Tracing data elements through a standard data flow

Alistair Dootson

Dootsonic, Manchester, UK

Ever wondered where a data point came from? Ever been looking through some Tables, Figures and Listings (TFLs) and thought 'I wonder how this number came to be here?'. Chances are you are wondering about the traceability through your data collection, management, and reporting process. Where do you start tracing the data? Where do you finish? How am I going to trace this data effectively? This paper looks into the 'breadcrumb trail' through the data management and reporting cycle of clinical trials. Is there a simple solution out there to manage the traceability of the data in your trial? What will you do when the FDA come knocking and want to see the links through the steps in your process? These questions will soon be answered.

Keywords: CDISC, Traceability, SDTM, ADaM, CDASH, Metadata, ETL, TFLs, Data flow

Introduction

Metadata, CDISC® (PRM, CDASH, SDTM, ADaM), Tables, Figures and Listings (TFLs), Extract, Transform & Load (ETL), and Traceability seem to be some of the buzz words around at the moment. There have been previous papers written about tracing between SDTM and ADaM. There have been previous papers written about managing metadata and both topics are valid. But why stop there? Surely we need to know about where the data that feeds SDTM came from — data do not tend to be collected in those standard variables. When looking at a TFL, is there an easy way of finding out the roots of a data point?

- Metadata — data about data. Information on the data column's names, labels, lengths, type, and so on. Why is it so important?
- Standards — CDISC springs to mind as a whole. Can the siloed groups pull themselves together and make the standards flow?
- ETL — We will examine the use of some tools through this process and discuss their benefits and downsides. ETL tools are springing up as part of this process, and this paper will examine their roles.
- TFLs — Is there going to be a revolution? What part do they play in the traceability of data elements?
- Traceability — 'Traceability refers to the completeness of the information about every step in a process chain.

The formal definition: Traceability is the ability to chronologically interrelate uniquely identifiable entities in a way that is verifiable.[1] In the past, traceability was something associated with software development — how did a developer get from the user requirements through the specifications to the testing documents. Now it also refers to how we follow a data element through the process of collection, management and reporting.

As we examine the pieces of the puzzle and throw down the breadcrumbs along the way, will we be able to find our way back?

The paper will look at a simple example of the traceability of AGE for Demographics, but consider the implications for a more complex data element.

Metadata

What makes up data? A dataset as we classically know it in SAS®, or a table (from SQL) forms the basis of the data as they are a container in which to place data elements and rows of captured data. There is also a shell of information that surrounds the guts of the observations. This is known as the metadata. Is it necessary to know all the information in a dataset in order to manipulate the actual data? If you know the make-up of data, can you process all you need to? If you can get the information about the data elements in a flow, can you build a flow using just the element names, types, lengths, and descriptions or do you need actual data to build it? How can you obtain metadata easily? One way of dealing solely with metadata is by using an ETL tool such as SAS Clinical DI Studio® (see Fig. 2 for an example of metadata view in CDI). The metadata is registered in the tool and although a pointer is set up to where the actual associated data resides, or will reside, the metadata can be used to produce a data flow.

Correspondence to: Alistair Dootson, Dootsonic Ltd, Manchester, UK. Email: Alistair@dootsonic.com

Published by Maney
DOI 10.1179/175709311X13166801334352

When looking at the standards a little later, they have been defined with specific implementation guides that reference their metadata, making it easy to specify the shells of data at each stage that would be of use. CDASH, SDTM and, to a lesser extent, ADaM have well defined structures that can be used to define process flows without the need for data — especially if relationships between the data elements is defined.

The alternative to using an ETL tool is to perform metadata management manually. One such way of doing this is by using the SAS Code Analyzer procedure.

The example snippet of code to add in the code analyzer is as in SAS Code 1:

```
proc scaproc;
 record "C:\Documents and Settings\My
Documents\phuse\record.txt" attr;
run;
… code …
proc scaproc;
write;
run;
```

