sufficient to truncate to 40 characters as long as it is unique. This author prefers to specify the QSTESTCD and QSTEST codes within a spreadsheet that is part of the data specifications, which can be read electronically. Additionally QSCAT, QSSCAT and any other repeated fields can be added into the spreadsheet. Then it is a matter of merging by using the shortest variable, which is QSTESTCD at length = 8. The spreadsheet is easy to review by others outside of programming such as the sponsor and also can be re-used when creating the metadata for the study. Additional explanatory columns can also be added which are not read into the merge, but can be used later when creating Define.xml.

Sample spreadsheet structure: (Appendix in the specifications document)

Raw Dataset	Raw Variable	QSTESTCD	QSTEST	QSCAT	QSSCAT	Source document
MMSE1_02	QSMM01	MMSEOTT	Orientation to Time	MINI-MENTAL STATUS EXAMINATION (MMSE)		SPI_Mini-Mental_Status_Examination _2012-04-12.doc
MMSE1_02	QSMM02	MMSEOTP	Orientation to Place	MINI-MENTAL STATUS EXAMINATION (MMSE)		
FSS1_01	QSFSQ01	FSS0101	Motivation Lower when Fatigued	FATIGUE SEVERITY SCALE (FSS)		

3) Guidelines on naming conventions for QSTESTCD

Some care and thought must be given for assigning the question names to QSTESTCD. The contents of this field are limited to 8 characters. They must be unique within the single questionnaire instrument, and also across all the questionnaires stored in QS if we wish to be CDISC-compliant. If you have a new questionnaire not previously

mapped to QSTESTCD and QSTEST, a good starting point is the CDISC website referenced above where the questionnaires have been mapped for you by the CDISC teams which are comprised of industry experts. Otherwise, your company might already have standards in place, and at this time it is allowable to have a variety of naming conventions for the individual questions. Here are some guidelines for naming your own questions to QSTESTCD if they have not already been defined by the sponsor company or by the CDISC team. First, the published question names are a good starting point to select useful and meaningful QSTESTCD's. Instruments such as the Fatigue Severity Scale, has the 9 items named FSS0101 through FSS0109. This type of naming convention ending in a range of numbers is especially useful for post-processing when we get to the analysis datasets (ADaM); more on that later.

4) Converting Raw data to SDTM QS Structure

As mentioned in the introduction, the raw data which will comprise QS comes from multiple eCRF pages and is gathered into multiple raw datasets that represent a page or a section of the page in a database. These small datasets need to be processed and aggregated to create one larger dataset that provides tabulation across both visits and questionnaire instruments. This can be written with a lot of if-then statements, some transpose, or SAS macro code.

Best practices indicate that the code written should maximize use of existing data rather than doing a lot of typing within the program. This promotes re-usability across sponsors. For example, when doing a transpose, use the keywords ID, IDLABEL and IDVAR instead of typing QSTESTCD names and QSTEST long text into the variable. Use formats, so that only the format catalog needs to be addressed when there are slight changes in labels, even slight things like capitalization which is important for output in the Tables requested by the statistician on the study.

Below are two examples, and at the end of the paper we provide a macro in Appendix A.

```
/* DATA STEP EXAMPLE 1*/

data QS_MADRS(keep = studyid domain usubjid qstestcd qstest qscat qsscat
                     qsorres visit visitnum qsdtc qsspid);
    length tname $8;
    set MADRS;

    qscat ="MADRS";
    qstest='MONTGOMERY-ASBERG-DEPRESSION RATING SCALE';
    qsscat=' ';
   /* Set to data row number of original dataset */
    qsspid=strip(put(datarow, best.));

    array qin  {*} $ madrs01 madrs02 madrs03 madrs04 madrs05 madrs06
                     madrs07 madrs08 madrs09 madrs10;

    do xx = 1 to dim(qin);
      call vname(qin{xx}, tname);
      qstestcd=put(upcase(tname), $qstest.);

      if xx eq 1 then qsorres=put(qin{xx}, $madrsa.);
       else if xx eq 2 then qsorres=put(qin{xx}, $madrsb.);
       else if xx eq 3 then qsorres=put(qin{xx}, $madrsc.);
       else if xx eq 4 then qsorres=put(qin{xx}, $madrsd.);
       else if xx eq 5 then qsorres=put(qin{xx}, $madrse.);
       else if xx eq 6 then qsorres=put(qin{xx}, $madrsf.);
       else if xx eq 7 then qsorres=put(qin{xx}, $madrsg.);
       else if xx eq 8 then qsorres=put(qin{xx}, $madrsh.);
       else if xx eq 9 then qsorres=put(qin{xx}, $madrsi.);
       else if xx eq 10 then qsorres=put(qin{xx}, $madrsj.);
      output;
    end;
run;
```

