

**Methods to Derive COVAL-COVALn in CO Domain**

Chunxia Lin, inVentiv Health Clinical, Indianapolis, IN

**ABSTRACT**

The COMMENTS (CO) domain contains free-text comments related to data in one or more SDTM domains. COVAL is the variable in CO to contain the text of comments. When text is longer than 200 characters, additional columns COVAL1-COVALn will be generated. This paper shares a couple of methods for how to split the content of comments into snippets of 200 characters long without truncating an intact word.

**INTRODUCTION**

The CO domain is a special-purpose domain that provides a solution for submitting free-text comments related to data in one or more SDTM domains or collected on a separate CRF page dedicated to comments. COMMENTS usually consist of voluntary, free-text, unsolicited observations. All SDTM variables are limited to a maximum of 200 characters due to FDA requiring datasets in SAS v5 transport format. The CO domain has a structure that allows for one comment to span multiple variables (COVAL-COVALn) in order to accommodate comments longer than 200 characters. In other words, when the comment text is longer than 200 characters, the first 200 characters of the comment will be in COVAL, the next 200 in COVAL1, and additional text stored as needed to COVALn (see example below). This paper shares two different macros/methods for automatically splitting the comment into snippets of 200 characters such that all words are left fully intact within each value.

ROW	STUDYID	DOMAIN	RDOMAIN	USUBJID	COSEQ	...	COVAL	COVAL1	COVAL2	COEVAL
1	1234	CO	AE	AB-99	1	...	First 200 characters	Next 200 characters	More Text	PRINCIPAL INVESTIGATOR
2	1234	CO	EX	AB-99	2	...	First 200 characters	Next 200 characters		PRINCIPAL INVESTIGATOR
3	1234	CO	VS	AB-99	3	...	First 200 characters	More text		PRINCIPAL INVESTIGATOR
4	1234	CO		AB-99	4	...	Total 150 characters			PRINCIPAL INVESTIGATOR

**Table 1. Example for CO Domain Model****CALL SCAN METHOD A**

CALL SCAN function returns the position and length of the *n*th word from a character string. Syntax: CALL SCAN (<string>, count, position, length <, <charlist> <, <modifier(s)>>>). The characteristics of the CALL SCAN macro are to use negative count of CALL SCAN function <CALL SCAN(temp, -1, Last\_Pos, Last\_Length, ', ')>, to return the position of first character in last word of given text, then based on the comment's length to automatically generate COVAL-COVALn. To use the macro, there are four points we need to keep in mind:

1. A negative count instructs the CALL SCAN routine to scan from right to left;
2. Last\_pos returns position of first character in last word of given text no matter last word is truncated or not;
3. Leading and trailing delimiters are ignored because the m modifier is not used;
4. A delimiter is any of several characters that are used to separate words; this paper specifies ",.", as delimiters in the charlist.

**CALL SCAN example:**

```
DATA last;
```

```

input String $60.;
call scan(string, -1, Last_Pos, Last_Length, '.', ');
Last_Word = substrn(string, Last_Pos, Last_Length);
datalines4;
Jack and Jill
& Bob & Carol & Ted & Alice &
Leonardo
! $ % & ( ) * + , - . / ;
;;;
RUN;

PROC PRINT data=last;
var Last;;
RUN;

```

Obs	Last_Pos	Last_Length	Last_Word
1	10	4	Jill
2	29	1	&
3	1	8	Leonardo
4	25	1	;

**Output 1. Output from CALL SCAN Example**

**CALL SCAN macro:**

```

%MACRO sdtm_coval_a (   indata =,      /*INVAR  - Input dataset; */
                      outdata =,     /*INVAR  - Output dataset; */
                      invar  =,      /*INVAR  - Input variable; */
                      outvar =,     /*OUTVAR - Output coval variable;*/
                      rvar   =       /*RVAR   - Left strings after output covaln
                                      Variable, if length (&invar)>200;*/
                      );
OPTIONS mprint mlogic symbolgen;

PROC SQL noprint;
select max(ceil(length(&invar)/200)-1) into: n from &indata;
QUIT;
%put &n;
/*To create global macro variable n used in below do loop*/
/*must use ceil function, e.g if length(&invar)/200=2 means 1 DO loop enough to
create coval/covall, however, if length(&invar)/200=2.1 means 2 do loop to create
coval/covall/coval2*/

DATA &outdata;

SET &indata.;

ovar=&invar; /*assign ovar as original &invar value*/

%do i=1 %to &n;

    if length(&invar)>200 then do;
        length temp $200;
        temp=substr(&invar,1,200);
        call scan(temp, -1, Last_Pos, Last_Length, '.', ');

/*1.The negative count instructs the CALL SCAN routine to scan
from right to left;2.Last_pos: returns position of first
character in last word no matter truncated or not;
3.Leading and trailing specified delimiters will be ignored */

        if substr(&invar,201,1) in (' ', '.',',')