SAS Code 1

The record option specifies the file to write to and the attr option specifies to capture the data element attributes. The results of the analyzer are written to the file 'C:\Documents and Settings\My Documents\phuse\record.txt' and an example portion for one of the data steps in the code used later on is given in SAS Log 1:

```
/* JOBSPLIT: DATASET INPUT SEQ
WORK.DEMO.DATA */
/* JOBSPLIT: LIBNAME WORK V9
' C:\DOCUME~1\LOCALS~1\Temp\SAS
Temporary Files\_TD5868' */
/* JOBSPLIT: DATASET OUTPUT SEQ
WORK.DEMO1.DATA */
/* JOBSPLIT: LIBNAME WORK V9
' C:\DOCUME~1\LOCALS~1\Temp\SAS
Temporary Files\_TD5868' */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:subjid TYPE:NUMERIC LENGTH:8
LABEL:Subject ID FORMAT:BEST12.
INFORMAT:F12. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:siteid TYPE:CHARACTER
LENGTH:2 LABEL:Center or Site ID
FORMAT:$F2. INFORMAT:$F2. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:invname TYPE:CHARACTER
LENGTH:4 LABEL:Investigator Name
FORMAT:$F4. INFORMAT:$F4. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:WEIGHT TYPE:NUMERIC LENGTH:8
LABEL:Weight in Kilograms
FORMAT:BEST12. INFORMAT:F12. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:HEIGHT TYPE:NUMERIC LENGTH:8
LABEL:Height in Centimeters
FORMAT:BEST12. INFORMAT:F12. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:studyid TYPE:CHARACTER
LENGTH:8 LABEL:Study ID FORMAT:$F8.
INFORMAT:$F8. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:country TYPE:CHARACTER
LENGTH:13 LABEL:Country FORMAT:$F13.
INFORMAT:$F13. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:birthdt TYPE:NUMERIC
LENGTH:8 LABEL:Date of Birth
FORMAT:DATE9. INFORMAT:F9. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:sex TYPE:CHARACTER LENGTH:6
LABEL:Sex FORMAT:$F6. INFORMAT:$F6. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:race TYPE:CHARACTER LENGTH:9
LABEL:Race FORMAT:$F9. INFORMAT:$F9.
*/
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:age TYPE:NUMERIC LENGTH:8
LABEL:Age in Years at Baseline
FORMAT:BEST12. INFORMAT:F12. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:TRTGRP TYPE:CHARACTER
LENGTH:8 LABEL:Treatment Group
FORMAT:$F8. INFORMAT:$F8. */
/* JOBSPLIT: ATTR WORK.DEMO.DATA INPUT
VARIABLE:trtcd TYPE:NUMERIC LENGTH:8
LABEL:Treatment Code FORMAT:BEST12.
INFORMAT:F12. */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:siteid TYPE:CHARACTER
LENGTH:2 LABEL:Center or Site ID
FORMAT:$F2. INFORMAT:$F2. */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:invname
TYPE:CHARACTER LENGTH:4
LABEL:Investigator Name FORMAT:$F4.
INFORMAT:$F4. */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:WEIGHT TYPE:NUMERIC
LENGTH:8 LABEL:Weight in Kilograms
FORMAT:BEST12. INFORMAT:F12. */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:HEIGHT TYPE:NUMERIC
LENGTH:8 LABEL:Height in Centimeters
FORMAT:BEST12. INFORMAT:F12. */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:studyid
TYPE:CHARACTER LENGTH:8 LABEL:Study ID
FORMAT:$F8. INFORMAT:$F8. */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:birthdt TYPE:NUMERIC
LENGTH:8 LABEL:Date of Birth
FORMAT:DATE9. INFORMAT:F9. */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:race TYPE:CHARACTER
LENGTH:9 LABEL:Race FORMAT:$F9.
INFORMAT:$F9. */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:age TYPE:NUMERIC
LENGTH:8 LABEL:Age in Years at Baseline
FORMAT:BEST12. INFORMAT:F12. */
```

```
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:SUBJID1 TYPE:NUMERIC
LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:COUNTRY1
TYPE:CHARACTER LENGTH:13 LABEL:
FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:TRTSTDT TYPE:NUMERIC
LENGTH:8 LABEL: FORMAT:DATETIME.
INFORMAT: */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:TRTENDT TYPE:NUMERIC
LENGTH:8 LABEL: FORMAT:DATETIME.
INFORMAT: */
/* JOBSPLIT: ATTR WORK.DEMO1.DATA
OUTPUT VARIABLE:SEX1 TYPE:NUMERIC
LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: SYMBOL GET SYS_IOUSEEE */
/* JOBSPLIT: ELAPSED 14 */
/* JOBSPLIT: PROCNAME DATASTEP */
/* JOBSPLIT: STEP SOURCE FOLLOWS */
data DEMO1;
 set DEMO (obs=20);
 SUBJID1=SUBJID;
 COUNTRY1=COUNTRY;
 TRTSTDT=(BIRTHDT+120+(AGE*365.25
 ))*60*60*24;
 TRTENDT=(BIRTHDT+122+(AGE*365.25
 ))*60*60*24;
 if SEX='Male' then SEX1=1;
 else if SEX='Female' then SEX1=2;
 format TRTSTDT TRTENDT datetime.;
 drop SUBJID SEX TRTCD TRTGRP COUNTRY;
run;
```

SAS Log 1

As we can see from the results in SAS Log 1, the metadata attributes are given and can be used to list out the properties of the elements; however, there is no reference to the calculations of the new data elements. So traceability from old to new is still not possible. This procedure does have some benefits, and it is similar to a proc content output, but without having to read datasets — so just analyzing code. This would come in useful when just dealing with code to attain metadata.

We can use metadata only to build a data flow, but in order to test that flow we need to put actual data through the process.

Standards

It is a great thing that companies have standards in place to capture their data flow. It makes life easier to reuse programs and internal systems and addresses one portion of the biggest challenges for industry, getting the drug to market as quickly as possible. Great, now convert it to SDTM! Isn't that what the authorities want to see? The challenge a lot of companies have is seeing beyond their own scope of vision. Now, break out of the bubble. What could speed up the process for a company? Deliver the data to the authorities in a fashion that they want to receive it in! If you are one company, then your standard is great for you; if you are the authorities and you have hundreds of standards to wade through, not so useful. So, sooner or later, the authorities will require it in a certain format.

Consider now, CDISC, which has been around for a while and is progressively becoming larger in the minds of companies. The contributors at CDISC do so out of their own time and contribute a great deal and that is hugely appreciated. The ideas are sound, but perhaps they have been siloed into the individual portions. Is there a great deal of cohesion between the standards — CDASH, SDTM, ADaM, etc.? Once all the pieces are put together, we will be able to easily trace from one end to the other.

'CDISC SHARE is a global, accessible, electronic library, which through advanced technology, enables precise and standardized data element definitions that can be used within applications and across studies to improve biomedical research and its link with healthcare'.[2] This new group at CDISC seems like they will be adding a lot of value to the traceability of data elements as they progress through their work and to the targeted completion in 2012. Not only will it provide consistency between the existing standards (CDASH, SDTM) but that would immediately allow for easier tracing of data elements. However, their work does not cover actual traceability of elements, more consistency of elements across standards.

ETL

ETL tools can define the data flow using metadata only. This defines how a variable or column is manipulated from its source through to final analysis data, one step from the tables. This approach can offer improvements in reusability when combining with strong standards. Clinical Data Integration Studio (CDI) is a SAS ETL (Extract/Transformation/Load) tool with a point-and-click interface that allows users to generate SAS Foundation code. The code generated is for data transformations or jobs and achieved by designing a visual process flow composed of distinct 'nodes' and 'connectors' that each define unique process parameters and represent individual points of data integration and transformation. Information about the process flow, the individual nodes, the data, and the libraries they touch is all stored on the metadata server.

