

**SESUG Paper 238-2018**  
**A Macro to Add SDTM Supplemental Domain to Standard Domain**  
Pratap Kunwar, Emmes Corporation

## ABSTRACT

Many pharmaceutical and biotechnology industries are now preferring to set up Study Data Tabulation Model (SDTM) mapping in the beginning of the study rather than at the end, and use SDTM datasets to streamline the flow of data from collection through submission. When you have SDTM datasets at your disposal, it is a logical choice to use them for any clinical reports.

Getting information from the supplemental (SUPP) domain back to the parent domain is a regular step that programmers cannot avoid. But, this step can be very tricky when either (1) SUPP domain contains multiple types of identifying variables, or (2) SUPP domain is empty or does not exist.

In this presentation, I will present an easily understandable macro that will produce correct results in every possible scenario.

## INTRODUCTION

Generating any type of deliverable from SDTM datasets requires merging supplementary domain information back to the main domain. To perform this, one needs a reliable macro that works in all scenarios, including:

1. SUPP domain has multiple types of Identifying Variables (IDVAR)
2. SUPP domain is empty or does not exist.

Merging the SUPP domain back to the parent domain is done by key variables such as Unique Subject Identifier (USUBJID), Reference ID (domain REFID) or other key variables depending on types of IDVAR that exit in the SUPP domain such as Sequence Number (ASEQ).

	Study Identifier	Domain Abbreviation	Unique Subject Identifier	Sequence Number	Group ID	Reference ID	Sponsor-Defined Identifier	Link ID	Link Group ID	Reported Term for the Adverse Event	Modified Reported Term	Lowest Level Term	Lo T
1	1000	AE	VMP.00904		1	AES.32	AES.32	002		HEADACHE		HEADACHE	
2	1000	AE	VMP.00904		2	AES.9	AES.9	001		TACHYCARDIA		TACHYCARDIA	
3	1000	AE	VMP.00905		3	AES.293	AES.293	017		ABDOMINAL PAIN		ABDOMINAL PAIN	
4	1000	AE	VMP.00905		4	AES.147	AES.147	014		ABDOMINAL PAIN UPPER		ABDOMINAL PAIN UPPER	
5	1000	AE	VMP.00905		5	AES.49	AES.49	005		ARTHRALGIA		ARTHRALGIA	
6	1000	AE	VMP.00905		6	AES.224	AES.224	016		BRONCHITIS		BRONCHITIS	
7	1000	AE	VMP.00905		7	AES.79	AES.79	008		FOREIGN BODY SENSATION IN EYE		FOREIGN BODY SENSATION IN EYES	
8	1000	AE	VMP.00905		8	AES.63	AES.63	006		HEADACHE		HEADACHE	
9	1000	AE	VMP.00905		9	AES.80	AES.80	009		HEADACHE		HEADACHE	
10	1000	AE	VMP.00905		10	AES.401	AES.401	018		HICCUP		HICCUP	

Display 1. AE Domain

	Study Identifier	Related Domain Abbreviation	Unique Subject Identifier	Identifying Variable	Identifying Variable Value	Qualifier Variable Name	Qualifier Variable Label	Data Value
2876	1000	AE	VMP.01164	AEREFID	AES.443	AESSAEDT	SAE onset date	2017-08-16
2877	1000	AE	VMP.01165	AEREFID	AES.851	AESSAEDT	SAE onset date	2018-06-08
2878	1000	AE	VMP.01123	AEREFID	AES.853	AESSAEDT	SAE onset date	2018-06-15
2879	1000	AE	VMP.01165	AEREFID	AES.851	AESSMISP	AE medical event specify	ABORTION
2880	1000	AE	VMP.01123	AEREFID	AES.853	AESSMISP	AE medical event specify	FATAL OUTCOME FOR THE PREGNANCY
2881	1000	AE	VMP.00918	AEREFID	AES.76	AESSMISP	AE medical event specify	PULMONARY TUBERCULOSIS
2882	1000	AE	VMP.00904	AESQ	1	AETRTEM	Treatment Emergent Flag	Y
2883	1000	AE	VMP.00904	AESQ	2	AETRTEM	Treatment Emergent Flag	Y
2884	1000	AE	VMP.00905	AESQ	3	AETRTEM	Treatment Emergent Flag	Y
2885	1000	AE	VMP.00905	AESQ	4	AETRTEM	Treatment Emergent Flag	Y
2886	1000	AE	VMP.00905	AESQ	5	AETRTEM	Treatment Emergent Flag	Y

