

Exploring the Possibility of Automating the SDTM Subject Visits (SV) Domain

Dony Sabu, Genpro Research, Thiruvananthapuram, India
Akhil Aniyar, Genpro Research, Thiruvananthapuram, India

ABSTRACT

SV is one of the required domains as part of every SDTM submission. In this paper, we explore the possibility of automating the creation of SV domain across therapeutic areas or phases as the structure or logic remains similar. Automation process contains five steps:

1. Finding required raw datasets and appropriate variables.
2. Obtaining earliest and latest dates (SVSTDTC/SVENDTC) corresponding to each visit.
3. Directly maps the corresponding VISITNUM and VISITDY from TV domain for scheduled visits.
4. Handles the final hurdle, that is the derivation of VISITNUM for Unplanned visits.
5. Derives the SVSTDY and SVENDY variables corresponding to SVSTDTC and SVENDTC.

The main inputs for this macro are the raw datasets, DM and Trial Visits datasets. A standard SV dataset will be generated as the final output. The idea for automation of Subject Visits domain (based on SDTM IG 3.3 and before) across studies as well as the SAS code will be covered in this article.

INTRODUCTION

Like Trial Visits (TV), SV is a special purpose domain that accommodates the actual start and end date for each visit of each individual subject. The Subject Visits domain altogether consolidates the timing information of each subject visits that is spread across other domains in variables VISITNUM and VISIT and/or VISITDY. As per the SDTM IG 3.3 the topic variable for SV domain remains as VISITNUM and thus it should be non-missing in the domain.

The derivation of start and end date corresponding to each visit become complex when there are multiple dates for a visit for a particular subject is collected, but at the same time the derivation or logic remains same across therapeutic area and phases. The method for deriving these values should be consistent with the visit definitions in the Trial Visits (TV) dataset. Hence, we created a macro for generating the whole SV dataset where we provide the RAW datasets and SDTM datasets like DM and TV as input, which helps to map the actual visit and date variables in SV. One tricky part of SV dataset is the derivation of VISITNUM for unscheduled/unplanned visits which we cannot map from Trial Visits (TV) domain, which can easily handle by the macro.

SV MACRO

The sv macro is used to create Subject Visits datasets based on the given input datasets and SDTM domains like DM and TV. This macro will create a dataset SV in the output library that we have mentioned in outlib parameter or otherwise in the work library with all the standard variables. If the programmer needs to make some modification to the SV dataset created from the macro, he/she can output it to the work library and make the required modifications. The macro parameters in the SV macro are explained below.

Macro Parameter	Description	Examples
Inplib (Required)	Input library which contains of input/raw datasets necessary for SV domain generation.	RAW or EDC
Subjid (Required)	Variable name in which subject identifier is specified in all input datasets	SUBJECT or PATIENT_ID
Outlib (Required)	The library in which the final SV output should be placed.	SDTM or SDS
vis1 (Required)	Variable which contains visit information in the input dataset which we use to derive the planned visit variable (VISIT) in SV. If the visit variable is different across datasets, we use vis2 and vis3 macro parameters to assign these variables. Note that, there can be only 1 visit variable corresponding to each dataset.	instancename
vis2 (Optional)	If there are multiple visit variables from input dataset that we use to derive VISIT along with vis1, we use vis2 macro parameter to assign the other visit parent variable.	foldername

vis3 (Optional)	If there are multiple visit variables from input dataset that we use to derive VISIT, along with vis1 and vis2, we use vis3 macro parameter to assign the other visit parent variable.	instancerepeatnumber
Incr (Required)	This is the increment of VISITNUM for unscheduled visits. For instance, consider VISITNUM of a scheduled visit is 101, and if we are planning to assign the visit number of an unscheduled visit just after the scheduled visit as 101.01 (101 + 0.01), then incr = 0.01.	0.1 or 0.01 or 0.001
exc_dt (Required)	The date variables that need to be excluded from the input datasets for deriving SVSTDTC and SVENDTC. When we consider different input datasets, there are some dates we never consider deriving SV dates such as system generated variables.	MAXUPDATED, RECORDDATE, DEATHDAT, UNBLNDAT..etc.
exc_data (Required)	List of datasets located in input library specified in "inplib" which doesn't have any role in generation of SV dataset either to derive VISIT or SVSTDTC/SVENDTC.	AE, CM, MH, CP etc.
Uplndes (Optional)	Specify the dataset which contains unplanned visits description. Make sure that the dataset should be in inplib library. Note that, macro parameter only accepts name of single dataset with unplanned visits description. If you don't want to map SVUPDES in SV, then keep this parameter as missing.	Unsc or unpldes or unspc etc.
Byval (Optional)	List of variables which we need to merge unplanned visit description and that variables should be common to other raw datasets too. Note that, variables passing to above macro parameters vis1,vis2 or vis3 will be preferable.	Instancename instancerepeatnumber
desvar1 (Optional)	Variable from the dataset specified in parameter uplndes, which contains unplanned visits description.	unsrns
desvar2 (Optional)	Variable from the dataset specified in parameter uplndes, which contains unplanned visits description for any other specified values.	unplspec

Initial requirements for running the macro.