Working in a highly regulated industry still poses a few issues when it comes to ETL tools. It is perhaps difficult to trace and audit what happens with processes in the tools themselves. However, it is possible that change management features can be used within CDI to provide some control over the processes and audit trail over who has changed what,

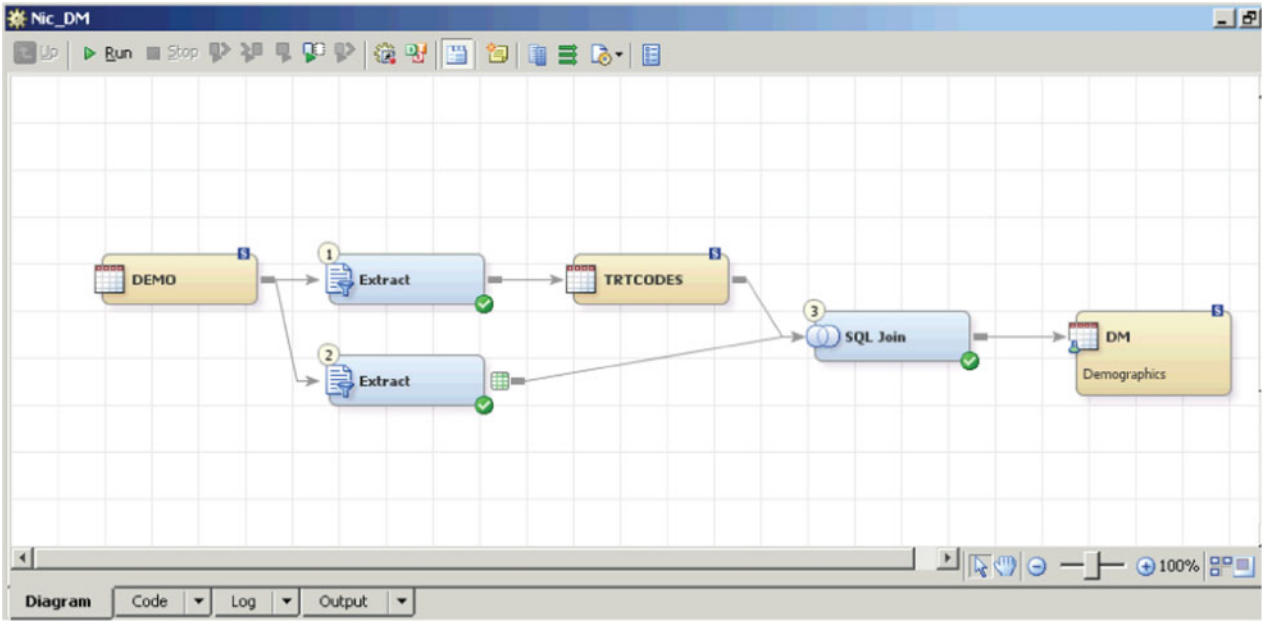

Figure 1 DM domain Job in CDI

but it is by no means all encompassing and, if necessary, a version control tool such as SAS Drug Development® can be used to provide a 'wrapper' around the code generated by the tool for regulatory compliance.

Example

Data are captured in an electronic data capture system and extracted to a raw SAS dataset. How can we follow those data points through the flow to the TFL?

Clinical DI Studio mapping

There are several ETL tools on the market that would be able to produce the mapping from the raw data to analysis. With sufficient standards in place, it would allow for easy mapping between standard variables being read in and transformed to the target analysis tables. One such example is SAS Clinical DI Studio which benefits from the use of SAS — a skill that most programmers in the pharmaceutical industry are familiar with, interaction with many types of data, a clinical component to manage standards — especially SDTM, and an easy to use interface.

The process to set up a metadata mapping for a study in Clinical DI Studio is as follows:

- assume that the SDTM standard is imported into the system;
- set up a clinical component of 'Study' as a container to hold the mappings;
- register the source (or Raw) data and the target SDTM datasets from the imported standard;
- create a mapping Job for each target domain (Fig. 1).

Here is where the standard structures and metadata come in to play. The inputs could be variable, different columns of data captured depending on the trial, but the output domains should be the same. Again, there could be variations due to therapeutic area and compounds, but the set-up should account for that. Reusability comes in the form of similar jobs but perhaps with different inputs. An existing job can be copied and the input data switched out, thus giving a large percentage of ready prepared work.

The job above is a simple example using the DEMO data as input. The **extract** transformations are used to select data from the input data. In this case, for effect, the treatment codes are extracted to a new table before being merged back in the **SQL Join** transformation.

There are no complex derivations and the data come from only one source, which is why this DM mapping is simple. Other Jobs can be much more complex with many inputs and complex derivations. The main source of the work here is performed within the SQL Join as we can see in Fig. 2.

The expression column contains the SAS code associated with the manipulation of the columns. This is just basic SAS code put into the expression.

Tracing columns

Clinical DI Studio allows for impact analysis — or tracing — of both datasets and columns. The dataset or column can be traced throughout the repository and this shows where it is used. In the example in Fig. 3, the DM domain can be seen to trace back to the DEMO dataset with a SQL Join and two extract transformations present.

By clicking on the DM Domain being analyzed - shown in Fig 4. and Fig. 5, the column analysis can be found. In this case, AGE can be traced back to the original data where it was located, and the user can see that it was derived along the way. The drawback here is that there is no further information. To delve deeper, the user must be in the system and drill into the job itself to see the derivation / manipulation.