In the above example 1, there is good use of array processing and the output statement creates each variable as a new row, with qsorres as the variable of interest. Note the need for formats a through j to identify all the decodes that

were used in the original study. Not shown, the variables were renamed with the format $qstestcd. Within this program there were many different types of processing depending on the instrument. Some were individual output steps, others were as above. Use of one macro would make this more user-friendly, regardless of questionnaire.

```
/* DATA STEP EXAMPLE 2*/

/* WORK PRODUCTIVITY AND ACTIVITY IMPAIRMENT GENERAL HEALTH QUESTIONNAIRE V2.0
(WPAI-GH)*/

data wpai;
    set raw.wpai;
    attrib misshr misroth workhr length = $200;

    if hrhlth ^= . then misshr = strip( put( hrhlth, 8.)) ||'#' || 'HOURS';
    if hroth ^= . then misroth = strip( put( hroth, 8.)) ||'#' || 'HOURS';
    if hrwrk ^= . then workhr = strip( put( hrwrk, 8.)) ||'#' || 'HOURS';
run;
```

Example 2 above shows how code will differ from questionnaire to questionnaire. In the second example we are mapping numeric responses in the Work Productivity and Activity Impairment questionnaire to fit into character length 200 fields. It would be more convenient to use one macro that handles both, and because the standards group is here to help each other, we are providing additional solutions to make the programmer's task easier. The Appendix A brings is all together and provides a macro that can handle most situations.

5) Description of Conversion Macro for QS (Appendix A).

Use of macros is encouraged by the authors rather than writing individual steps as detailed above. They can be as simple or as complex as you desire. In Appendix A this macro code can to do this conversion within a data step. It is named QS_TRANS.sas, and can also be used for other datasets such as FA since all have same vertical structure.

```
/* WORK PRODUCTIVITY AND ACTIVITY IMPAIRMENT GENERAL HEALTH QUESTIONNAIRE V2.0
(WPAI-GH) */


%qs_recode( in = wpai, out = wpai_1,
            invars   = curempl,
            outvars  = EMPLOYED,
            type     = C,
            qfmt_c   = $ny.,
            remblank = no); /* use this dataset to capture wpaiperf */

%qs_recode( in = wpai, out = wpai_2,
            invars   = hrhlth hroth hrwrk,
            outvars  = MISSHR MISROTH WORKHR,
            qfmt_n   = best12.,
            type     = N,
            remblank = no);

%qs_recode( in = wpai, out = wpai_3,
            invars   = hlthprod hlthdact,
            outvars  = WORKPROD ACTPROD,
            type     = C,
            qfmt_c   = $ztoten.,
            remblank = no);

data wpai_all;
    set wpai_1 wpai_2 wpai_3 (in = c);
    length qsstresu $5 qsstresc $200 qsevlint $100 qsdtc qsorresu  $20 qsstat $8
           qsdtc $19;
    domain = 'QS';
    if anyalpha(qsstresc) = 0 then do;
        qsstresn = input(qsstresc, best.);
    end;
run;
```

The inputs are as follows**: in** is the input dataset, **out** is the output dataset. Multiple datasets need to be created as there are 5 different formats which are called by **qfmt_c** and defined as character by **type** = C. Finally **remblank** is to remove blank questionnaire records that were either skipped or marked 'NOT DONE' if you so desire. The macro is called for each set of different questions. Then for post processing you merely need to concatenate datasets.

