An FSEDIT Screen Building Program to Allow Multiple Observations Per Screen

Judith A. Burns, Thomas R. Hoffman, Leon T. Hairie
Lederle Laboratories

ABSTRACT

Currently the SAS/FSP FSEDIT procedure allows the user to edit only one observation per screen. For some applications it is more convenient to edit and/or browse a data set when all observations in a BY group appear together on one screen. This can be accomplished by first writing code to create a horizontal data set that contains one observation for each BY group and next rebuilding the FSEDIT screen data set. This latter step requires the user to format the screen and inform FSEDIT of the location of each variable on the screen by moving the cursor to the proper location and pressing the 'ENTER' key. For data sets with a large number of variables, this procedure can be quite time consuming.

This paper describes an alternative procedure. A program has been written that 1) builds the code to horizontalize the data set, 2) creates the new screen dataset, and 3) builds the code to restore the dataset to its original vertical form. The only input required by the program are the variable names that identify a BY group, the maximum number of observations to be displayed on a screen, and a screen dataset that identifies the location of the first observation. Once the screen dataset and transformation code are built, a single macro command can be used to edit or browse the dataset.

BACKGROUND

Consider the dataset SAVE.CONMED containing the following information:

Variable	Length	Format	Description
PT	3	4.	Patient number
DAY	3	4.	Days on treatment
RX	3	RX.	Concomitant medication
DUR	3	4.	Duration (in days)
RXIND	3	RXIND.	Indication

The PRINT procedure produces the following output:

```
PT=3001

DAY      RX      DUR RXIND

168 PREDNISONE    4 ARTHRITIS
186 ACHROMYCIN    7 U R I
210 PREDNISONE   84 ARTHRITIS
233 ANTIVERT     46 DIZZINESS/VERTIGO


PT=3002

DAY RX                      DUR RXIND

 30 PREDNISONE               27 ARTHRITIS
 39 DARVON COMPOUND          21 PAIN
141 MYLANTA                 122 NAUSEA
142 PREDNISONE               2D ARTHRITIS
319 MYLANTA                  92 NAUSEA
483 WINSTROL                111 WEIGHT REDUCTION
500 MYLANTA                  64 INDIGESTION
582 ARISTOCORT XYLO INJ       1 ARTHRITIS
```

If the print output has been directed to an external file, then the FSLIST procedure can be used to browse the dataset. In fact, in this example, FSLIST has two advantages over FSBROWSE: 1) all data for one patient appear together 2) the FIND command can be used to locate character data in the RX and RXIND fields.

However, to edit the dataset requires the FSEDIT procedure. In this case the default edit screen appears as follows:

```
PT:    3001
DAY:   188
RX:    PREDNISONE
DUR:   4
RXIND: ARTHRITIS
```

Only one observation is shown per screen. This structure is acceptable for locating and changing a particular observation, but is not convenient for reviewing the data.

In order to display multiple observations per screen, the dataset must be restructured so that each observation contains several concomitant medications. The code that creates this horizontal dataset follows:

```
DATA HDS;SET SAVE.CONMED END=EOF;
     BY PT ;
      I_ + 1;
     KEEP PT ;
     LENGTH DAY1 - DAY10 3 ;
     ARRAY _DAY DAY1 - DAY10 ;
      DAY = DAY ;
     KEEP DAY1 - DAY10 ;
     RETAIN DAY1 - DAY10 ;
     FORMAT DAY1 - DAY10 4. ;
     LENGTH RX1 - RX10 3 ;
     ARRAY _RX RX1 - RX10 ;
      RX = RX ;
     KEEP RX1 - RX10 ;
     RETAIN RX1 - RX10 ;
     FORMAT RX1 - RX10 RX26. ;
     LENGTH DUR1 - DUR10 3 ;
     ARRAY _DUR DUR1 - DUR10 ;
      DUR = DUR ;
     KEEP DUR1 - DUR10 ;
     RETAIN DUR1 - DUR10 ;
     FORMAT DUR1 - DUR10 4. ;
     LENGTH RXIND1 - RXIND10 3 ;
     ARRAY _RXIND RXIND1 - RXIND10 ;
      RXIND = RXIND ;
     KEEP RXIND1 - RXIND10 ;
     RETAIN RXIND1 - RXIND10 ;
     FORMAT RXIND1 - RXIND10 RXIND25. ;
     IF _I_=10 OR LAST.PT THEN
     DO;
        OUTPUT;
        ARRAY _CHAR_ _CHARACTER_;
        ARRAY _NUM_ _NUMERIC_;
        DO OVER _CHAR_; _CHAR_='';END;
        DO OVER _NUM_;_NUM_=.;END;
        I_ = 0;
     END;
RUN;
```