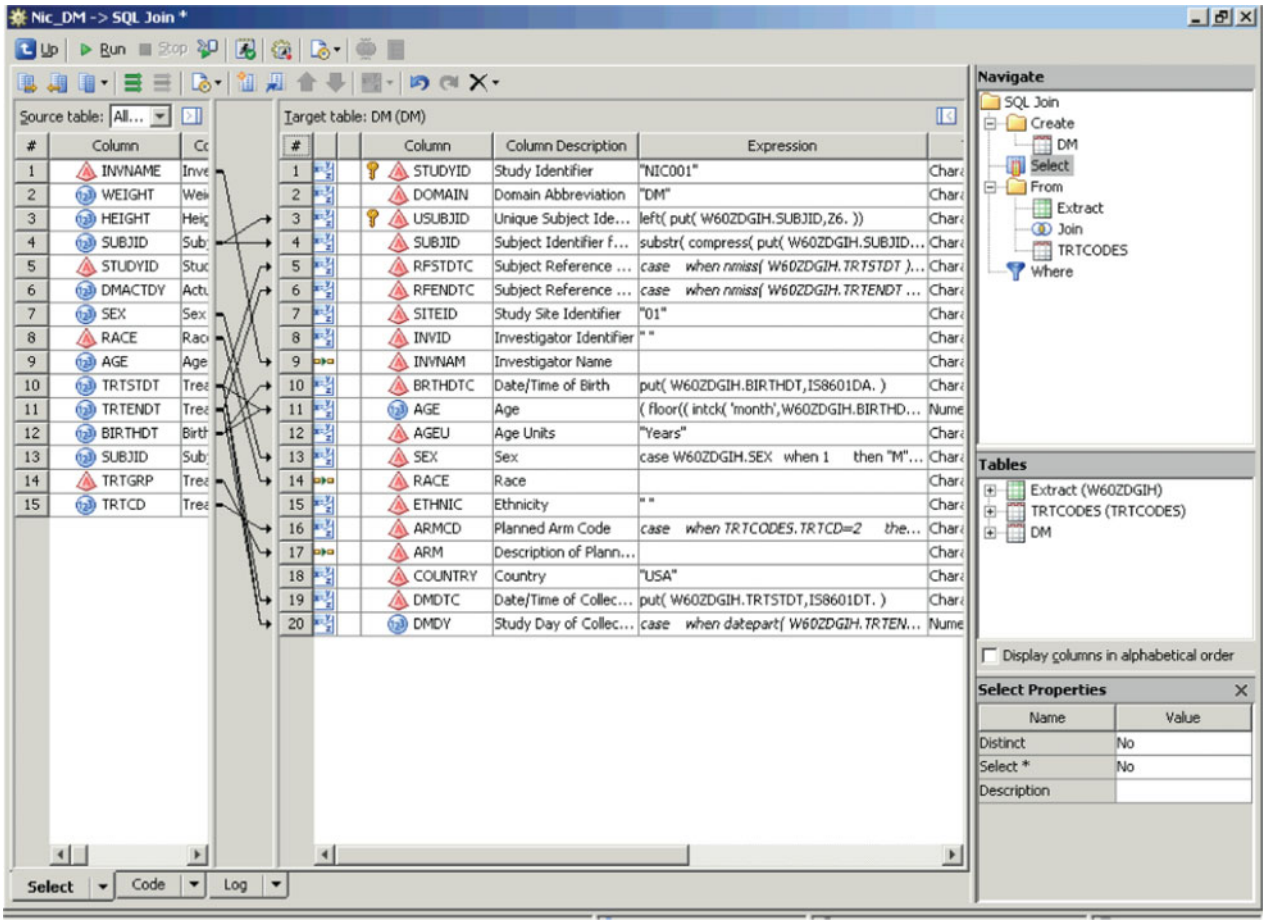

Figure 2 SQL Join

This covers each step quite nicely. But it only covers these same metadata variables within the metadata repository. Even to add the next step of creating ADaM in CDI would see more traceability back to the origins, but you cannot trace the full flow from the TFLs. So, how can we see, in one easy step, how a variable is created?

Let's consider trying the same thing as above but in Base SAS. Code is written from scratch — or copied and pasted…. The basic coding is the same, with expressions in SAS Clinical DI Studio replaced by basic manipulation in SAS. The code is listed in SAS Code 2.

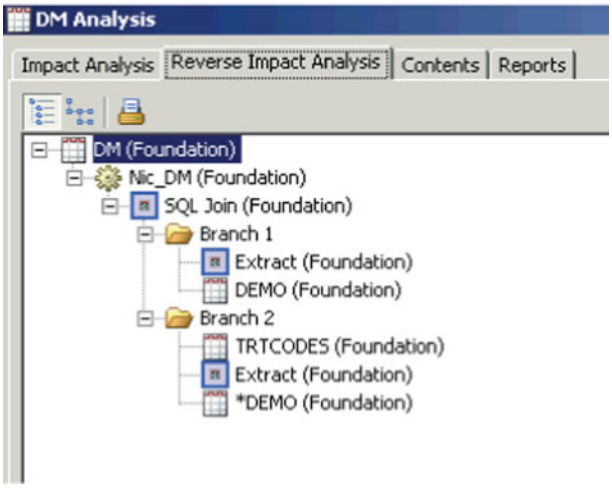

Figure 3 Reverse impact analysis of DM domain

```
*****************************************
Dootsonic Ltd : Programming
PROGRAM NAME : SDTM_DM.sas
PROJECT NAME : SDTM Creation
DESCRIPTION : Code to create DM SDTM
standard dataset
SOFTWARE/VERSION# : SAS 9.2
INFRASTRUCTURE : Windows
VALIDATION LEVEL : 1
OUTPUTS : SDTM DM Dataset
REGULATORY STATUS : GCP
-------------------------------------
Requirements: Create SDTM Dataset
according to IG
-------------------------------------
Ver # Author & Program History
Description
Peer Reviewer
-------------------------------------
001 Ali Dootson Draft version of the
program
*****************************************;
*** Set Up Libnames ***;
libname RAWDATA "C:\Documents and
Settings\My Documents\phuse";
*** Pull in data required for DM Domain
***;
proc SORT data=RAWDATA.DEMO out=DEMO;
by SUBJID;
```