```

```

        or substr(temp,200,1) in (' ', '.', ',') then do;
        &outvar._&i=temp;
        &rvar._&i=substr(&invar.,201);
    end;
/*1. If 200th or 201th position of input var is any of specified
delimiters ( ". , " ), which means last word is intact;
2. Else means the last word is truncated and will go to next 200*/

    else do ;
        &outvar._&i=substr(temp,1,last_pos-1);
        &rvar._&i=substr(&invar, last_pos);
    end;
end;

    else if length(&invar) le 200 then &outvar._&i=&invar;

/*If i=1, length(&invar)=<400, means only 1 do loop, coval/covall will be
generated*/
    %if &i=1 %then %do;
        coval=&outvar._&i;
        if length(&rvar._&i) le 200 then covall=&rvar._&i;

    %end;

/*If i^=1, means more than 2 do loops, starting at i=2, covall will be generated;
if max i=3, then coval2/coval3 will be generated in last do loop*/

    %if &i^=1 %then %do;
        %let j=%eval(&i-1);
        coval&j=strip(&outvar._&i);
        if length(&rvar._&i) le 200 then coval&i=strip(&rvar._&i);
    %end;

    &invar=&rvar._&i;
    /*reassign &invar to go to next do loop if length>200*/

    %end;
RUN;
%MEND;

```

## INDEXC/REVERSE METHOD B

The INDEXC function searches source from left to right, for the first occurrence of any character present in the excerpts and returns the position in source of that character. If none of the characters in excerpt-1 through excerpt-n in source are found, INDEXC returns a value of 0. The REVERSE function creates backward writing, such that the last character in the argument becomes the first character in the result, the next-to-last character in the argument becomes the second character in the result, and so on. Below macro combines INDEXC/REVERSE function to automatically derive COVAL-COVALn based on comment's length.

### INDEXC/REVERSE example

```

DATA a;
  input string $7.;
  reverse=reverse(string);
  pos=indexc(reverse, ".", " ");
  last_word=substr(string,7-pos+2);
  cards;
ABC.DEF
JX KIEE
JXX,XYZ
;

```

```
RUN;

PROC PRINT data=a;
  var string reverse pos last_word;
RUN;
```

Obs	string	reverse	pos	last_word
1	ABC.DEF	FED.CBA	4	DEF
2	JX KIEE	EEIK XJ	5	KIEE
3	JXK,XYZ	ZYX,KXJ	4	XYZ

**Output 2. Output from INDEXC/REVERSE Example**

**INDEXC/REVERSE macro:**

```
%MACRO sdtm_coval_b (   indata =,      /*INVAR - Input dataset; */
                       outdata=,      /*INVAR - Output dataset; */
                       invar =,       /*INVAR - Input variable; */
                       outvar =,      /*OUTVAR - Output coval variable;*/
                       rvar =         /*RVAR - Left strings after output covaln
                                       Variable, if length (&invar)>200;*/
                       );
OPTIONS mprint mlogic symbolgen;

PROC SQL noprint;
  select max(ceil(length(&invar)/200)-1) into: n from &indata;
QUIT;
%put &n;
/*To create global macro variable n used in below do loop*/
/*must use ceil function, e.g if length(&invar)/200=2 means 1 DO loop enough to
create coval/covall, however, if length(&invar)/200=2.1 means 2 DO loop to create
coval/covall/coval2*/

DATA &outdata;

  SET &indata.;

  ovar=&invar; /*assign ovar as original &invar value*/

  %do i=1 %to &n;

    if length(&invar)>200 then do;
      length temp $200; /*have to define temp length*/
      temp=substr(&invar,1,200);
      reverse=reverse(temp);
      /*backforward writing of temp variable*/
      pos=indexc(reverse, ". ");
      /*The INDEXC function searches source for the first occurrence of
      any character present in the excerpts and returns the position of
      that character in source*/

      if substr(reverse,temp,1,1) in (" ", ".",".")
      or substr(&invar,201,1) in (" ", ".",".") then do;
        &outvar._&i=temp;
        &rvar._&i=substr(&invar.,201);
      end;
      /*1. If 200th or 201th position of input var is any of specified
      delimiters ( ". " ), which means last word is intact;
      2. Else means the last word is truncated and will go to next200*/
      else do ;
        &outvar._&i=substr(temp,1,200-pos+1);
        &rvar._&i=substr(&invar,200-pos+2);
      end;
    end;
  end;