1. A SAS library of raw datasets that we make use of deriving visits and corresponding dates.
2. An output library which we need to save the final output of SDTM SV dataset.
3. Identify those variables from raw datasets which help to the derivation of SV.VISIT.
4. For the derivation of unplanned/unscheduled visits and visit numbers, the increment for planned visit should be required which is captured in incr macro parameter.
5. SDTM domains like DM and TV should be present in the output library.

SV MACRO - WORK FLOW

Finding required raw datasets and appropriate variables.

As an initial step, we need to identify the raw datasets that helps the macros for mapping actual visits and dates. For that, we can pass the library name where input datasets are located, into the first macro parameter INPLIB. The macro identifies all the datasets in that library.

An example is given below.

```
%sv(inplib=raw ,
    subjid=subject,
    outlib=sdm ,
    vis1=instancename,
    vis2=foldername,
    vis3=instancerepeatnumber,
```

```
incr=0.01,
exc_dt=("MAXUPDATED","RECORDDATE","DEATHDAT","UNBLNDAT"),
exc_data = %str(AE, CAM, CLZBM, CM, CP, DA2, DA3, DM)
uplndes=unsc,
byval=instancename instancerepeatnumber,
desvar1= unrsn,
desvar2 = unplspect;
```

Here, the macro collects the name of all datasets from "RAW" library and make use of these datasets as input for actual visit and dates calculation.

Once we run the macro for the first time by providing all the required parameters, an excel file will be generated at the location of output library mentioned in the macro (in the above macro call, the library is sdtm) with the name **required_data1.xls**.

Working of intermediate excel file required_data1.xls

When we initially run the macro, then SAS stops executing with a custom generated warning as

WARNING: Add flag1 column with values 'exclude var' in the 1st tab RAW_DATASETS. Add the planned visits from TV in the VISIT_FMT tab.

Once we get this warning message, an excel file named required_data1.xls is generated in the location of output library with two tabs raw_datasets and visit_fmt. So, as per the warning message in the log, we need to update the first tab of required_data1.xls by creating the variable flag1 with value as "exclude var" to all those rows corresponding to the unwanted variables listed in the variable NAME. Also, edit the second tab VISIT_FMT by creating a column of name VISIT_ with values mapped from TV domain.

For example, the below screen shot from required_data1.xls details how the raw_datasets and visit_fmt tabs needs to be updated.

LIBNAME	MEMNAME	NAME	TYPE	LABEL	datasetname	flag1
RAW	DA1	DADAT	1	Date Of Assessment	'RAW.DA1'	
RAW	DA1	DADAT_DD	1	Date Of Assessment Day	'RAW.DA1'	exclude var
RAW	DA1	DADAT_INT	1	Date Of Assessment Interpolated	'RAW.DA1'	exclude var
RAW	DA1	DADAT_MM	1	Date Of Assessment Month	'RAW.DA1'	exclude var
RAW	DA1	DADAT_YYYY	1	Date Of Assessment Year	'RAW.DA1'	exclude var

This screenshot explains how the flag1 variable is created in raw_datasets tab, by considering NAME column. Here values which are highlighted in yellow like DADAT_DD, DADAT_INT, DADAT_MM and DADAT_YYYY are not useful to map date of actual visit, so we present value "exclude var" in flag1 column as highlighted in green.

Below is the screenshot from visit_fmt tab.

INSTANCENAME	VISIT_
008 Day 1	Day 1
008 Day 11	Day 11 - Telephone Contact
008 Day 15	Day 15
008 Day 18	Day 18 - Telephone Contact
008 Day 22	Day 22
008 Day 29/Early Withdrawal	Day 29/Early Withdrawal
008 Day 4	Day 4 - Telephone Contact

Here, column instancename is the unique visits collected in input datasets from raw library and we create column VISIT_ by manually mapping the planned visits from Trial Visits dataset corresponding to each collected visit.