The array size should be chosen large enough to meet future needs. Note that all variables must be initialized after the output statement.

Figure 1 shows the default FSEDIT screen for dataset HDS. After some time in screen modification mode, this screen can be rearranged as shown in Figure 2.

After editing dataset HDS, the following code is necessary to reconstruct dataset SAVE.CONMED.

```
DATA VDS;SET HDS;
     RETAIN _FLAG_;
     KEEP PT ;
     KEEP DAY ;
     ARRAY _DAY 3 DAY1 - DAY10 ;
     DO OVER _DAY ;
     DAY = _DAY ;
     IF DAY NE . THEN _FLAG_=1;
     KEEP RX ;
     ARRAY _RX 3 RX1 - RX10 ;
     RX = _RX ;
     IF RX NE . THEN _FLAG_=1;
     KEEP DUR ;
     ARRAY _DUR 3 DUR1 - DUR10 ;
     DUR = _DUR ;
     IF DUR NE . THEN _FLAG_=1;
     KEEP RXIND ;
     ARRAY _RXIND 3 RXIND1 - RXIND10 ;
     RXIND = _RXIND ;
     IF RXIND NE . THEN _FLAG_=1;
     IF _FLAG_ THEN OUTPUT;
     _FLAG_=0;
     END;
PROC SORT DATA=VDS;BY PT;
DATA SAVE.CONMED;SET SAVE.CONMED;STOP;
PROC APPEND BASE=SAVE.CONMED DATA=VDS FORCE;
RUN;
```

Although the procedure just described for displaying multiple observations per FSEDIT screen works, it is time consuming and error prone. For example, forgetting to initialize the values in dataset HDS could result in adding some concomitant medications to the next patient.

The alternative is to generate the code for transforming the dataset and to modify the screen dataset in a data step rather than FSEDIT.

BUILDING THE TRANSFORMATION CODE

Assume that the user has been prompted for the following macro variables:

&DATA	SAS dataset name
&SORT	List of sort variables
&BYGROUP	Last sort variable that defines by group
&MAXOBS	Maximum number of observations
&SDS	Screen dataset name

A dataset VARS containing the variables VAR, LENGTH, FORMAT, and INFORMAT, can be created by reading the output from the CONTENTS procedure using macro READCNT:

```
***; %MACRO READCNT(DATA=_LAST_);
PROC PRINTTO UNIT=40 NEW;
PROC CONTENTS DATA=&DATA POSITION NOSOURCE;
PROC PRINTTO;
DATA VARS;
  LENGTH VAR $8 LENGTH $4 FORMAT $15
         INFORMAT $15 LABEL $40;
  INFILE FT40F001 MISSOVER COL=COL;
  KEEP VAR--LABEL;
  INPUT @2 NUM $ VAR TYP $ @;
  IF NUM='#' AND VAR='VARIABLE' THEN DO;
     FLAG=1;
     INPUT LEN $ POS $ FOR $ @;
     FCOL=COL-7;
     INPUT INF $ @;
     ICOL=COL-9;
     INPUT LAB $;
     IF LAB='LABEL' THEN LCOL=COL-6;
     ELSE LCOL=.;
     RETAIN FLAG FCOL ICOL LCOL;
     RETURN;
  END;
  IF NUM='ALPHABET' AND VAR='LIST' THEN STOP;
  IF FLAG AND (TYP='NUM' OR TYP='CHAR') THEN DO;
     INPUT LENGTH POS $ @FCOL FORMAT $CHAR15.
       @ICOL INFORMAT $CHAR15. @;
     IF TYP='CHAR' THEN LENGTH='$'!!LENGTH;
     IF FORMAT=:' ' THEN FORMAT='';
     IF INFORMAT=:' ' THEN INFORMAT='';
     IF LCOL NE . THEN INPUT @LCOL LABEL $40.;
     ELSE DO;
        INPUT;
        INPUT @2 LABEL &;
     END;
     OUTPUT;
  END;
  OPTION NONOTES;RUN;OPTION NOTES;
***; %MEND READCNT;
```