```

```

        end;
    end;

    else if length(&invar) le 200 then &outvar._&i=&invar;
/*If i=1, length(&invar)=<400, means only 1 do loop, coval/coval1 will be
generated*/
    %if &i=1 %then %do;
        coval=&outvar._&i;
        if length(&rvar._&i) le 200 then coval1=&rvar._&i;

    %end;

/*If i^=1,means more than 2 do loops, starting at i=2, coval1 will be generated;
If max i=3, then coval2/coval3 will be generated in last do loop*/

    %if &i^=1 %then %do;
        %let j=%eval(&i-1);
        coval&j=strip(&outvar._&i);
        if length(&rvar._&i) le 200 then coval&i=strip(&rvar._&i);
    %end;

    &invar=&rvar._&i;
/*reassign &invar to go to next do loop if length>200*/

    %end;
RUN;
%MEND;

```

#### TEST EXAMPLE

```

DATA text;
    infile cards   truncover;
    input coval_t $600. ;
    len_t=length(coval_t);
cards;
COMPARED TO PREVIOUS ECG, SIGNIFICANT CHANGES HAVE OCCURRED DUE TO CRITERIA FOR
POSSIBLE ANTERIOR INFARCT GONE
COMPARED TO BASELINE ECG, SIGNIFICANT CHANGES HAVE OCCURRED DUE TO NON-SPECIFIC
CHANGES HAVE OCCURRED DUE TO NON-SPECIFIC INTRA-VENTRICULAR CONDUCTION DELAY IS
SEEN
POSSIBLE WOLFF-PARKINSON-WHITE COMPARED TO BASELINE ECG, SIGNIFICANT CHANGES HAVE
CCURRED DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN COMPARED TO PREVIOUS ECG,
SIGNIFICANT CHANGES HAVE OCCURREDD DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN
COMPARED TO BASELINE ECG, SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AGE
UNDETERMINED, SEPTAL MI IS SEEN COMPARED TO PREVIOUS ECG, SIGNIFICANT CHANGES HAVE
OCCURRED DUE TO AGE UNDETERMINED, SEPTAL MI IS SEEN
PROLONGED QT FULLY PACED BEAT, THE QT CHANGE AND QT PROLONGATION SHOULD BE
ONSIDERED UNDER THESE CIRCUMSTANCES AND UNLIKELY TO BE DRUG EFFECT COMPARED TO
BASELINE ECG, SIGNIFICANT CHANGES HAVE OCCURRED DUE TO QTCB CHANGED BY >60 MSEC
FROM BASELINE COMPARED TO BASELINE ECG, SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AV
SEQUENTIAL OR DUAL CHAMBER ELECTRONIC PACEMAKER IS SEEN COMPARED TO PREVIOUS ECG,
SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AV SEQUENTIAL OR DUAL CHAMBER ELECTRONIC
PACEMAKER IS SEEN
;
RUN;

/*CALL SCAN method to derive COVAL-COVAL2;*/

%sdm_coval_a (indata=text,outdata=test,invar=coval_t,outvar=coval,rvar=rcoval);

DATA methoda;
    SET test;
    keep ovar coval coval1 coval2;
RUN;

```

Methods to Derive COVAL-COVALn in CO Domain, continued

```

/*INDEXC/REVERSE method to derive COVAL-COVAL2;*/

%sdm_coval_b (indata=text,outdata=test,invar=coval_t,outvar=coval,rvar=rcoval);

DATA methodb;
  SET test;
  keep ovar coval covall coval2;
RUN;

/*To create coval-coval2 without avoiding truncation*/

DATA temp;
  SET text;
  length coval covall coval2 $200 ;
  ovar=coval_t;
  if len_t<=200 then coval=substr(coval_t,1,200);
  if 200<len_t<=400 then do;
    coval=substr(coval_t,1,200);
    covall=substr(coval_t,201,400);
  end;
  if len_t>400 then do;
    coval=substr(coval_t,1,200);
    covall=substr(coval_t,201,400);
    coval2=substr(coval_t,401);
  end;
  keep ovar coval covall coval2;
RUN;

