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 SDTM domain such as Sequence Number (ASEQ).

Display 1. AE Domain

<table>
<thead>
<tr>
<th>Study Identifier</th>
<th>Related Domain Abbreviation</th>
<th>Unique Subject Identifier</th>
<th>Sequence Number</th>
<th>Group ID</th>
<th>Reference ID</th>
<th>Supporting Information</th>
<th>Link ID</th>
<th>Event ID</th>
<th>Reported Term for the Adverse Event</th>
<th>Modified Reported Terms</th>
<th>Lowest Level Terms</th>
<th>Last Event Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>2076</td>
<td>AE</td>
<td>VMP 01104</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2077</td>
<td>AE</td>
<td>VMP 01105</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2078</td>
<td>AE</td>
<td>VMP 01123</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2079</td>
<td>AE</td>
<td>VMP 01102</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2081</td>
<td>AE</td>
<td>VMP 00610</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2082</td>
<td>AE</td>
<td>VMP 00604</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2083</td>
<td>AE</td>
<td>VMP 00604</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2084</td>
<td>AE</td>
<td>VMP 00605</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2085</td>
<td>AE</td>
<td>VMP 00605</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Display 2. AE SUPP Domain

<table>
<thead>
<tr>
<th>Study Identifier</th>
<th>Related Domain Abbreviation</th>
<th>Unique Subject Identifier</th>
<th>Sequence Number</th>
<th>Group ID</th>
<th>Reference ID</th>
<th>Supporting Information</th>
<th>Link ID</th>
<th>Event ID</th>
<th>Reported Term for the Adverse Event</th>
<th>Modified Reported Terms</th>
<th>Lowest Level Terms</th>
<th>Last Event Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>2076</td>
<td>AE</td>
<td>VMP 01104</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2077</td>
<td>AE</td>
<td>VMP 01105</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2078</td>
<td>AE</td>
<td>VMP 01123</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2079</td>
<td>AE</td>
<td>VMP 01102</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2081</td>
<td>AE</td>
<td>VMP 00610</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2082</td>
<td>AE</td>
<td>VMP 00604</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2083</td>
<td>AE</td>
<td>VMP 00604</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2084</td>
<td>AE</td>
<td>VMP 00605</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2085</td>
<td>AE</td>
<td>VMP 00605</td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
<td>AETRIO</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Display 1. AE Domain
Macro Details:

```nag
%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:

```sas
%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:

```sas
%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.

```sas
%else %do;
  %let nidvarlist = %sysfunc(CountW(&idvarlist));
  %do i = 1 %to &nidvarlist;
    %let idvar=%scan(&idvarlist, &i);

    PROC SORT data=&supp out=zsupp1&i nodupkeys dupout=zsupp_dups;
      by usubjid idvarval qnam;
      where idvar="&idvar";
    RUN;

    options nolabel;
    PROC TRANSPOSE data=zsupp1&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 ADDCOMP=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