Be sure to first allocate FT40F001. Note that this macro does not change the user's linesize and pagesize options.

Also assume that FILESAS1 and FILESAS2 are ddnames that have been previously allocated. The following two macros can be used to generate the vertical to horizontal and horizontal to vertical transformation code.

```
***; %MACRO HDS;
DATA _NULL_;SET VARS END=EOF;
  FILE FILESAS1 OLD;
  RETAIN LVAR EXPAND;
  LVAR="&BYGROUP";
  IF %LENGTH(&SORT)=0 THEN
     SORT="&BYGROUP NOTSORTED";
  ELSE SORT="&SORT";
  IF LVAR='' THEN EXPAND=1;
  IF _N_=1 THEN DO;
     PUT
  "DATA HDS;SET &DATA END=EOF;                    ";
     IF LVAR NE '' THEN PUT
  "     " "BY " SORT ";                           ";
     PUT
  "     _I_ + 1;                                  ";
  END;
  IF EXPAND THEN DO;
     VAR8=SUBSTR(VAR,1,6);
     VAR1=COMPRESS(VAR8!!'1');
     VAR2=COMPRESS(VAR8!!"&MAXOBS");
     PUT
  "     LENGTH " VAR1 "- " VAR2 LENGTH ";         "/
  "     ARRAY  " VAR8       VAR1 "- " VAR2 ";     "/
  "       " VAR8 "= " VAR ";                      "/
  "     KEEP " VAR1 "- " VAR2 ";                  "/
  "     RETAIN "  VAR1 "- " VAR2 ";               ";
     IF FORMAT NE '' THEN PUT
  "     FORMAT " VAR1 "- " VAR2 FORMAT ";         ";
     IF INFORMAT NE '' THEN PUT
  "     INFORMAT " VAR1 "- " VAR2 INFORMAT ";     ";
  END;
  ELSE PUT
  "     KEEP " VAR ";                             ";
  IF VAR=LVAR THEN EXPAND=1;
  IF EOF THEN DO;
     IF LVAR NE '' THEN PUT
  "     " "IF _I_=&MAXOBS OR LAST.&BYGROUP THEN "/
  "     DO;                                       ";
     ELSE PUT
  "     " "IF _I_=&MAXOBS OR END=EOF THEN DO;     ";
     PUT
  "       OUTPUT;                                 "/
  "       ARRAY _CHAR_ _CHARACTER_;               "/
  "       ARRAY _NUM_ _NUMERIC_;                  "/
  "       DO OVER _CHAR_; _CHAR_='';END;          "/
  "       DO OVER _NUM_; _NUM_=.;END;             "/
  "       _I_ = 0;                                "/
  "     END;                                      "/
  "RUN;                                           ";
  END;
RUN; %INC FILESAS1/NOSOURCE2;
***; %MEND HDS;

***; %MACRO VDS;
DATA _NULL_;SET VARS END=EOF;
  FILE FILESAS2 OLD;
  RETAIN LVAR EXPAND;
  LVAR="&BYGROUP";
  IF LVAR='' THEN EXPAND=1;
  IF _N_=1 THEN PUT
  "DATA VDS;SET HDS;                              "/
  "     RETAIN _FLAG_;                            ";
  PUT
  "     KEEP " VAR ";                             ";
```

```
  IF EXPAND THEN DO;
     CNT+1;
     VAR6=SUBSTR(VAR,1,6);
     VAR1=COMPRESS(VAR6!!'1');
     VAR2=COMPRESS(VAR6!!"&MAXOBS");
     PUT
  "     ARRAY _" VAR6 LENGTH VAR1 "- " VAR2 ";   ";
  IF CNT=1 THEN PUT
  "     DO OVER _" VAR6 ";                       ";
  PUT
  "     "VAR "=_" VAR6 ";                        ";
  IF LENGTH='$' THEN PUT
  "     IF " VAR " NE '' THEN _FLAG_=1;          ";
  ELSE PUT
  "     IF " VAR " NE . THEN _FLAG_=1;           ";
  END;
  IF VAR=LVAR THEN EXPAND=1;
  IF EOF THEN DO;
     PUT
  "     IF _FLAG_ THEN OUTPUT;                   "/
  "     _FLAG_=0;                                "/
  "     END;                                     ";
     IF %LENGTH(&SORT) > 0 THEN PUT
  "PROC SORT DATA=VDS;BY &SORT;                  ";
     PUT
  "DATA &DATA;SET &DATA;STOP;                    "/
  "PROC APPEND BASE=&DATA DATA=VDS FORCE;        "/
  "RUN;                                          ";
  END;
RUN; %INC FILESAS2/NOSOURCE2;
***; %MEND VDS;
```