Invars is the list of variables from raw to be mapped. **Outvars** are the new variable names. The program uses a trick function to use the outvars to rename the variables but not create new ones.

The SAS function vname (<variable>) returns the name of the variable rather than its contents as the data. So the statement below maps qstestcd to the variable name in the second array.

```
qstestcd = vname(conv_q(i));
```

Arrays are not as popular as they used to be, and they are a very powerful tool to process this kind of data. The following two lines in the macro allow the string of **&invars** and **&outvars** to be processed as individual variables in an iterative loop. Note the (*) which allows SAS to count the number of elements in the array.

```
array raw_q (*) &invars;
array conv_q (*) &outvars;
```

The use of dim as in dimension of array, allows us to call the array from the first to the last element without needing to know how many there are.

```
do i = 1 to dim(conv_q);
```

This macro also allows transposing data for numeric variable as shown in WPAI_2 on the previous page. These then are type = N and the format does not contain the $ sign. The macro requires a format, so if no format is needed, for character we can use a length such as **$50.**, and for numeric we can use a generic format such as **best12.** .

Additional code is presented in the macro that recodes standard items such as various spellings of 'Yes' to 'Y' and 'No' to 'N'.

Finally, output statements are used rather than transpose to output multiple new rows, one for each question or item within that questionnaire, creating the necessary vertical structure. Since the conversion process is tedious and time consuming, the macros minimize mistakes by reuse of as much code as possible. Also, QC becomes simpler, since the macro inputs can be compared against the PROC CONTENTS to ensure the appropriate formats have been applied from the raw data to each group of questions.

CONVERSION TO ADAM FOR ANALYSIS AND TABLE CREATION

When it comes to the ADaM dataset creation the most important document in the study is the SAP, and if the SAP is not available then the Protocol can be used to get started. However, it is essential for the statistician to define all the derivations and analyses in great detail in the SAP and then translate this into Table Shells or Mocks. The statisticians are your friends. They are there to help and to guide the study teams to complete the analyses that are needed to support the submission of the study to the FDA or other regulatory agency. Statisticians fill a unique role in the study as they are there at the beginning when the protocol is being developed. They have a measure of input during the design of the eCRF, to make sure all that is needed will be collected. They also provide ultimate quality control on the Tables, Listings and Figures, as they are ultimately responsible for signing off on the study.

Individual questions within a questionnaire might appear to be interesting or of interest, but they do not carry any weight unless added together or converted into a scale of some sort. The source for all this information on scoring questionnaires is first described in the study protocol. It should then be elaborated in the SAP and the algorithms for scoring each scale should be described in detail, even if it is a published metric scale score. Then it is just a matter for the programmer to create the scoring logic in the program that is defined in the SAP. Additional considerations will be described in the SAP as to what to do with missing data, some imputation rules and maximum number of missing values allowed for computation of each score. All this information needs to be repeated by the programmer in the mapping document's Comments or Computational Algorithms section for use in the define.xml.

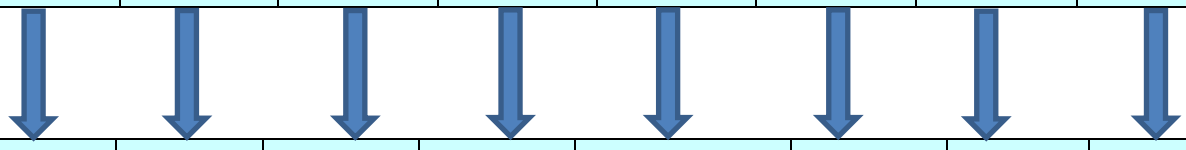
The derivations required for scoring questionnaires usually include grouping questions by category, taking sums, sometimes multiplying by a weighting factor, sometimes reversing the answers as per the instructions, and sometimes imputing for the missing questions as defined in the validated instrument documentation, or in the study-specific SAP. Although many are in published documents, it is recommended to repeat these instructions for each scale in the SAP so that they are readily available to the study team and programming groups. The derived records are usually a combination of many questions and therefore need to be output in the ADAM dataset as a new record.

It is these kinds of derivations that make any derivations in SDTM counter-intuitive. Also baseline records and changes from baseline are better handled in the ADaM datasets.