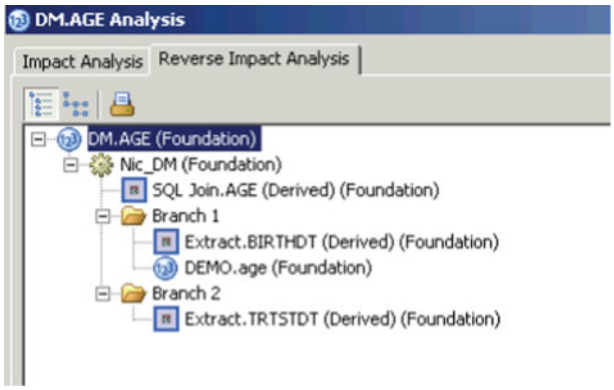

Figure 4 Reverse impact analysis for AGE

```
run;
proc SORT data=RAWDATA.NIC1_TRTCODES
out=TRTCODES;
by SUBJID;
run;
data DEMO1;
 set DEMO (obs=20);
 SUBJID1=SUBJID;
 COUNTRY1=COUNTRY;
 TRTSTDT=(BIRTHDT+120+(AGE*365.25
 ))*60*60*24;
 TRTENDT=(BIRTHDT+122+(AGE*365.25
 ))*60*60*24;
 if SEX='Male' then SEX1=1;
 else if SEX='Female' then SEX1=1;
 format TRTSTDT TRTENDT datetime.;
 drop SUBJID SEX TRTCD TRTGRP COUNTRY;
run;
*** Merge the raw data ***;
data TRTCODES;
set TRTCODES (obs=20);
by SUBJID;
SUBJID1=SUBJID;
drop SUBJID;
run;
*** Create the DM domain ***;
data DM;
attrib STUDYID length = $40 label =
'Study Identifier';
attrib DOMAIN length = $8 label =
'Domain Abbreviation';
attrib USUBJID length = $40 label =
'Unique Subject Identifier';
attrib SUBJID length = $40 label =
'Subject Identifier for the Study';
attrib RFSTDTC length = $64 label =
'Subject Reference Start Date/Time';
attrib RFENDTC length = $64 label =
'Subject Reference End Date/Time';
attrib SITEID length = $40 label =
'Study Site Identifier';
attrib INVID length = $40 label =
'Investigator Identifier';
attrib INVNAM length = $40 label =
'Investigator Name';
attrib BRTHDTC length = $64 label =
'Date/Time of Birth';
attrib AGE length = 8 label = 'Age in AGEU
at RFSTDTC';
attrib AGEU length = $10 label = 'Age
Units';
attrib SEX length = $2 label = 'Sex';
attrib RACE length = $40 label =
'Race';
attrib ETHNIC length = $40 label =
'Ethnicity';
attrib ARMCD length = $20 label =
'Planned Arm Code';
attrib ARM length = $40 label =
'Description of Planned Arm';
attrib COUNTRY length = $3 label =
'Country';
attrib DMDTC length = $64 label =
'Date/Time of Collection';
attrib DMDY length = 8 label = 'Study
Day of Collection';
merge DEMO1(in=a)
TRTCODES(in=t);
by SUBJID1;
keep STUDYID DOMAIN USUBJID SUBJID
RFSTDTC RFENDTC SITEID INVID INVNAM
BRTHDTC AGE AGEU SEX RACE ETHNIC ARMCD
ARM COUNTRY DMDTC DMDY;
STUDYID='NIC001';
DOMAIN='DM';
USUBJID=left(put(SUBJID1,Z6.));
SUBJID=substr(compress(put(
SUBJID1,Z6.)),4,3);
if nmiss(TRTSTDT)=0 then RFSTDTC=put(
TRTSTDT, IS8601DT.);
if nmiss(TRTENDT)=0 then
RFENDTC=put(TRTENDT, IS8601DT.);
SITEID="01";
INVID=' ';
INVNAM=INVNAME;
if nmiss(BIRTHDT)=0 then
BRTHDTC=put(BIRTHDT,IS8601DA.);
if nmiss(BIRTHDT, TRTSTDT)=0 then
AGE=(floor((intck(
'month',BIRTHDT,datepart(TRTSTDT))-
(day(datepart(TRTSTDT))<day(
BIRTHDT)))/12));
AGEU='YEARS';
ETHNIC=' ';
if TRTCD=1 then ARMCD='NIC75';
else if TRTCD=2 THEN ARMCD='NIC15';
else ARMCD='';
ARM=TRTGRP;
```

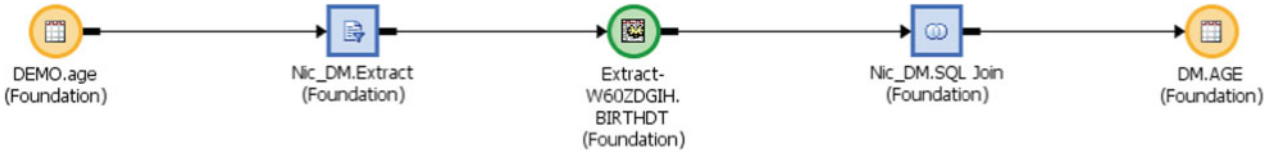

Figure 5 Tracing the AGE column

```
if upcase( COUNTRY1 )="UNITED STATES"
then COUNTRY='USA';
SEX=SEX1;
DMDTC=put( TRTSTDT,IS8601DT. );
if nmiss( TRTSTDT,TRTENDT )=0 then do;
if datepart( TRTENDT ) >= datepart(
TRTSTDT ) then DMDY=datepart( TRTENDT )
-datepart( TRTSTDT ) + 1;
else DMDY=datepart( TRTENDT )-date-
part( TRTSTDT );
end;
run;
```

SAS Code 2

From this code, we can see that there is no way of tracking where a variable comes from, so we need to add in some code that will build a separate tracking dataset. We can add the code analyzer as shown in the metadata section to extract the metadata information, but that does not help tracing.

Building the trace

Consider the requirements specifications for the DM SDTM domain as we can see in Fig. 6. Each of the target variables has a source and derivation listed. These can either be sourced in the raw data or derived in the domain transformation. Figures 6–9 are represented in EXCEL® format, but a SAS Dataset would be built in the same guise.

