PharmaSUG 2018 - Paper AD-09

Macro-Supported Metadata-Driven Process for Mapping SDTM VISIT and VISITNUM

Eric Crockett, Pragathi Dayananda, Covance Inc.

ABSTRACT

Programming SDTM VISIT and VISITNUM for a specific domain is often a simple task. Programming SDTM VISIT and VISITNUM consistently across all domains and multiple input data sources is a much bigger task. While it is tempting to generate visit mapping code on the fly, time spent designing a generalized visit-mapping process will save time and money, and better guarantee a conformant, accurate result.

Companies seeking to establish a general solution for SDTM visit mapping often base the process on metadata that associates raw visit content with SDTM VISIT and VISITNUM. When properly designed, visit mapping metadata provides a single input that can be used to assign visits for all raw data sources. Where things can get tricky is that the merge strategy to assign visits using the mapping metadata often varies across raw data sets.

Two macros will be shared, including examples and code, which facilitate the use of visit mapping metadata to assign SDTM visits. These two macros automatically identify the proper raw visit variables and determine the appropriate join strategy between the source data and the visit mapping metadata.

INTRODUCTION

The relation between metadata and SDTM visit mapping is the best choice when it comes to a generalized approach for standard visit mapping strategies. The question then becomes “how should this be approached?” In the beginning, stipulations included:

- A minimal number of inputs
- Communication of issues through notes in the log
- Cannot have any mismapped visits arising from metadata duplication on the supporting data set
- Assigns as many SDTM visits as possible

It was unknown whether there would be a place for standard macros in the process. This process has the ability to simplify database migrations by merely updating the metadata mapping values post-migration in a single spreadsheet. It simplifies the mapping process on the individual specifications since only the relevant raw data variables are required and a reference to using macros. Additionally, and more importantly this process expedites refresh requests even when metadata is not provided (this *never* happens) as issues with the mapping are communicated through the logs.

Generally speaking, the gap between visit mapping and programming is left to programmers to decide how to handle—each taking their own approach. This loose organization has a tendency to lead to inconsistencies in visit programming strategies and more importantly is always open to a lack of robustness. In terms of robustness, this concern is eliminated. This process is fully supported, even in lieu of tangible metadata for which a supplementary macro solution is available.

PROCESS

Figure 1. Process Flow

The process starts with receiving the raw data. The first task is to gather the available metadata and combine these into a single spreadsheet with special naming conventions. Once all the visit metadata is gathered on a spreadsheet, the next step is to map the SDTM visits. To simplify the creation of this supporting data set, referred to as tv_map, only the VISIT values are required with the exception of unscheduled visits which are study-specific and not captured in the TV domain. This nicely coincides with the mapping strategy as VISITNUM values can update on the TV domain and be easily translated through to the tv_map data set. Unscheduled visit VISITNUM values are not captured in the trial visits domain and are therefore determined by the value provided by the user on the spreadsheet. This eliminates any inconsistencies between the tv_map VISIT and VISITNUM and TV since the VISITNUM values are pulled directly from the trial visits domain.

FolderName_C	VISIT_C	VISITNUM_C	FolderSeq_N	InstanceRepeatNumber_N	VISITNUM_N	VISITNUM	VISIT
Unscheduled			70	15		999	Unscheduled
Unscheduled			70	16		999	Unscheduled
Unscheduled			70	17		999	Unscheduled
Week 1 Day 1			8	0			Week 1 Day 1
Week 1 Day 2			9	0			Week 1 Day 2
Week 11 Day 1			15	0			Week 11 Day 1
Week 13 Day 1			16	0			Week 13 Day 1
Week 15 Day 1			17	0			Week 15 Day 1

Figure 2. Example of Spreadsheet Requirements

Each row on the spreadsheet must be considered when evaluating whether the metadata maps to an SDTM visit or not. Once the mapping is completed, the spreadsheet is read into a SAS® session and joined with the existing SDTM TV data set to ensure proper formats of VISIT and VISITNUM are applied throughout programming. The newly generated tv_map data set with the proper SDTM VISIT and VISITNUM variables is then made available to the programming team. The programming team then uses the two SAS® macros and this data set to join the SDTM visits to the raw data.