Below is a graphic representation of how the conversions take place. It is basically a two step process.

Step 1. Map the data from SDTM to ADaM BDS structure. The items in green are renamed to ADaM Standards and the items in black (QSORRES) are carried in as a direct copy for traceability. Note that the mapped variables are not also carried as direct copies.

USUBJID	QSCAT	QSSCAT	QSTESTCD	QSTEST	QSORRES	QSSTRESC	QSSTRESN



USUBJID	PARCAT1	PARCAT2	PARAMCD	PARAM	QSORRES	AVALC	AVAL
ALPHA-001-003	CGI		CGI0101	CGI01-Severity of illness	Severely ill	6	6
ALPHA-001-003	CGI		CGI0102	CGI02-Global improvement	Much worse	6	6
ALPHA-001-003	FSS		FSS0101	My motivation is lower when I am fatigued	Strongly Agree	7	7
ALPHA-001-003	FSS		FSS0102	Exercise brings on my fatigue	4	4	4

Step 2. Add derived rows and derived row flags. The flag ANL01FL indicates that this is the row needed for the first set of analyses. The rows with carrying the original scale scores from each question are not flagged, unless used individually in other analyses. Then they can be flagged with other flags such as ANL02FL.

USUBJID	PARCAT1	PARCAT2	PARAMCD	PARAM	QSORRES	AVALC	AVAL
ALPHA-001-003	FSS		FSS	FSS SCALE SCORE		42	42

DTYPE	PARAMTYP	ABLFL	BASE	CHG	ANL01FL
SUM	DERIVED	Y	42	0	Y

1) SAS CODING TRICKS AND TIPS FOR SMOOTH CONVERSIONS TO ADAM

The scales can be either computed from the vertical structure of QS using PROC SQL steps, or they can be transposed temporarily for processing and then transposed again or output to a new record within the Data Step. The Data Step is preferred after the data is transposed to derive scores. It is a good programming practice to first do

all conversion of the SDTM data into ADaM format prior to creating scale scores. This way we can use AVAL rather than QSORRES, QSSTRESC or QSSTRESN in the transpose statement. There are generally no sums required with AVALC variables. If a score requires summing 'Y' or Yes responses, then a SQL step would be more advantageous.

```
/* TRANSPOSE FOR FURTHER DERIVATIONS */
proc sort data = sdtm.qs (where=(paracat1 = "MSIS-29");
    by usubjid avisitn <other variables you want to keep >;
run;

proc transpose data = qs (where = (parcat1 = "MSIS-29")
    out = tr_qs (drop=_name_ _label_ );
    by usubjid avisitn <other variables you want to keep >;
    id paramcd;
    idlabel param;
    var aval;
run;
```

First, transpose the data back to a one row per subject per visit using transpose shown above, then do a sum in the subsequent data step. Then use code on the below to denormalize the data to compute derived variables. Most scales consist of 2 or more questions added or weighted or multiplied together in some form. Additionally some

```
/* ADaM EXAMPLE CODE FOR DERIVATIONS*/
data qsmsis (drop = msis001-msis029 );
    set qs2;
    length param $40 ablfl anl01fl imputefl $1 paramtyp dtype $20 paramcd $8;

    miss1 = nmiss(of msis001-msis020);
    miss2 = nmiss(of msis021-msis029);
    mean1 = mean(of msis001--msis020);
    mean2 = mean(of msis021--msis029);
    anl01fl = 'Y';

    *** CREATE PHYSICAL WELL-BEING, PARAMCD = 'PHYSWB';
    if miss1 <= 4 then do;
        paramcd = 'PHYSWB';
        param = 'PHYSICAL WELL-BEING SCORE';
        AVAL = sum(of msis001--msis020) + miss1 * mean1;
        paramtyp = 'DERIVED';
        dtype = 'SCORE';
        if miss1 gt 0 then imputefl = 'Y';
        if qsblfl='Y' then ablfl = 'Y';
        output;
    end;
    *** CREATE PSYCOLOGICAL WELL-BEING, PARAMCD = 'PSYCWB';
    if miss2 <= 2 then do;
        paramcd = 'PSYCWB';
        param = 'PSYCHOLOGICAL WELL-BEING SCORE';  ;
        AVAL = sum(of msis021--msis029) + miss2 * mean2;
        paramtyp = 'DERIVED';
        dtype = 'SCORE';
        if miss1 gt 0 then imputefl = 'Y';
        if qsblfl = 'Y' then ablfl = 'Y';
        output;
    end;
run;
```

questions need to be reversed in order to go in the same direction as the rest. In that case, some handy code is for an item with 5 answers 1-5, see example on previous page. This item is then added to the rest using the new variable name.