Once the excel sheet is altered by adding flag1 and VISIT_ variables, ensure to save and close the file. As next step, we run the macro for second time.

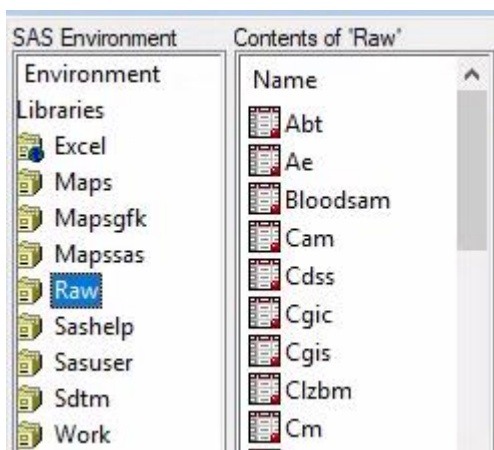
Obtaining earliest and latest dates (SVSTDTC/SVENDTC) corresponding to each visit.

Sv macro derives the date variables SVSTDTC/SVENDTC by making use of all date variables collected in datasets from the input library.

Macro automatically identifies the date variables from all datasets in input library by considering label of each

variables.

For example, assume there are multiple datasets as input, and each dataset contains multiple variables too. Some of them are date and rest of them are not. Here Sv macro tries to identify only the date variable by making use of the label and check whether the word 'Date' is present.



Here raw library contains different datasets, so SV macro identifies datasets ABT, AE, BLOODSAM, CAM, CDSS, CGIS, CLZBM, CM etc. from raw library. Then, it checks whether all these datasets contain date variables by checking the labels of each variable from each dataset. The next figure gives you an idea how sv macro collects date variables by considering string "Date" from the label.

LIBNAME	MEMNAME	NAME	LABEL
RAW	ABT	ABTDAT	Date of Assessment
RAW	ABT	ABTDAT_DD	Date of Assessment Day
RAW	ABT	ABTDAT_INT	Date of Assessment Interpolated
RAW	ABT	ABTDAT_MM	Date of Assessment Month
RAW	ABT	ABTDAT_YYYY	Date of Assessment Year
RAW	ABT	FOLDERNAME	Folder name
RAW	ABT	INSTANCENAME	Folder instance name
RAW	ABT	INSTANCEREPEATNUMBER	InstanceRepeatNumber
RAW	ABT	MAXUPDATED	Latest data update time
RAW	ABT	RECORDDATE	Clinical date of record (ex: visit date)
RAW	AE	AEENDAT	End Date
RAW	AE	AEENDAT_DD	End Date Day
RAW	AE	AEENDAT_INT	End Date Interpolated
RAW	AE	AEENDAT_MM	End Date Month
RAW	AE	AEENDAT_YYYY	End Date Year
RAW	AE	AESTDAT	Onset Date
RAW	AE	AESTDAT_DD	Onset Date Day
RAW	AE	AESTDAT_INT	Onset Date Interpolated

These are the date and visit variables from each dataset in raw library.

In some cases, all the date variables that macro identified are not useful to derive SVSTDTC and SVENDTC. For example, the date variables such as AESTDAT and AEENDAT are not useful for SV dataset creation. So, make sure that input AE dataset is passed to the exc_data parameter. Also, the dates ABTDAT_DD, ABTDAT_MM, ABTDAT_YYYY are not useful since we have the full date collected in ABTDAT variable. So, once the required_data1.xls file is created, make sure to update flag1 variable as "exclude var" for those variables. Similarly, for system generated variables like MAXUPDATED or RECORDDATE. In such cases, we make use of macro parameter exc_dt. We can pass those date variables that are common to most of the input dataset and are not useful to derive the dates, to the macro parameter exc_dt.

For example, this is how we pass system generated date variables that needs to be excluded into the parameter exc_dt.

```
%sv(inplib=raw ,
    Subjid=subject ,
    outlib=sdm ,
    vis1=instancename,
    vis2=foldername,
    vis3=instancerepeatnumber,
    incr=0.01,
    exc_dt=("MAXUPDATED","RECORDDATE","DEATHDAT","UNBLNDAT"),
    exc_data = %str(AE, CAM, CLZBM, CM, CP, DA2, DA3, DM)
    uplndes=unsc,
    byval=instancename instancerepeatnumber,
    desvar1= unsrsn,
    desvar2 = unplspect);
```

Directly maps the corresponding VISITNUM and VISITDY from TV domain.