The same is applicable for the ADaM domains. The target columns can be traced back to the DM domain or if they were derived.

The next logical step is to append these together and trace back from Analysis to the raw data.

It would make sense to have one more column for the calculations performed when creating the analysis datasets. Include the procedures used to get the statistics so that when tracing back from the TFLs, even the calculation can be seen. For example, a proc means to obtain the statistics for the Age or Height variables could be added.

```
proc MEANS data=DM noprint;
var AGE;
output out=MEANS median=MedianAge
max=MaxAge min=MinAge
std=StdAge mean=MeanAge;
run;
```

It could be considered that the define document is being extended. The computed column space is used for some of this tracing. Consider it further down the line and the hyperlink is included on the TFL itself. The code, the derivations, and source could be referenced in one click.

A standard table shell is shown in Table 1, but contains a hyperlink on the AGE variable. That link takes you to the code. There is a hyperlink on the data=DM portion of the code that links back to the table of sources and descriptions. That is the simplest way to make the links. A more complex way is to graphically link them, but there is no easy way of doing that.

TFLs

Standard TFLs have been the same for years. When will the static information displayed in the tables become insufficient for the needs of the reviewing bodies? With the advent of define.xml and the pdf version with bookmarked links, we have seen the introduction of dynamic reports to send to the regulatory bodies. Let's consider how we can enhance that experience. Display the TFLs in the front end of the report and link the data elements displayed through to the computational algorithm allocated in the define.pdf portion of the report.

Alternatively, a different way of displaying the linked information could be devised, one showing the flow of data elements either using a table (Fig. 9) or a flow diagram (Fig. 10).

Consider, then, a more complex efficacy variable such as progression free survival. The easiest way to trace that variable is to work the path backwards.

- How is it calculated for display in the table?
- What are the inputs for that calculation in the ADaM data?
- What are the observed data elements (SDTM) that feed into the ADaM derivation?
- What are the original data elements in CDASH?

Considering those four points, and with information at the end of the flow, it should be possible to make those connections through the process, just with more inputs and calculations. The input items may be used in more than one final calculation, but the final use in the TFLs could be identified by a unique number and then have the traceability listed back from there.

If this traceability was done at the start of the trial, could it help to streamline the volume of data collected? If you know which data elements drive your trial reporting and as we progress backwards through the flow, there are a finite number of data elements used — would you need to collect the additional spurious elements?

The tables are simple and easy to read, but are limited in their informational capabilities in regard to where the displayed information comes from. The approach given in this paper would depend upon, not the creation of the tables — as the links are fairly easy to produce, but the creation of the back end traceability data that lend themselves to the links, and how those data are stored. While these innovations are not yet in practice, the author considers that this approach may be favorable in order to ease the tracing of data for the authorities.

Traceability

Some ETL tools provide the facility to trace data elements through their system. From source to target

Variable Name	Variable Label	Variable Type	Core	Display Format	Codelist / Controlled Terms	Source / Derivation
STUDYID	Study Identifier	Char	Req	50		"NIC001"
DOMAIN	Domain Abbreviation	Char	Req	2	**DM	"DM"
USUBJID	Unique Subject Identifier	Char	Req	50		left(put(DEMO.SUBJID1,Z6.));
SUBJID	Subject Identifier for the Study	Char	Req	50		substr(compress(put(DEMO.SUBJID1,Z6.)),4,3);
RFSTDTC	Subject Reference Start Date/Time	Char	Exp	50	ISO 8601	put(DEMO.TRTSTDT, IS8601DT.)
RFENDTC	Subject Reference End Date/Time	Char	Exp	50	ISO 8601	put(DEMO.TRTENDT, IS8601DT.)
SITEID	Study Site Identifier	Char	Req	8		"01";
INVID	Investigator Identifier	Char	Perm	8	*	" "
INVNAM	Investigator Name	Char	Perm	50		DEMO.INVNAME;
BRTHDTC	Date/Time of Birth	Char	Perm	50	ISO 8601	put(DEMO.BIRTHDT,IS8601DA.);
AGE	Age in AGEU at RFSTDTC	Num	Exp	8		(floor((intck('month',DEMO.BIRTHDT,datepart(DEMO.TRTSTDT)) - (day(datepart(DEMO.TRTSTDT)) < day(DEMO.BIRTHDT))) / 12));
AGEU	Age Units	Char	Exp	50	** YEARS	"YEARS"
SEX	Sex	Char	Req	1	**M, F	DEMO.SEX
RACE	Race	Char	Exp	100	**	DEMO.RACE

Figure 6 SDTM DM implementation guide example

is the end of the line. But unless the tool is used from end to end, from the input of the data through to the TFLs, how can that data be fully traced? The point of traceability is to ensure the quality and integrity of the final variable values. In hindsight of a study, when tracing back variables from the tables, would it shed light on how many variables are collected and not used? This leads to the question of what drives the data flow of a study. Are the specifications and database built first? Those data are then collected and passed through the flow to the tables at the end. However, consider defining the tables first and working backwards. Would that make the data capture and flow more efficient? Finally, how would the ETL tool be able to link through to the display TFLs unless the code for those was too created within the ETL tool? There seems to be great benefits available for reusability from ETL tools, but not necessarily for full traceability.