Sometimes imputations are required, such as when 2 out of 9 items are missing and the means needs to be added. Some logic in the metrics indicates imputations rules and also maximum amount of elements allowed to be missing. For example, if 20% or more of items are missing then set to missing, otherwise impute the missing elements to the

mean of the remaining items and add together.

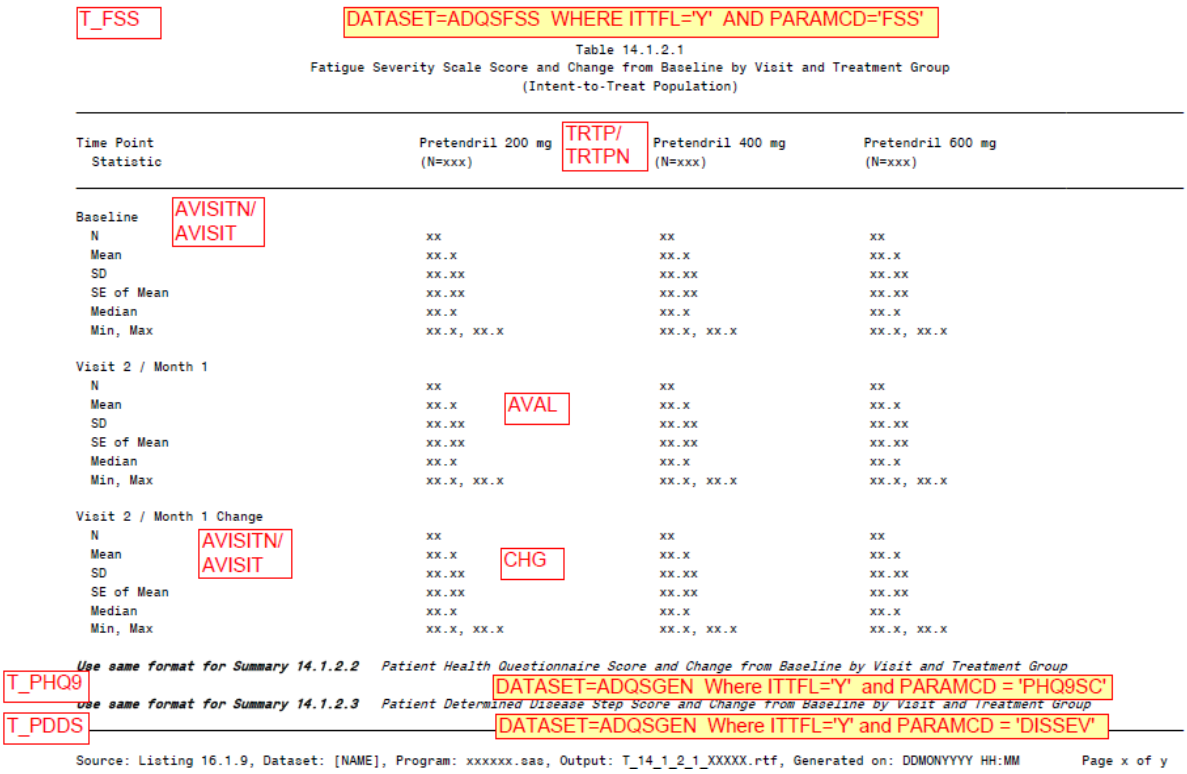
Finally a record is output for each new scale using Output statements. A summed value for AVAL is created and the originally transposed values are dropped.