VARIABLE TYPES AND NAMING CONVENTIONS

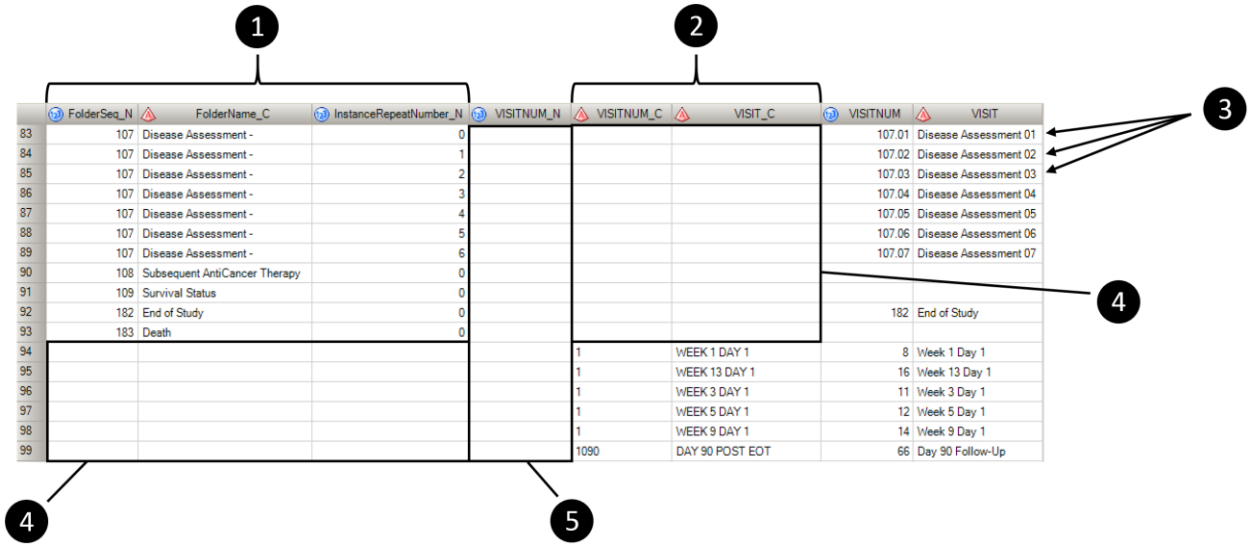
The first stop along this treacherous endeavor is the handling of variable types and naming conventions. Under this process the naming convention for the metadata is related to the variable types from the raw data. To get straight to the point, "_N" is appended to the variable name(s) of numeric variable type—likewise, "_C" is appended to the variable name(s) of character variable type. Following this convention eliminates the issue of distinguishing between metadata and the SDTM VISIT and VISITNUM variables. See Figure 2 above as an example of a case where tv_map ends up with the variables VISITNUM_C, VISITNUM_N and VISITNUM where the latter holds the role of the SDTM VISITNUM. So, as a natural consequence of this naming convention, the variables VISIT, and VISITNUM are reserved variable names on tv_map and are intended to only represent the actual SDTM visits to be mapped.

STRUCTURE AND GENERAL USE OF TV_MAP DATA SET

The tv_map data set is misshapen since this data set is comprised of all the metadata for an entire study. However, we are not concerned with the variables from the other source data sets while processing. When any of the variables in common with tv_map are populated, we would not be using the remaining metadata mapping to bring visits onto the raw data set.

Depicted below in Figure 3 is an example of a tv_map data set after it has been created. Columns under

items (1) and (2) represent two different raw data sources. The first is the Medidata Rave™ metadata and the latter is a set of vendor metadata. When one applies, the other does not. Item (3) shows a succession of disease assessment visits that are distinguished by the variable InstanceRepeatNumber_N. In order to target the correct records from tv_map, the macros query tv_map excluding the below conditions marked by item (4), where all the values are null for the set of metadata that does not apply, dropping the irrelevant metadata mapping records from tv_map. Finally, item (5) depicts other vendor metadata that is not depicted below, and so the variable VISITNUM_N is null for the records shown here.