We have considered simple and briefly touched on more complex data flows. The principles of traceability remain the same, and there can be no doubt that a simple way of looking back from the results to the raw data via the twists and turns that data elements take along the way will provide value for those reviewers who may have to sort their way through tables and tables of data to ensure correct calculation. There is no one system out there that

Variable Name	Variable Label	Variable Type	Core	Display Format	Codelist / Controlled Terms	Source / Derivation
STUDY IDENTIFIERS: KEY VARIABLES THAT IDENTIFY THE STUDY, SUBJECT, AND SITE						
STUDYID	Study Identifier	Char	Req	9		DM.STUDYID
USUBJID	Unique Subject Identifier	Char	Req	18		DM.USUBJID
SUBJID	Subject Identifier for the Study	Char	Req	8		DM.SUBJID
SITEID	Study Site Identifier	Char	Req	3		DM.SITEID
COUNTRY	Country	Char	Perm	3		DM.COUNTRY
DEMOGRAPHICS: INCLUDES SUBJECT DEMOGRAPHICS, GROUPING VARIABLES, AND FLAGS.						
AGE	Age	Num	Req	integer		DM.AGE
AGEU	Age Units	Char	Req	6	(AGEU)	DM.AGEU
AGECAT1	Age Category 1	Char	Perm	6	AGECAT1	if AGE<50 then '<50' else if AGE>50 then '>=50'
SEX	Sex	Char	Req	8	(SEX)	DM.SEX
RACE	Race	Char	Req	40	(RACE)	DM.RACE
RACECAT1	Race Category 1	Char	Perm	15	RACECAT1	if RACE='WHITE' then 'Caucasian' else 'Non-Caucasian'
ARM	Description of Planned Arm	Char	Req	50		DM.ARM
DISCREAS	Reason for Discontinuation	Char	Cond	40		DS.DSDECOD where DSCAT="DISPOSITION EVENT" and EPOCH="XXXXXX"

Figure 7 ADaM ADSL implementation guide example

Variable Name	Variable Label	Variable Type	Core	Display Format	Codelist / Controlled Terms	Source / Derivation	Original Source / Derivation
STUDY IDENTIFIERS: KEY VARIABLES THAT IDENTIFY THE STUDY, SUBJECT, AND SITE							
STUDYID	Study Identifier	Char	Req	9		DM.STUDYID	"NIC001"
USUBJID	Unique Subject Identifier	Char	Req	18		DM.USUBJID	left(put(DEMO.SUBJID1,Z6.));
SUBJID	Subject Identifier for the Study	Char	Req	8		DM.SUBJID	substr(compress(put(DEMO.SUBJID1,Z6.)),4,3);
SITEID	Study Site Identifier	Char	Req	3		DM.SITEID	"01";
COUNTRY	Country	Char	Perm	3		DM.COUNTRY	if upcase(COUNTRY1)="UNITED STATES" then COUNTRY='USA'
DEMOGRAPHICS: INCLUDES SUBJECT DEMOGRAPHICS, GROUPING VARIABLES, AND FLAGS.							
AGE	Age	Num	Req	integer		DM.AGE	(floor((intck('month',DEMO.BIRTHDT,datepart(DEMO.TRTSTDT)) - (day(datepart(DEMO.TRTSTDT)) < day(DEMO.BIRTHDT))) / 12));
AGEU	Age Units	Char	Req	6	(AGEU)	DM.AGEU	"YEARS"
AGECAT1	Age Category 1	Char	Perm	6	AGECAT1	if AGE<50 then '<50' else if AGE>50 then '>=50'	
SEX	Sex	Char	Req	8	(SEX)	DM.SEX	DEMO.SEX
RACE	Race	Char	Req	40	(RACE)	DM.RACE	DEMO.RACE
RACECAT1	Race Category 1	Char	Perm	15	RACECAT1	if RACE='WHITE' then 'Caucasian' else 'Non-Caucasian'	
ARM	Description of Planned Arm	Char	Req	50		DM.ARM	DEMO.TRTGRP
DISCREAS	Reason for Discontinuation	Char	Cond	40		DS.DSDECOD where DSCAT="DISPOSITION EVENT" and EPOCH="XXXXXX"	

Figure 8 ADaM implementation Guide+SDTM derivations

does this completely, but there are ways of manually putting this together to add value to the data flow.

Conclusion

There are several points to raise after considering the evidence:

1. Get on the standards path. If all the companies used industry standards, think how easy it would be for new recruits, CROs, and the authorities to easily use the data. For companies, there is a chance it would reduce their time to market.

2. Get ahead of the game, no point in being reactive, so drive the change. To that point, look at an ETL solution. Standards+ETL=Reusable 'Jobs' → Cutting down the time to do standard work. Leave more time for the complex work. Programmers need not be scared of their jobs changing for the worse. They will be able to use more skill and freedom on complex derivations and programming.

3. CDISC SHARE Committee is narrowing the gap between different standards and their initiatives will provide a platform upon which better traceability can be obtained.

4. TFLs remain static for now, but the 'Define' documents have been around for a while and who can say they cannot be enhanced to include data element traceability. A hyperlink in a pdf document links to source code, source variables, and derivations to provide the link from raw data to the tables.

5. There are no tools currently out there that can trace a data element from its raw data form through to the tables figures and listings.

6. It is better to start at the end of the data flow and work backwards. Figure out the TFLs required, work out the inputs, trace it back to the common data elements required for collection, and save on unused data elements collected.

7. Although not yet required, there will become a time where the transparency of data will be a requirement, where ease of referring to critical variables and how they were formulated from the original data will be an important part of the reporting process.

References

1 Available from: http://en.wikipedia.org/wiki/Traceability (cited 2011 Sep 1).
2 Available from: http://www.cdisc.org/cdisc-share (cited 2011 Sep 1).