Display 2. AE SUPP Domain

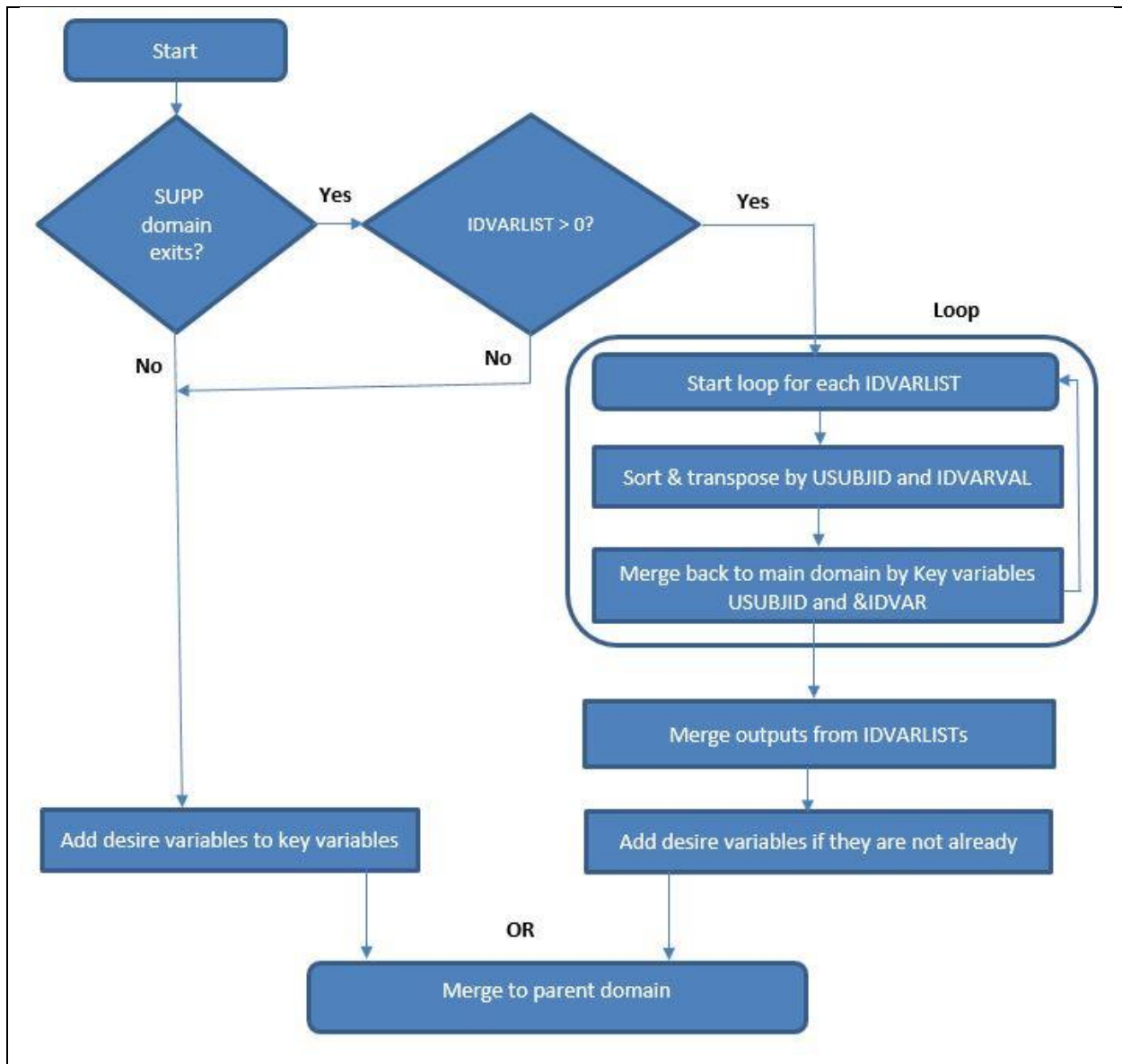


Figure 1: Macro Flow Diagram

### Macro Details:

```

%macro addsupp (
    main=,
    supp= ,
    name=,
    addcomm=,
    varlist=);
  
```

Display 3. Macro Parameter

### Step 1:

The code shown below in Display 4 gets a list of IDVAR if the SUPP domain exists. Otherwise, it assigns macro variable idvarlist = 0:

```
%let idvarlist=0;
%if %sysfunc(exist(&supp)) %then
    %do;
        PROC SQL noprint;
            select distinct IDVAR into: idvarlist separated by ' '
                from &supp;
        QUIT;
    %end;
%put &idvarlist;
```