MODIFYING A SCREEN DATASET

Before using a data step to modify a screen dataset, some understanding of its structure is necessary. The output from printing the screen dataset for SAVE.CONMED(after some modification) is shown in Figure 3. Figure 4 lists the final screen dataset used for formatting dataset HDS. It can be seen that these datasets contain six different types of records:

Record	TYPE#	Information in TEXT#
1	1000	Number of screens, rows/screen
2	>1000	Text format
3	0	Variable location
4	1	Character string?
5	2	Variable type, length, and screen attributes
6	99	End of file

It should be noted that the screen dataset shown in Figure 4 is the "expanded" version of the one shown in Figure 3.

The code to expand the screen dataset is shown in Figure 5.

The special variables _NUMERIC_ and _CHARACTER_ were necessary since TYPE# and TEXT# are not valid SAS names. The program assumes that the first six characters of variables requiring expansion are unique. The program also assumes that all rows following the last row of text (see TYPE#=1008) should be modified. This is the correct assumption if the variables in the edit dataset are properly ordered. If necessary, additional screens are added,in which case the heading information is repeated on the additional screens. If the macro variable MAXOBS is blank, the program sets MAXOBS equal to number of rows available on the first screen. The program also will handle the case where the variables to be expanded require more than one row.

USING THE PROGRAM

The program can be driven by a macro that contains %PUT and %INPUT statements to prompt the user for the required information. Consider the macro BUILDSCR:

```
%MACRO BUILDSCR;
OPTION NOCAPS;
%PUT ENTER THE NAME OF YOUR SAS DATASET;
%INPUT DATA;
%PUT DO YOU WANT TO DISPLAY DATA IN GROUPS?;
%PUT Y OR N;
%INPUT YES;
%IF &YES=Y %THEN %DO;
%PUT HOW IS THE DATASET SORTED? ;
%INPUT SORT;
%PUT WHICH ONE OF THESE VARIABLES IDENTIFIES;
%PUT THE DISPLAY GROUP?;
%INPUT BYGROUP;
%PUT ENTER MAXIMUM NUMBER OF OBSERVATIONS THAT YOU;
%PUT WISH TO DISPLAY PER GROUP;
%INPUT MAXOBS;
%END;
%PUT WHAT DO YOU WANT TO NAME YOUR;
%PUT SAS SCREEN DATASET?;
%INPUT SDS;
PROC FSEDIT OPT=6 DATA=&DATA SCREEN=&SDS;RUN;
%MODSCR;%READCNT(DATA=&DATA);%HDS;
PROC FSEDIT OPT=6 DATA=HDS SCREEN=&SDS;RUN;
%MEND BUILDSCR;
```

This is a 'no frills' prompting routine. Additional assistance could be provided. When the macro command %BUILDSCR is entered, the user is first prompted for the necessary macro variables. The program then invokes FSEDIT in screen modification mode. The user must format the heading information and the first row of data. If desired, the user also can add attributes to the variables on the screen.