Macro directly maps VISITNUM and VISITDY from SDTM TV dataset. So, the availability of TV dataset in output library is one of the initial requirements for running the macro.

Macro collects the planned visits information from VISIT variable in TV and maps the corresponding VISITNUM and VISITDY variables to SV domain.

DOMAIN	VISITNUM	VISIT	SUBJECT	INSTANCENAME
TV	1	Screening	801-030	008 Day 1
TV	2	Baseline	801-030	008 Day 11
TV	3	Day 1	801-030	008 Day 15
TV	4	Day 4 - Telephone Contact	801-030	008 Day 18
TV	5	Day 8	801-030	008 Day 22
TV	6	Day 11 - Telephone Contact	801-030	008 Day 29/Early Withdrawal
TV	7	Day 15	801-030	008 Day 4
TV	8	Day 18 - Telephone Contact	801-030	008 Day 8
TV	9	Day 22	801-030	Baseline
TV	10	Day 29/Early Withdrawal	801-030	Safety follow-up - Day 30
TV	11	Safety Follow-up	801-030	Safety follow-up - Day 7
TV	12	SAE Follow-up - Telephone Contact	801-030	Screening

Planned VISITNUM and VISIT variables from TV dataset.

Actual visits for a particular subject in raw data.

SV macro will do the mapping of actual VISIT from raw datasets to the standardized version from TV dataset by making use of visit_fmt tab in required_data1.xls file.

Handles the final hurdle that is the derivation of VISITNUM for Unplanned visits.

One of the main hurdles on SV programming is the approach of deriving the VISITNUM for unplanned/unscheduled visits. The approach varies from study to study and it completely depends on the standards from client/sponsor.

The approach that we used to derive VISITNUM in the macro is by incrementing 0.01 as a default value to the preceding VISITNUM of the planned VISIT.

For customizing the increment value, we have a macro parameter **incr** specified in the macro.

For instance, consider the preceding VISIT just before an unplanned visit is CYCLE 1 DAY 3 with VISITNUM as 103, then we add 0.01 to the first unplanned visit just after the planned VISIT. I.e., Unplanned VISITNUM is 103.01. If there are multiple visits after a planned VISIT, we will use the same approach to create the VISITNUM for all the unplanned visits.

USUBJID	VISITNUM	VISIT
TRID-02-801-030	1	Screening
TRID-02-801-030	2	Baseline
TRID-02-801-030	3	Day 1
TRID-02-801-030	4	Day 4 - Telephone Contact
TRID-02-801-030	5	Day 8
TRID-02-801-030	6	Day 11 - Telephone Contact
TRID-02-801-030	7	Day 15
TRID-02-801-030	8	Day 18 - Telephone Contact
TRID-02-801-030	8.01	Unscheduled 8.01
TRID-02-801-030	9	Day 22
TRID-02-801-030	10	Day 29/Early Withdrawal
TRID-02-801-030	11	Safety Follow-up
TRID-02-801-030	12	SAE Follow-up - Telephone Contact

In addition to VISITNUM corresponding to each unscheduled visit, we can map SVUPDES by making use of macro parameters uplndes, byval, desvar1 and desvar2. Sv macro only handles the situation where unplanned visits description is collected only on one dataset. The macro will not handle cases where unscheduled visit descriptions are collected in multiple datasets and make sure to give these macro parameters as missing for those instances. We can create the final sv dataset in work library using the macro and further modification can be done to SV program to map SVUPDES variables.

Derives the SVSTDY and SVENDY variables corresponding to SVSTDTC and SVENDTC.

Deriving the day variables are the final touch to develop SV dataset. Similar to the derivation of day variables that we derive in other SDTM domains, we consider date part of DM.RFSTDTC (if time is available) to find the difference in days from SVSTDTC/SVENDTC. Here comes the importance of availability of SDTM DM datasets in output library.

Screenshot of the SDTM SV dataset from Output library is pasted below.