2) ADDITIONAL VARIABLES NEEDED TO IDENTIFY ADAM DERIVED ROWS

There are additional variables described in the ADaMIG to categorize or identify derived rows for the questionnaire. These are DTYPE and PARAMTYP. The derived row should also be marked with ANL01FL= 'Y' so that it can be easily selected in the table programs, leaving the original questions behind in the analysis dataset. The original questions are only carried forward to the ADaM dataset for traceability. At this time in the derived row, we also compute the variables BASE, CHG, and PCTCHG.

EASY TABLE PROGRAMMING USING ANALYSIS-READY ADAM DATASETS

The ADQSxx datasets are in vertical BDS structure; this lends itself well to creating one program that will basically satisfy multiple Table Shells. This allows us to use repeat-table programming techniques on repeat tables and sometimes even on similar primary tables. For example, below is a standard table shell with means other basic statistics, then change from baseline statistics. Interestingly this shell looks similar to lab tables, vital signs, and other vertical table structures. This ADaM annotated table shell also provides clarity to the statisticians and programmers on exactly what variables to create and use during programming and validation. It is recommended to make this part of your normal process.



Using a simple macro wrapper around the table program body, a multitude of similar tables can be created. Here are some of the commonalities: AVAL, AVISITN, AVISIT, TRTA, TRTAN, TRTP, and TRTPN. These are used in the table program body to generate statistics from the correct visit and corresponding treatment group.

Here are some examples values for variables which can be passed as parameters including where clauses:

Dataset name: ADQS, ADQSGEN, ADQSMSIS, ADQSSF12, ADLB, ADVS
Population: SAFFL, FASFL, ITTFL and PPROTFL
PARAMCD: (where = (paramcd = 'xxxxxxxx');
AVISITN : (where=(avisitn in(1,2,4,6,8)));
APERIODN: (where =(aperiodn = 1));

Additional formatting can be passed with parameters as well, such as number of decimal places to show for each table. Again, this can be handled in the wrapper program, or if too complex, then another macro for statistics within the body of the program might pass these parameters on.

CONCLUSION

In order to effectively implement standards in an organization, different departments need to work together towards the common goal. This goal is providing the research in a readable format for the regulatory agencies to make their final decisions on new potentially life-saving compounds. Although this paper was focused on one small segment of standards, questionnaire data, the paper shows how data flows from collection to TFLs. The process described in the paper will also maximize the traceability of the collected data to eventual analyses for the submission.

The paper walked through the challenges of processing questionnaire data. First, there are many different forms of questionnaire data that are collected, mixing character and numeric data in the same variable. Techniques for representing that in a compliant SDTM form were discussed. Second, when combining multiple questionnaire types consistently across dissimilar questions and answers planning is needed with the use of SDTM variables QSORRES, QSSTRESC, QSSTRESN, QSTEST, QSTESTCD, QSCAT, and QSSCAT. Third, code was provided to demonstrate how to correctly process the Raw collected data into the vertical QS domain structure. The challenge at this point is to present as collected and not derive values unless absolutely needed by the investigator at the site to determine patient safety. Fourth, when creating ADaM datasets the scale scores must be derived as per the instrument and missing elements may or may not be imputed as per very specific rules that are detailed in the SAP. The CDISC organization is currently discussing where the derived scores should reside. Programming code was demonstrated to show how to transpose data to easy derivation creation. ADaM is naturally well-matched to add additional derived rows. Thoughts were presented around when to divide up a set of questionnaire datasets into multiple ADaM datasets and which ones to stay combined. Finally, annotating a TFL shell will create clarity to the statisticians and programmers on exactly what variables to create and use during programming and validation. It is recommended to do this at the beginning of a project, potentially even before EDC design and SDTM conversion, when and if possible.

We hope that this paper will give you some of the tools needed to work through the complex mapping and derivations needed for questionnaire data.