	FolderSeq_N	FolderName_C	InstanceRepeatNumber_N	VISITNUM_N	VISITNUM_C	VISIT_C	VISITNUM	VISIT
83	107	Disease Assessment -	0				107.01	Disease Assessment 01
84	107	Disease Assessment -	1				107.02	Disease Assessment 02
85	107	Disease Assessment -	2				107.03	Disease Assessment 03
86	107	Disease Assessment -	3				107.04	Disease Assessment 04
87	107	Disease Assessment -	4				107.05	Disease Assessment 05
88	107	Disease Assessment -	5				107.06	Disease Assessment 06
89	107	Disease Assessment -	6				107.07	Disease Assessment 07
90	108	Subsequent AntiCancer Therapy	0					
91	109	Survival Status	0					
92	182	End of Study	0				182	End of Study
93	183	Death	0					
94					1	WEEK 1 DAY 1	8	Week 1 Day 1
95					1	WEEK 13 DAY 1	16	Week 13 Day 1
96					1	WEEK 3 DAY 1	11	Week 3 Day 1
97					1	WEEK 5 DAY 1	12	Week 5 Day 1
98					1	WEEK 9 DAY 1	14	Week 9 Day 1
99					1090	DAY 90 POST EOT	66	Day 90 Follow-Up

Figure 3. Example of tv_map Highlighting Differing Raw Data Sources

Generally speaking, the visit mapping for individual specifications is simplified since the relevant variables are pre-determined by the specification writer (or designee) who generated the tv_map. The suggested use for the person responsible for generating tv_map is to include as many visit-related variables as possible. This practice ensures that the most succinct join possible is available to the macros when joining each raw data set with the tv_map data set.

REQUIREMENTS FOR PROCESSING

The only input is the data set. The macros use the metadata from the raw data set and tv_map to target the maximum number of variables required to join the two. The macro utilizes all the variables in common to write the best possible join strategy with the tv_map data set, bringing VISIT and VISITNUM variables onto the raw data set.

WHAT MUST BE DETERMINED WHILE PROCESSING

Once the data set is passed to the macros, there are a few items that can only be addressed at the time each data set is processed. It is important to remember from the original stipulations that the idea is to create a minimum number of inputs. The macros have the nuanced task of creating the best join possible bringing in as much information as can be discerned from the existing mapping.

Macro Assessment of tv_map

First and foremost, the join variables are determined. The tv_map data set is queried for only non-missing values of the join variables. This step is intended to omit the records from tv_map that correspond to other source data (See item (4) in Figure 3). Put another way, when processing a raw data set to join with tv_map, for every record on tv_map where all the join variable are null, then that mapping does not apply because it corresponds to the other raw data source(s).

One-to-One Requirement

The one-to-oneness of the join is a critical component in this process. It should never be the case that additional records are created when passing a data set through these macros. So a distinct query of the join variables on tv_map is performed, calling in additionally VISIT and VISITNUM. Only the one-to-one mappings are maintained from this query.

The many-to-one matches are written to the log bringing attention to a possible source of issues should the macro fail to cleanly join tv_map to the raw data. When this occurs, it is important to review the log for warnings generated as a result of missing a corresponding record from the tv_map.

```
WARNING: Unsupported raw visit information found.  Initiate the update of
tv_map.sas7bdat.  Number Found:  561
```

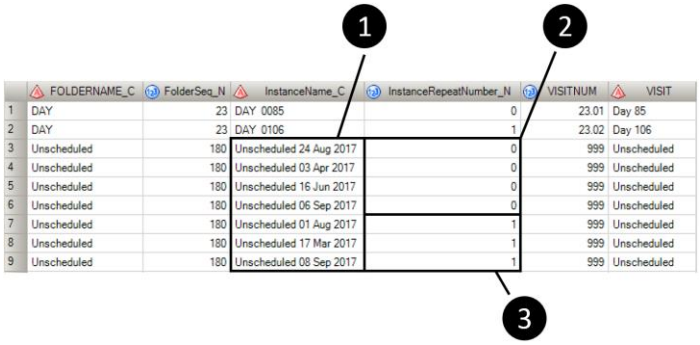
Output 1. Log Output from Missed Mapping