STUDYID	DOMAIN	USUBJID	VISITNUM	VISIT	VISITDY	SVSTDTC	SVENDTC	SVSTDY	SVENDY	SVUPDES
TRID-02	SV	TRID-02-801-030	1	Screening		2020-07-30	2020-07-31	-21	-20	
TRID-02	SV	TRID-02-801-030	2	Baseline		2020-08-20	2020-08-20	1	1	
TRID-02	SV	TRID-02-801-030	3	Day 1		2020-08-20	2020-08-20	1	1	
TRID-02	SV	TRID-02-801-030	4	Day 4 - Telephone Contact		2020-08-24	2020-08-24	5	5	
TRID-02	SV	TRID-02-801-030	5	Day 8		2020-08-27	2020-08-27	8	8	
TRID-02	SV	TRID-02-801-030	6	Day 11 - Telephone Contact		2020-08-31	2020-08-31	12	12	
TRID-02	SV	TRID-02-801-030	7	Day 15		2020-09-03	2020-09-03	15	15	
TRID-02	SV	TRID-02-801-030	8	Day 18 - Telephone Contact		2020-09-08	2020-09-08	20	20	
TRID-02	SV	TRID-02-801-030	8.01	Unscheduled 8.01		2020-09-08	2020-09-08	20	20	Other-KIT REPLACEMENT
TRID-02	SV	TRID-02-801-030	9	Day 22		2020-09-10	2020-09-10	22	22	
TRID-02	SV	TRID-02-801-030	10	Day 29/Early Withdrawal		2020-09-16	2020-09-16	28	28	
TRID-02	SV	TRID-02-801-030	11	Safety Follow-up		2020-09-24	2020-09-24	36	36	
TRID-02	SV	TRID-02-801-030	12	SAE Follow-up - Telephone Contact		2020-10-13	2020-10-13	55	55	

If the outlib parameter is specified, the final dataset will be created in the output library with the name SV with all the attributes of SV datasets like label, length, format etc. Above is the example of a single subject and the display value of VISIT will change according to the mapping we do in the excel required_data1.xls.

Example of %sv() macro calling

1. If we don't need SVUPDES in the final output.

```
%sv(inplib=raw ,
    Subjid = subject ,
    outlib=work ,
    vis1=instancename,
    vis2=foldername,
    vis3=instancerepeatnumber,
    incr=0.01,
    exc_dt=("MAXUPDATED","RECORDDATE","DEATHDAT","UNBLNDAT"),
    exc_data = %str(AE, CAM, CLZBM, CM, CP, DA2, DA3, DM);
```

2. If we need SVUPDES, but 'other specified' info is not available in RAW data.

```
%sv(inplib=raw ,
    Subjid = subject ,
    outlib=sdtm ,
    vis1=instancename,
    vis2=foldername,
    vis3=instancerepeatnumber,
    incr=0.01,
    exc_dt=("MAXUPDATED","RECORDDATE","DEATHDAT","UNBLNDAT"),
    exc_data = %str(AE, CAM, CLZBM, CM, CP, DA2, DA3, DM),
    uplndes=unsc,
    byval=instancename instancerepeatnumber,
    desvar1= unsrsn);
```

Please note that, once you run the macro, an excel file required_data1.xls is generated in the folder of OUTLIB parameter. Ensure to make updates to this file as mentioned above. Once the edits are completed and saved, rerun the macro to get the final output SV in the OUTPUT library.

LIMITATIONS

1. As the macro does not use the information from specification it contains numerous parameters.
2. The input date variables should be in numeric type.
3. If there are input datasets with visit information captured not on the primary visit variable, this should be handled separately. For example, assume INSTANCENAME is the variable having visit information captured across datasets. But in one dataset, INSTANCENAME contains the form name and actual visit is captured in another variable. It becomes difficult to understand which variable needs to be considered for derivation and those scenarios will be not be handled by the macro.
4. For variable SVUPDES, the macro handles the case when unplanned visits description is collected only on one dataset.

CONCLUSION

The approach and logic behind Subject Visits dataset are similar across therapeutic areas and phases, so why do we want to spend time on writing codes over and over. Why can't we think about "macroizing" the scenario and reuse it with minor updates. Some of the advantages of using these macros for SV generation are

1. Saves time and effort.
2. Improve efficiency and quality.
3. As the macro is built by simple SAS codes, any programmers with basic knowledge in macros can make changes as per their need if required.

REFERENCES

SV macro program is available in <https://github.com/donysabu/SV-Macro/blob/main/SV%20Macro%20code/code>

CONTACT INFORMATION

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

Author Name : Dony Sabu
 Company : Genpro Research Inc
 Address : Second Floor, Nila Building
 Technopark Campus

Thiruvananthapuram, Kerala, India

Pin : 695581

Work Phone : +91.4714.026.700

Email : dony.sabu@genproresearch.com

Website : <https://genproresearch.com/>

Author Name : Akhil Aniyar

Company : Genpro Research Inc

Address : Second Floor, Nila Building

Technopark Campus

Thiruvananthapuram, Kerala, India

Pin : 695581

Work Phone : +91.4714.026.700

Email : akhil.aniyar@genproresearch.com