Display 4. Get IDVAR list

### Step 2:

If the SUPP domain does not exist or is empty then add the desired variables (&varlist) to the main domain with key variables, as shown below in Display 5:

```
%if &idvarlist=0 OR NOT %sysfunc(exist(&supp)) %then
    %do;
        DATA zsupp5;
            length %upcase(&varlist) $200;
            set &main;
            keep usubjid &name.grpid &name.refid &name.seq &varlist;
        RUN;
    %end;
```

Display 5. Add desired variables to key variables of main domain

### Step 3: if SUPP domain exists

For each IDVAR, sort and transpose by key variables, and then merge to the main domain by USUBJID and IDVAR. Repeat the process for each IDVAR.

```
%else %do;
%let nidvarlist = %sysfunc(CountW(&idvarlist));

%do i = 1 %to &nidvarlist;
%let idvar=%scan(&idvarlist, &i);

PROC SORT data=&supp out=zsuppl&i nodupkeys dupout=zsupp_dups;
by usubjid idvarval qnam;
where idvar="&idvar";
RUN;

options nolabel;
PROC TRANSPOSE data=zsuppl&i out=zsupp2&i(drop=_name_
rename=(idvarval=&idvar));
by usubjid idvarval;
var qval;
id qnam;
idlabel qlabel;
RUN;
options label;

/*merge to main domain by key variables*/
PROC SQL;
create table zsuppz3&i as
select x.usubjid, x.&name.grpid, x.&name.refid,
input(x.&name.seq, best12.) as &name.seq, *
from (select usubjid, &name.grpid, &name.refid,
strip(put(&name.seq, 7.)) as &name.seq from &main)
as x left join zsupp2&i as y on
x.usubjid=y.usubjid and x.&idvar=y.&idvar
order by 1,2,3,4;
QUIT;

%end;

/*merge above dataset(s) by key variables*/
DATA zsupp4;
merge zsuppz3;;
by usubjid &name.grpid &name.refid &name.seq;
RUN;

/*add desire variables (&varlist) if not already added*/
DATA zsupp5;
length %upcase(&varlist) $200;
set zsupp4;
keep usubjid &name.grpid &name.refid &name.seq &varlist;
RUN;
%end;
```

Display 6. Macro Loop for IDVARLIST(s)

#### Step 4: Get Final Dataset

Depending on the availability and number of observations in the SUPP domain, the output from either Step 2 or Step 3 above will produce a dataset named ZSUPP5, which is then merged back to the main domain by key variables as shown in Display 7:

```
/*final output*/
PROC SQL;
  create table &name.1(drop=usubjid &name.grpid &name.refid
    &name.seq idv:) as
  select *, 1 as idv9
    from &main as x left join zsupp5 as y on
      x.usubjid=y.usubjid and x.&name.grpid=y.&name.grpid and
      x.&name.refid=y.&name.refid and x.&name.seq=y.&name.seq

      %if %upcase(&addcomm)=YES %then
        %do;
          left join CO as z on x.usubjid=z.usubjid and
            x.&name.grpid=z.idvarval;
        %end;;
QUIT;

PROC DATASETS nolist;
  delete zsupp;;
Quit;

%mend addsupp;

%addsupp(
  main=AE,
  supp=SUPPAE,
  name=ae,
  addcomm=NO,
  varlist=%str(AESDIS AESDOSE AETRTEM AEMSWFL AENRMVFL));
```

Display 7. Producing final dataset

This macro also has an option to merge the CO domain with an option in macro parameter ADDCOMM=YES, provided CO domain exists.

## CONCLUSION

Merging the SUPP domain back to the main domain is an essential step while programming with SDTM datasets. In this paper I have presented a simple macro that works in every possible scenario for all SDTM domains.

## ACKNOWLEDGMENTS

The author would like to thank my colleagues at the EMMES Corporation for their feedback and encouragement. The author would also like to thank Ryan Bratt for proofreading this paper.

## CONTACT INFORMATION

Your comments/questions/criticisms are valued and encouraged. Please contact the authors at:

Pratap Kunwar  
The Emmes Corporation  
401 N Washington St.  
E-mail: [pkunwar@emmes.com](mailto:pkunwar@emmes.com)