Variable Name	Variable Label	Variable Type	Core	Display Format	TFL Calculation	Source / Derivation
STUDY IDENTIFIERS: KEY VARIABLES THAT IDENTIFY THE STUDY, SUBJECT, AND SITE						
STUDYID	Study Identifier	Char	Req	9		DM.STUDYID
USUBJID	Unique Subject Identifier	Char	Req	18		DM.USUBJID
SUBJID	Subject Identifier for the Study	Char	Req	8		DM.SUBJID
SITEID	Study Site Identifier	Char	Req	3		DM.SITEID
COUNTRY	Country	Char	Perm	3		DM.COUNTRY
DEMOGRAPHICS: INCLUDES SUBJECT DEMOGRAPHICS, GROUPING VARIABLES, AND FLAGS.						
AGE	Age	Num	Req	integer	proc MEANS data=DM noprint; var AGE; output out=MEANS median=MedianAge max=MaxAge min=MinAge std=StdAge mean=MeanAge; run;	DM.AGE

Figure 9 Traceability for DM TFL (note: some columns are hidden)

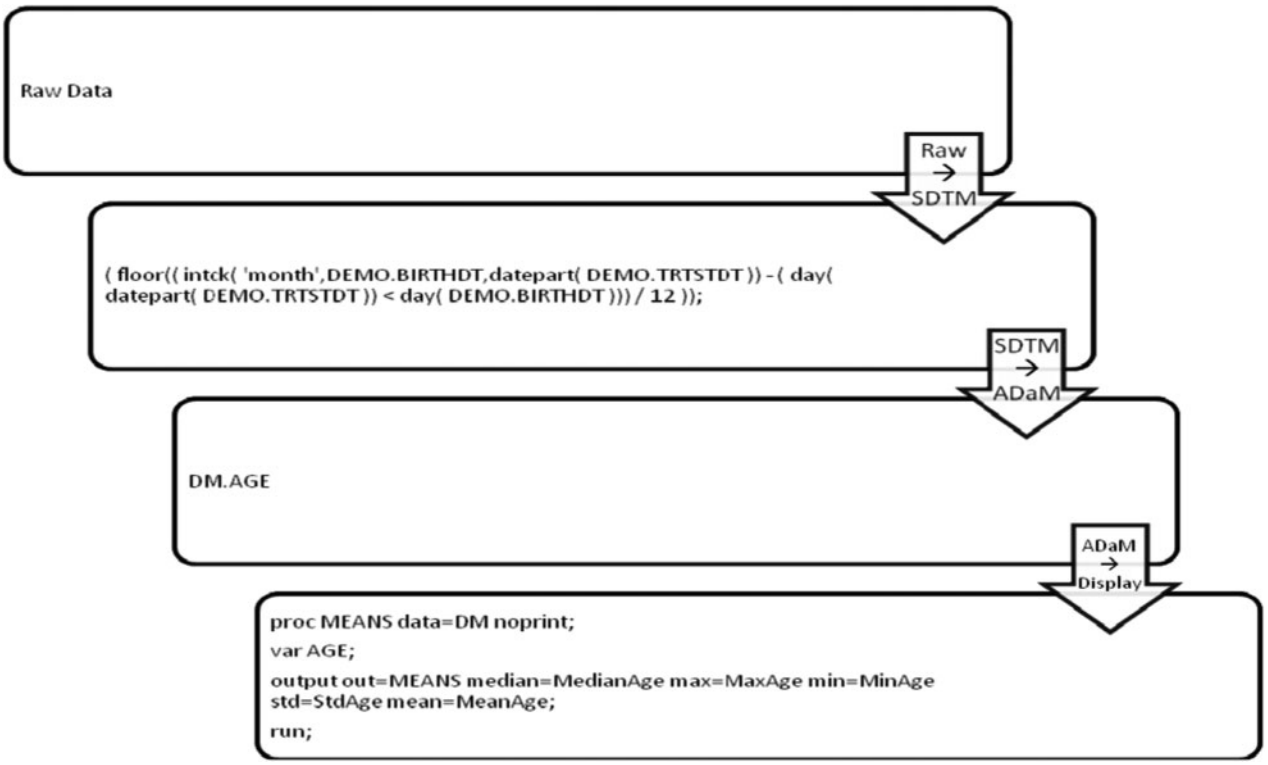

Figure 10 Flow diagram (example) for AGE

Table 1 Example demographics table

Subject demographics — Page x of y

hh:mm ddmmmyyyy

ITT population

status

STUDY #

	Arm A (pem+cis)		Arm B (placebo+cis)		Total	
	(*N*=xxx)		(*N*=xxx)		(*N*=xxx)	
Parameter	n	(%)	n	(%)	n	(%)
Gender						
Female	xx	(xx.x)	xx	(xx.x)	xx	(xx.x)
Male	xx	(xx.x)	xx	(xx.x)	xx	(xx.x)
Race						
Caucasian	xx	(xx.x)	xx	(xx.x)	xx	(xx.x)
Hispanic	xx	(xx.x)	xx	(xx.x)	xx	(xx.x)
Asian	xx	(xx.x)	xx	(xx.x)	xx	(xx.x)
Age (years)						
Number of subjects	xx		xx		xx	
Mean	xx.x		xx.x		xx.x	
SD	xx.x		xx.x		xx.x	
Median	xx.x		xx.x		xx.x	
Minimum	xx.x		xx.x		xx.x	
Maximum	xx.x		xx.x		xx.x	
Height (cm)						
Number of subjects	xx		xx		xx	
Mean	xx.x		xx.x		xx.x	
SD	xx.x		xx.x		xx.x	
Median	xx.x		xx.x		xx.x	
Minimum	xx.x		xx.x		xx.x	
Maximum	xx.x		xx.x		xx.x	

Recommended Reading

1 Electronic Source Documentation in Clinical Investigations. Available from: http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM239052.pdf. [Accessed September 1st 2011].

2 CFR — Code of Federal Regulations Title 21. Available from:http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=11. [Accessed September 1st 2011].

3 Computerized Systems Used in Clinical Investigations. Available from: http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM070266.pdf. [Accessed September 1st 2011].

4 Part 11, Electronic Records; Electronic Signatures — Scope and Application. Available from: http://www.fda.gov/downloads/RegulatoryInformation/Guidances/UCM126953.pdf. [Accessed September 1st 2011].

5 General Principles of Software Validation; Final Guidance for Industry and FDA Staff. Available from: http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm085371.pdf. [Accessed September 1st 2011].

6 CDISC Shared Health and Clinical Research Electronic Library. Available from: http://www.cdisc.org/cdisc-share. [Accessed September 1st 2011].