

Macro and Sample Source Code To Wrap Character Variable Text Conditionally on Two Lines Within DATA _Null_-generated Report Column(s).

Robert Andresen, Sergeantsville, New Jersey

ABSTRACT

The wrap macro requires knowledge of base SAS ® software Data _Null_ processing, SAS functions, and SAS Macro Language, and can be utilized anywhere that Base SAS and SAS Macro Language are available. Macro design assumes that report generation uses a DATA _Null_ to write one observation per line in multiple columns, and that some character variable values are accommodated on two lines, rather than one, within a column.

The programmer can store the macro in a macro library or copy it into a program. When invoking the macro, the programmer supplies parameters, such as the width of the column, the character variable to “wrap”, and whether to indent the second line. If the text cannot fit within the column, the macro sets a flag to so indicate, and divides the text between two programmer-supplied variables whose length equals the column width. After filling the first variable, the macro assigns the overflow (splitting the text at a space or dash) to the second variable. The programmer uses the macro flag value to conditionally write the observation variables on two report lines, instead of one, if necessary.

The macro can be invoked for multiple columns and their associated character variables within the same observation.

INTRODUCTION

If the length of the character variable value exceeds the report column width, the wrap macro splits the text as close to the end of the column as possible searching for programmer-supplied characters, such as a blank space, dash, or slash, at which to initiate the break. Two programmer-named variables store the initial text portion and the overflow, enabling the text to be “wrapped” onto two report lines within the same report column, instead of truncating the value. For multiple character variables and their associated columns, multiple invocations are possible simultaneously within the same observation. The macro design assumes that the longest value of the character variable can fit on two lines of a report column. Otherwise, truncation occurs. Implementing the macro into DATA _Null_ source code is straightforward.

Following is the macro source code, a description of the parameters passed to the macro upon invocation, a

sample data set, program, and output, as well as usage comments.

MACRO SOURCE CODE

```
%MACRO WRAP(WRAPFLG, LONGVAR, SPLIT_ON, WV1, WV2, COL_LEN,
            INDENT);
    LENGTH &WRAPFLG $1;
    ENDPOS=0;
    IF LENGTH(&LONGVAR) > &COL_LEN THEN
        DO;
            &WRAPFLG='Y';
            LENGTH &WV1 &WV2 $ &COL_LEN ;
            ENDPOS=&COL_LEN -
                INDXC(REVERSE(SUBSTR(&LONGVAR, 1, &COL_LEN)),
                    &SPLIT_ON);
            &WV1=SUBSTR(&LONGVAR, 1, ENDPOS);
            IF "&INDENT"="Y" THEN
                &WV2=' '||LEFT(SUBSTR(&LONGVAR, ENDPOS+1));
            ELSE &WV2=LEFT(SUBSTR(&LONGVAR, ENDPOS+1));
        END;
%MEND;  %*WRAP(WRAPFLG, LONGVAR, SPLIT_ON, WV1, WV2,
              COL_LEN, INDENT);
```

Macro positional parameters are :

WRAPFLG	- New variable : 'Y' flags that LONGVAR value will be wrapped
LONGVAR	- the variable or constant to be wrapped
SPLIT_ON	- character value(s) at which text is split
WV1	- name of new variable holding the first wrap line
WV2	- name of new variable holding the second wrap line
COL_LEN	- width of column
INDENT	- whether to indent second line (WV2), Y=YES

Non-macro variable is :

ENDPOS	- End position. Last argument of SUBSTR function
--------	--------------------------------------------------

Note :

- 1) wrapflg, wv1, & wv2 variable names should not be the names of existing variables, otherwise data values can be overwritten.
- 2) if the indent option is used, the use of SAS \$CHARW. format in the PUT statement insures that the leading blank in the 2nd line (wv2) prints
- 3) for multiple invocations of wrap in a report, use different variable names for wrapflg, wv1, & wv2 for

each invocation so that a unique wrapflg variable associates with each variable to be wrapped

SAMPLE DATA

```
DATA TST;
  INPUT CV $CHAR41.;
  CV2=CV;
  CV3=CV;
CARDS;
NOW IS THE TIME FOR ALL GOOD MEN
A QUICK BROWN FOX JUMPS OVER THE LAZY DOG
COCA-COLA HITS THE SPOT
A STITCH IN TIME SAVES NINE
WE HAVE NOTHING TO FEAR - BUT FEAR ITSELF
ONE TWO THREE FOUR FIVE SIX
IN GOD WE TRUST
123 W 55TH STREET, SUITE 7000
ANTIDISESTABLISHMENTARIANISM
;
RUN;
```