CONSIDERATIONS AND CASE USES

There are numerous considerations and uses of this process. It is tantamount to determine whether or not a variable should be included in the tv_map or not. Generally speaking however, the more variables considered in determining VISIT and VISITNUM mapping, the better. This is due to the fact that within this process is a step that brings in a distinct query of the tv_map data set to determine the one-to-oneness of the join with tv_map. It is easier to create the tv_map with more information than may be necessary than it is to create it with less visit variables potentially omitting variables that distinguish SDTM VISIT and VISITNUM.

MEDIDATA RAVE™ INSTANCENAME WITH SUBJECT LEVEL CAPTURES

Suppose that in a Medidata Rave™ database that InstanceName contains subject-level captures as in Figure 4, item (1), but elsewhere in the database this variable is a determining factor distinguishing between visits as in the first two rows of Figure 4 below. In such a case it is still necessary to maintain the variable on tv_map but an alternative use of tv_map is suggested. Prior to processing, dropping the variable InstanceName from the data set containing the variable InstanceName with subject-level captures will ensure that when tv_map is distinctly queried on the join variables that it simplifies and reduces down to two visit mapping metadata rows. Depicted below in items (2) and (3) is another opportunity to further reduce this metadata mapping if the variable InstanceRepeatNumber is dropped in addition to InstanceName prior to processing, fully supporting the same unscheduled visit mapping by a single row mapping to the unscheduled visit. However, it is important to note that dropping both variables opens up the possibility of a many-to-one match if both rows 1 and 2 are also on the same data set. Not to worry, the omission of many-to-one mappings will automatically fire a warning in the log if this were to be the case. Processing raw data sets with the subject-level captures in this way circumvents constantly updating tv_map on subsequent data cuts when new subject-level captures are added.



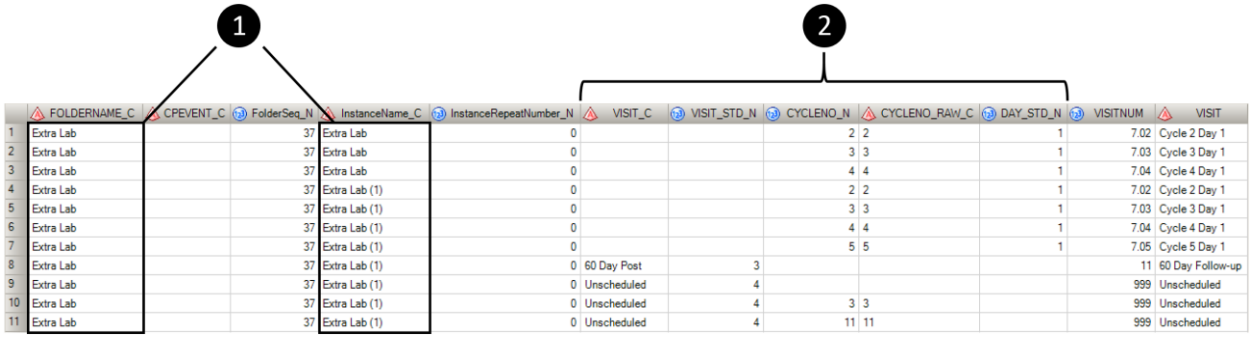
	FOLDERNAME_C	FolderSeq_N	InstanceName_C	InstanceRepeatNumber_N	VISITNUM	VISIT
1	DAY	23	DAY 0085	0	23.01	Day 85
2	DAY	23	DAY 0106	1	23.02	Day 106
3	Unscheduled	180	Unscheduled 24 Aug 2017	0	999	Unscheduled
4	Unscheduled	180	Unscheduled 03 Apr 2017	0	999	Unscheduled
5	Unscheduled	180	Unscheduled 16 Jun 2017	0	999	Unscheduled
6	Unscheduled	180	Unscheduled 06 Sep 2017	0	999	Unscheduled
7	Unscheduled	180	Unscheduled 01 Aug 2017	1	999	Unscheduled
8	Unscheduled	180	Unscheduled 17 Mar 2017	1	999	Unscheduled
9	Unscheduled	180	Unscheduled 08 Sep 2017	1	999	Unscheduled