When the user exits from FSEDIT, the program expands the screen, builds the vertical to horizontal code and returns to FSEDIT. At this time the user can make any final changes to the screen dataset.

After leaving FSEDIT, macro VDS can be called in order to build the horizontal to vertical code.

After the code and screen are built and saved, the following macro could be used to edit the CONMED dataset:

```
%MACRO ECONMED;
%HCONMED;
PROC FSEDIT OPT=1 DATA=HDS SCREEN=SAVE.S_CONMED;
%VCONMED;
%MEND ECONMED;
```

Macros HCONMED and VCONMED contain the transformation code that was saved in files FILESAS1 and FILESAS2, respectively.

For additional information contact:

Dr. T. R. Hoffman
Lederle Laboratories
Pearl River, NY 10965

Figure 1

```
                              EDIT SAS DATA SET: WORK.HDS                  | SCREEN 1
COMMAND ===>                                                               |---------
                                                                           | OBS    1
-------------------------------------------------------------------------------------
PT:      3001          DAY1:    168           DAY2:    186
DAY3:    210           DAY4:    233           DAY5:    ____
DAY6:    ____          DAY7:    ____          DAY8:    ____
DAY9:                  DAY10:   ____
RX1:     PREDNISONE
RX2:     ACHROMYCIN
RX3:     PREDNISONE
RX4:     ANTIVERT
RX5:     ____________________________
RX6:     ____________________________
RX7:     ____________________________
RX8:     ____________________________
RX9:     ____________________________
RX10:    ____________________________  DUR1:    4
DUR2:    7             DUR3:    84     DUR4:    46
DUR5:    ____          DUR6:    ____   DUR7:    ____
DUR8:                  DUR9:    ____   DUR10:   ____
RXIND1:  ARTHRITIS
RXIND2:  U R I
RXIND3:  ARTHRITIS
```

Figure 2

```
                              EDIT SAS DATA SET: WORK.HDS                  | SCREEN 1
COMMAND ===>                                                               |---------
                                                                           | OBS    1
-------------------------------------------------------------------------------------
PT: 3001

DAY   DRUG                        DURATION  DRUG INDICATION
====  ==========================  ========  ========================
168   PREDNISONE                     4      ARTHRITIS
186   ACHROMYCIN                     7      U R I
210   PREDNISONE                    84      ARTHRITIS
233   ANTIVERT                      46      DIZZINESS/VERTIGO
```

Figure 3

```
OBS TYPE# TEXT#

  1 1000  VER001    80  1 24 80
  2 1005  PT:
  3 1007  DAY  DRUG                          DURATION  DRUG INDICATION
  4 1008  ====  ==========================   ========  ========================
  5 1009  ____  __________________________   ____      ________________________
  6    0  PT          1  5  6  5  9
  7    0  DAY         1  9  2  9  5
  8    0  RX          1  9  8  9 33
  9    0  DUR         1  9 38  9 41
 10    0  RXIND       1  9 46  9 71
 11    1  OHSEOXDBFALRCHSEOXDBFALRCOHSEIDXBFOLRCHSEIDXBFOLRC
 12    2  PT          3 CGRHH_      0
 13    2  DAY         3 CGR H       0
 14    2  RX          3 CGR H       0
 15    2  DUR         3 CGR H       0
 16    2  RXIND       3 CGR H       0
 17   99  END OF FILE
```