```

	ovar	coval	covall	coval2
1	COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO CRITERIA FOR POSSIBLE ANTERIOR INFARCT GONE	COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO CRITERIA FOR POSSIBLE ANTERIOR INFARCT GONE		
2	COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO NON-SPECIFIC INTRA-VENTRICULAR CONDUCTION DELAY IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO NON-SPECIFIC INTRA-VENTRICULAR CONDUCTION DELAY IS SEEN	COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO NON-SPECIFIC INTRA-VENTRICULAR CONDUCTION DELAY IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO NON-SPEC	IFIC INTRA-VENTRICULAR CONDUCTION DELAY IS SEEN	
3	POSSIBLE WOLFF-PARKINSON-WHITE COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURREDD DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN	POSSIBLE WOLFF-PARKINSON-WHITE COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURREDD	DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN	
4	COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AGE UNDETERMINED, SEPTAL MI IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AGE UNDETERMINED, SEPTAL MI IS SEEN	COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AGE UNDETERMINED, SEPTAL MI IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AGE UNDETERMINED, SEPTAL MI I	S SEEN	
5	PROLONGED QT FULLY PACED BEAT. THE QT CHANGE AND QT PROLONGATION SHOULD BE CONSIDERED UNDER THESE CIRCUMSTANCES AND UNLIKELY TO BE DRUG EFFECT COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO QTcB CHANGED BY >60MSEC FROM BASELINE COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AV SEQUENTIAL OR DUAL CHAMBER ELECTRONIC PACEMAKER IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES	PROLONGED QT FULLY PACED BEAT. THE QT CHANGE AND QT PROLONGATION SHOULD BE CONSIDERED UNDER THESE CIRCUMSTANCES AND UNLIKELY TO BE DRUG EFFECT COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURR	ED DUE TO QTcB CHANGED BY >60 MSEC FROM BASELINE COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AV SEQUENTIAL OR DUAL CHAMBER ELECTRONIC PACEMAKER IS SEEN COMPARED TO PREVIOUS ECG.	SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AV SEQUENTIAL OR DUAL CHAMBER ELECTRONIC PACEMAKER IS SEEN

Display 1. TEMP data set with truncated word

Methods to Derive COVAL-COVALn in CO Domain, continued

	ovar	coval	coval1	coval2
1	COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO CRITERIA FOR POSSIBLE ANTERIOR INFARCT GONE	COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO CRITERIA FOR POSSIBLE ANTERIOR INFARCT GONE		
2	COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO NON-SPECIFIC INTRA-VENTRICULAR CONDUCTION DELAY IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO NON-SPECIFIC INTRA-VENTRICULAR CONDUCTION DELAY IS SEEN	COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO NON-SPECIFIC INTRA-VENTRICULAR CONDUCTION DELAY IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO	NON-SPECIFIC INTRA-VENTRICULAR CONDUCTION DELAY IS SEEN	
3	POSSIBLE WOLFF-PARKINSON-WHITE COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN	POSSIBLE WOLFF-PARKINSON-WHITE COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED	DUE TO POSSIBLE WOLFF-PARKINSON-WHITE IS SEEN	
4	COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AGE UNDETERMINED. SEPTAL MI IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AGE UNDETERMINED. SEPTAL MI IS SEEN	COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AGE UNDETERMINED. SEPTAL MI IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AGE UNDETERMINED. SEPTAL MI	IS SEEN	
5	PROLONGED QT FULLY PACED BEAT. THE QT CHANGE AND QT PROLONGATION SHOULD BE CONSIDERED UNDER THESE CIRCUMSTANCES AND UNLIKELY TO BE DRUG EFFECT COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO QTCB CHANGED BY >80 MSEC FROM BASELINE COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AV SEQUENTIAL OR DUAL CHAMBER ELECTRONIC PACEMAKER IS SEEN COMPARED TO PREVIOUS ECG. SIGNIFICANT CHANGES	PROLONGED QT FULLY PACED BEAT. THE QT CHANGE AND QT PROLONGATION SHOULD BE CONSIDERED UNDER THESE CIRCUMSTANCES AND UNLIKELY TO BE DRUG EFFECT COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE	OCCURRED DUE TO QTCB CHANGED BY >80 MSEC FROM BASELINE COMPARED TO BASELINE ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AV SEQUENTIAL OR DUAL CHAMBER ELECTRONIC PACEMAKER IS SEEN COMPARED TO	PREVIOUS ECG. SIGNIFICANT CHANGES HAVE OCCURRED DUE TO AV SEQUENTIAL OR DUAL CHAMBER ELECTRONIC PACEMAKER IS SEEN