Figure 4. Example of tv_map with Subject-Level Captures

MEDIDATA RAVE™ LAB DATA WITH ADDITIONAL VISIT CONTENT

When a case report form (CRF) page for unscheduled lab data contains additional visit content, in some situations a sponsor may choose to associate instead the scheduled SDTM visit for the unscheduled results. In such a case, it makes sense to include this content in tv_map. Due to some database restrictions, this additional unscheduled visit content ends up remaining only on the individual unscheduled pages. If this is the case, still, the best solution is to include all the metadata from the unscheduled "Extra Lab" pages as shown below by Figure 5.

It is still probably relatively unclear at this point that there is an issue. This boils down to a many-to-one mapping that if used directly on the LAB data set, VISIT and VISITNUM will be null for all of the unscheduled "Extra Lab" pages. This is due to the fact that many-to-one mappings from tv_map are always omitted. The variables in item (2), Figure 5 below, actually distinguish between VISIT and VISITNUM but are not on the LAB data set. Since the LAB data set does not contain this content, the omission of the many-to-one mappings would occur and a warning will issue in the log and no visits would be mapped for these unscheduled visits.



	FOLDERNAME_C	CPEVENT_C	FolderSeq_N	InstanceName_C	InstanceRepeatNumber_N	VISIT_C	VISIT_STD_N	CYCLENO_N	CYCLENO_RAW_C	DAY_STD_N	VISITNUM	VISIT
1	Extra Lab		37	Extra Lab	0			2	2	1	7.02	Cycle 2 Day 1
2	Extra Lab		37	Extra Lab	0			3	3	1	7.03	Cycle 3 Day 1
3	Extra Lab		37	Extra Lab	0			4	4	1	7.04	Cycle 4 Day 1
4	Extra Lab		37	Extra Lab (1)	0			2	2	1	7.02	Cycle 2 Day 1
5	Extra Lab		37	Extra Lab (1)	0			3	3	1	7.03	Cycle 3 Day 1
6	Extra Lab		37	Extra Lab (1)	0			4	4	1	7.04	Cycle 4 Day 1
7	Extra Lab		37	Extra Lab (1)	0			5	5	1	7.05	Cycle 5 Day 1
8	Extra Lab		37	Extra Lab (1)	0	60 Day Post	3				11	60 Day Follow-up
9	Extra Lab		37	Extra Lab (1)	0	Unscheduled	4				999	Unscheduled
10	Extra Lab		37	Extra Lab (1)	0	Unscheduled	4	3	3		999	Unscheduled
11	Extra Lab		37	Extra Lab (1)	0	Unscheduled	4	11	11		999	Unscheduled

Figure 5. Example of tv_map with Additional Unscheduled Visit Metadata

At this point, it is suggested to bring in this additional content on matching Subject and RecordID (a unique identifier for each CRF page in Medidata Rave™). Once this content is available on the LAB data set, a tv_map that includes this additional visit content (Figure 5) can easily be leveraged to apply the scheduled SDTM visits.

CONCLUSION

Making use of the available metadata and mapping all visits to a single source input has drastically simplified both the mapping and programming of SDTM VISIT and VISITNUM. Additionally, this process features the communication of missed mapping through log warnings and calls attention to any duplicate mappings that could be causing issues. Implementing this process has been beneficial for reducing the turnaround time in the diagnosis and resolution of SDTM VISIT and VISITNUM mapping issues.

ACKNOWLEDGMENTS

A special thanks to Covance, and Steven Kirby for supporting this endeavor.

CONTACT INFORMATION

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

Eric Crockett
Covance Inc.
escrockett@live.com