REFERENCES

[1] Analysis Data Model (ADaM) and Implementation Guide, v.2.1, IG v.1.0 (2009) , Available at www.cdisc.org/adam

[2] Pasta, David J.; Suhr, Diana. “Creating Scales from Questionnaires: PROC VARCLUS vs. Factor Analysis”, Proceedings of the SUGI 29 Conference, available at: http://www2.sas.com/proceedings/sugi29/205-29.pdf

[3] http://www.cancer.gov/cancertopics/cancerlibrary/terminologyresources/cdisc

[4] www.cdisc.org

[5] CDISC Questionnaire Supplements http://www.cdisc.org/stuff/contentmgr/files/0/f42fd061dc3854fbeba6c982920dc603/misc/qs_documentation_table_2013_12_31_overall.pdf

[6] Study Data Tabulation Model (SDTM) and Implementation Guide, v.1.2, IG v.3.1.2, Available at www.cdisc.org/sdtm

[7] Study Data Tabulation Model (SDTM) and Implementation Guide, v.1.3, IG v.3.1.3, Available at www.cdisc.org/sdtm

[8] Study Data Tabulation Model (SDTM) and Implementation Guide, v.1.3, IG v.3.2, Available at www.cdisc.org/sdtm

[9] CDISC Questionnaire supplements, FSS document ‘FSS v2. Annotated CRF’

ACKNOWLEDGEMENTS

We would like to acknowledge David Fielding for his review of this paper, and my co-author for adding direction and heart to the paper, and for the great pictures. From Terek Peterson: I would like to thank Karin LaPann for her motivation to break down the barriers between data collection, data submission, and data analysis. It has been a privilege to work with someone that can see all aspects of the collection of accurate data and the important analysis of that data; a rare ability when we have so many existing silos in industry between groups. Her simplistic analyses of “data” emergencies are fun to watch when so many constraints are present with our industry including regulation, quality, timeliness, and patient safety! Go Karin!

CONTACT INFORMATION

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

Name: Karin LaPann
Principal CDISC Standards Consultant
Enterprise: PRA International
Address: Regional
Work Phone: (434) 951-3436
E-mail: LapannKarin@praintl.com
Twitter: Karin LaPann@CDISCatPRA

Name: Terek Peterson, MBA
Senior Director, Global Standards Strategies
Enterprise: PRA International
Address: 630 Dresher Road
City, State, ZIP: Horsham, PA 19044
Work Phone: (215) 444-8613
Fax: (215) 444-8601
E-mail: PetersonTerek@praintl.com
Web: www.praintl.com

APPENDIX A – SAMPLE CONVERSION MACRO FOR QS (RECODE_QS.SAS)

```
%macro qs_recode in       = ,
                 out      = ,
                 qfmt_c   = ,  /* use only if character */
                 qfmt_n   = ,  /* use only if numeric */
                 type     = ,  /* use C for char and N for numeric */
                 invars   = ,  /* list variable names in input dataset */
                 outvars  = ,  /* list variable names in output dataset,
                                  must have same number elements as above */
                 pref     = ,  /* for QSSCAT */
                 remblank = ); /*  to remove any obs that have QSORRES missing */
   options missing = ' ';

   data &out;
      length qstestcd $8 qsorres qsstresc $200  &outvars $100;
       set &in;

      %if &pref gt '' %then %do;
         length qsgrpid $25 ;
         qsgrpid = upcase(scan(&pref,1,":"));
      %end;

       array raw_q (*) &invars;
       array conv_q (*) &outvars;

       do i = 1 to dim(conv_q);

           qstestcd = vname(conv_q(i));
           qstestcd = upcase(qstestcd);

         %if &type=C %then %do;
           qsorres = left(put(raw_q(i),&qfmt_c.));
           if qsorres in ('YES' 'Yes' 'yes') then do;
               qsorres = 'Y';
               qsstresc = 'Y';
           end;
           else if qsorres in ('NO' 'No' 'no') then do;
               qsorres = 'N';
               qsstresc = 'N';
           end;
           else qsstresc = left(put(raw_q(i), $200.));;
       %end;

       %else %if %upcase(&type)= N %then %do;
           qsorres = left(put(raw_q(i),&qfmt_n.));
           qsstresc = qsorres;
           qsstresn = raw_q(i);
       %end;
           output;
     end;
  run;

   %if %upcase(&remblank) = YES %then %do;
       data &out;
       set &out;
          if qsorres ne '' then output;
       run;
   %end;
%mend qs_recode;
```