SAMPLE SOURCE CODE

```
DATA _NULL_;
  SET TST END=EOF;
  FILE PRINT;
  RETAIN COL1 5 COL2 19 COL3 39 COL4 67 ;
  %WRAP(WRAPFLG,CV," -/",WVA1,WVA2,15,Y);
  %WRAP(WRAPFLG2,CV2," -/",WVB1,WVB2,20,);
  %WRAP(WRAPFLG3,CV3," -/",WVC1,WVC2,25,Y);

  IF _N_=1 THEN
  DO;
    PUT @COL2 'PRINTING REPORT COLUMNS USING WRAP
  MACRO';
    PUT @COL2 '-----';
  --';
    PUT;
    PUT @ (COL2-3) '| ' @ (COL3-3) '| ' @ (COL4-3)
  '| ' @;
    PUT @COL2 'COL WIDTH 15'
      @COL3 'COL WIDTH 20'
```

```
      @COL4 'COL WIDTH 25' ;
    PUT @ (COL2-3) '| ' @ (COL3-3) '| ' @ (COL4-3)
  '| ' @;
    PUT @COL1 'OBS' ;
      @COL2 '(INDENT)'
      @COL3 '(NO INDENT)'
      @COL4 '(INDENT)';
    PUT @ (COL2-3) '| ' @ (COL3-3) '| ' @ (COL4-3)
  '| ' @;
    PUT @COL1 6*'- '
      @COL2 15*'- '
      @COL3 20*'- '
      @COL4 25*'- ' ;
  END;
  PUT @ (COL2-3) '| ' @ (COL3-3) '| ' @ (COL4-3) '| ' ;
  IF WRAPFLG = 'Y' OR WRAPFLG2 = 'Y' OR WRAPFLG3 = 'Y'
  THEN
  DO;
    PUT @ (COL2-3) '| ' @ (COL3-3) '| ' @ (COL4-3)
  '| ' @;
    IF WRAPFLG = 'Y' THEN PUT @COL2 WVA1 $CHAR15. @;
    IF WRAPFLG2 = 'Y' THEN PUT @COL3 WVB1 $CHAR20. @;
    IF WRAPFLG3 = 'Y' THEN PUT @COL4 WVC1 $CHAR25. @;
    PUT;
    IF WRAPFLG = 'Y' THEN PUT @COL2 WVA2 $CHAR15. @;
    ELSE PUT @COL2 CV $CHAR15. @;
    IF WRAPFLG2 = 'Y' THEN PUT @COL3 WVB2 $CHAR20. @;
    ELSE PUT @COL3 CV2 $CHAR20. @;
    IF WRAPFLG3 = 'Y' THEN PUT @COL4 WVC2 $CHAR25. @;
    ELSE PUT @COL4 CV3 $CHAR25. @;
    PUT @ (COL2-3) '| ' @ (COL3-3) '| ' @ (COL4-3)
  '| ' @;
    PUT @COL1 _N_ WORDS6. ;
  END;
  ELSE
  DO;
    PUT @ (COL2-3) '| ' @ (COL3-3) '| ' @ (COL4-3)
  '| ' @;
    PUT @COL1 _N_ WORDS6.
      @COL2 CV $CHAR15.
      @COL3 CV2 $CHAR20.
      @COL4 CV3 $CHAR41. ;
  END;
  RUN;
```

SAMPLE OUTPUT

The SAS System
PRINTING REPORT COLUMNS USING WRAP MACRO

OBS	COL WIDTH 15 (INDENT)	COL WIDTH 20 (NO INDENT)	COL WIDTH 25 (INDENT)
one	NOW IS THE TIME FOR ALL G	NOW IS THE TIME FOR ALL GOOD MEN	NOW IS THE TIME FOR ALL GOOD MEN
two	A QUICK BROWN FOX JUMPS OVER	A QUICK BROWN FOX JUMPS OVER THE LAZY	A QUICK BROWN FOX JUMPS OVER THE LAZY DOG
three	COCA-COLA HITS THE SPOT	COCA-COLA HITS THE SPOT	COCA-COLA HITS THE SPOT
four	A STITCH IN TIME SAVES NIN	A STITCH IN TIME SAVES NINE	A STITCH IN TIME SAVES NINE
five	WE HAVE NOTHING TO FEA	WE HAVE NOTHING TO FEAR - BUT FEAR ITSE	WE HAVE NOTHING TO FEAR - BUT FEAR ITSELF
six	ONE TWO THREE FOUR FIVE SIX	ONE TWO THREE FOUR FIVE SIX	ONE TWO THREE FOUR FIVE SIX

seven	IN GOD WE TRUST	IN GOD WE TRUST	IN GOD WE TRUST
eight	123 W 55TH STREET, SUITE	123 W 55TH STREET, SUITE 7000	123 W 55TH STREET, SUITE 7000
nine	ANTIDISESTABLIS HMENTARIANISM	ANTIDISESTABLISHMENT ARIANISM	ANTIDISESTABLISHMENTARIAN ISM

USAGE NOTES

1. Observation three exemplifies wrapping of "COCA-COLA HITS THE SPOT" in the first two columns, but not in the third column, which can accommodate the full text without wrapping to a second line.
2. Observation seven, "IN GOD WE TRUST", fits without wrapping in all three columns, and consequently requires only one report line.
3. "ANTIDISESTABLISHMENTARIANISM", observation nine, contains none of the split characters, and consequently is wrapped at the end of the column width.
4. note that in column 1 (width=15), many of the text values are longer than thirty characters. Thus, text truncation occurs after the second report line fills.
5. Although not shown in the example, when "lines left" or "line" values trigger page breaks, the value may need to be decreased by one, recognizing that a single observation may generate two report lines, instead of one.
6. A program that produces address labels, could use the wrap macro to write two street address lines conditionally, using a comma as the split character. Observation eight, for example, would wrap "SUITE 7000" onto line two.

® SAS software is a registered trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Robert Andresen, Seven Lambert Road, Stockton, New Jersey 08559, (609) 397 - 8054.