**Display 2. Methoda/methodb data set with intact word (CALL SCAN or INDEXC/REVERSE method)**

A COMPARE procedure was used to compare the outputs generated by two methods.

```
PROC COMPARE base=methoda compare=methodb;
RUN;
```

```
The COMPARE Procedure
Comparison of WORK.METHODA with WORK.METHODB (Method=EXACT)

Data Set Summary

Dataset              Created              Modified  NVar    NObs
WORK.METHODA       06MAR13:16:25:41    06MAR13:16:25:41    4       5
WORK.METHODB       06MAR13:16:25:41    06MAR13:16:25:41    4       5

Variables Summary

Number of Variables in Common: 4.

Observation Summary

Observation          Base  Compare
First Obs            1     1
Last Obs             5     5

Number of Observations in Common: 5.
Total Number of Observations Read from WORK.METHODA: 5.
Total Number of Observations Read from WORK.METHODB: 5.

Number of Observations with Some Compared Variables Unequal: 0.
Number of Observations with All Compared Variables Equal: 5.

NOTE: No unequal values were found. All values compared are exactly equal.
```

**Output 3. COMPARE OUTPUT between METHODDA and METHODDB**

**Compare summary:**

Methoda/ Methodb vs Temp (w/o truncated vs with truncated)

- a. Blue arrow in TEMP data set showed truncated value in COVAL1 variable; while word kept intact in Methoda/Methodb data set when using CALL SCAN or INDEXC/REVERSE macro;
- b. Red arrow showed 201<sup>th</sup> as blank space, one of the specified delimiters in two macros, the leading space kept in TEMP, while removed in Methoda/Methodb to avoid compliance findings;
- Methoda vs Methodb (CALL SCAN vs INDEXC/REVERSE method)
  - a. Both methods generated same outputs per PROC COMPARE procedure;
  - b. Both methods automatically decided how many do loops needed based on comments length , and automatically generated coval-covaln.
  - c. CALL SCAN take advantage of:
    - i. Negative count---instructs the CALL SCAN routine to scan from right to left;
    - ii. Last\_pos---searches first character of last available word in given text, returns position of that character in the string;
    - iii. Leading and trailing specified delimiters will be ignored if M modifier not used.
  - d. INDEXC/REVERSE take advantage of:
    - i. REVERSE---backforward writing of the argument;
    - ii. INDEXC---searches source from left to right, for the first occurrence of any character present in the excerpts and returns the position of that character in source.

## CONCLUSION

SDTM programmers often come across the situations when comment texts are longer than 200 characters. Based on CDISC standards, these long texts need to be split into multiple variables (COVAL-COVALn). This paper introduced two methods how to meaningfully split these long comment text. The main advantages are: a) meaningfully split long text, avoid splitting word into two meaningless character strings; b) automatically generate COVAL-COVALn based on comment's length in one single macro call. This paper also compared the outputs and found no difference between methods. Upon writing this paper, both methods have been tested successfully on small real-world databases, however, no data available as to which one is more efficient. Finally I would like to mention, it is not a requirement that words have to be left intact, so this may be done if sponsor prefers this approach.

## REFERENCES

- Study Data Tabulation Model Implementation Guide: Human Clinical Trials, Version 3.1.2. Published by CDISC November 12, 2008.
- SAS/BASE Software: Version 9, SAS® Institute Inc., Cary NC
- SAS/MACRO Software: Version 9.1.3, SAS® Institute Inc., Cary NC

## ACKNOWLEDGMENTS

Many thanks to Sandra VanPelt Nguyen and Mark Matthews in providing the valuable suggestions and support.

## CONTACT INFORMATION

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

Name: Chunxia Lin  
Enterprise: inVentiv Health Clinical  
Address: 4745 Haven Point Boulevard  
City, State ZIP: Indianapolis, IN 46280  
Work Phone: (914) 345-0680  
E-mail: chunxia.lin@inventivhealth.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

Other brand and product names are trademarks of their respective companies.