Figure 4

OBS	TYPE#	TEXT#						
1	1000	VER001	80	1	24	80		
2	1005	PT:						
3	1007	DAY	DRUG				DURATION	DRUG INDICATION
4	1008	****	==========				========	==========
5	1009	____	_____				____	_____
6	1010	____	_____				____	_____
7	1011	____	_____				____	_____
8	1012	____	_____				____	_____
9	1013	____	_____				____	_____
10	1014	____	_____				____	_____
11	1015	____	_____				____	_____
12	1016	____	_____				____	_____
13	1017	____	_____				____	_____
14	1018	____	_____				____	_____
15	0	PT		1	5	6	5	9
16	0	DAY1		1	9	2	9	5
17	0	DAY2		1	10	2	10	5
18	0	DAY3		1	11	2	11	5
19	0	DAY4		1	12	2	12	5
20	0	DAY5		1	13	2	13	5
21	0	DAY6		1	14	2	14	5
22	0	DAY7		1	15	2	15	5
23	0	DAY8		1	16	2	16	5
24	0	DAY9		1	17	2	17	5
25	0	DAY10		1	18	2	18	5
26	0	RX1		1	9	8	9	33
27	0	RX2		1	10	8	10	33
28	0	RX3		1	11	8	11	33
29	0	RX4		1	12	8	12	33
30	0	RX5		1	13	8	13	33
31	0	RX6		1	14	8	14	33
32	0	RX7		1	15	8	15	33
33	0	RX8		1	16	8	16	33
34	0	RX9		1	17	8	17	33
35	0	RX10		1	18	8	18	33
36	0	DUR1		1	9	38	9	41
37	0	DUR2		1	10	38	10	41
38	0	DUR3		1	11	38	11	41
39	0	DUR4		1	12	38	12	41
40	0	DUR5		1	13	38	13	41
41	0	DUR6		1	14	38	14	41
42	0	DUR7		1	15	38	15	41
43	0	DUR8		1	16	38	16	41
44	0	DUR9		1	17	38	17	41
45	0	DUR10		1	18	38	18	41
46	0	RXIND1		1	9	46	9	71
47	0	RXIND2		1	10	46	10	71
48	0	RXIND3		1	11	46	11	71
49	0	RXIND4		1	12	46	12	71
50	0	RXIND5		1	13	46	13	71
51	0	RXIND6		1	14	46	14	71
52	0	RXIND7		1	15	46	15	71
53	0	RXIND8		1	16	46	16	71
54	0	RXIND9		1	17	46	17	71
55	0	RXIND10		1	18	46	18	71
56	1	OHSEOXDBFALRCHSEOXDBFALRCOHSEIDXBFOLRCHSEIDXBFOLRC						
57	2	PT		3	CGRHH		0	
58	2	DAY1		3	CGR H		0	
59	2	DAY2		3	CGR H		0	
60	2	DAY3		3	CGR H		0	
61	2	DAY4		3	CGR H		0	
62	2	DAY5		3	CGR H		0	
63	2	DAY6		3	CGR H		0	
64	2	DAY7		3	CGR H		0	
65	2	DAY8		3	CGR H		0	
66	2	DAY9		3	CGR H		0	
67	2	DAY10		3	CGR H		0	
68	2	RX1		3	CGR H		0	
69	2	RX2		3	CGR H		0	
70	2	RX3		3	CGR H		0	
71	2	RX4		3	CGR H		0	
72	2	RX5		3	CGR H		0	
73	2	RX6		3	CGR H		0	
74	2	RX7		3	CGR H		0	
75	2	RX8		3	CGR H		0	
76	2	RX9		3	CGR H		0	
77	2	RX10		3	CGR H		0	
78	2	DUR1		3	CGR H		0	
79	2	DUR2		3	CGR H		0	
80	2	DUR3		3	CGR H		0	
81	2	DUR4		3	CGR H		0	
82	2	DUR5		3	CGR H		0	
83	2	DUR6		3	CGR H		0	
84	2	DUR7		3	CGR H		0	
85	2	DUR8		3	CGR H		0	
86	2	DUR9		3	CGR H		0	
87	2	DUR10		3	CGR H		0	
88	2	RXIND1		3	CGR H		0	
89	2	RXIND2		3	CGR H		0	
90	2	RXIND3		3	CGR H		0	
91	2	RXIND4		3	CGR H		0	
92	2	RXIND5		3	CGR H		0	
93	2	RXIND6		3	CGR H		0	
94	2	RXIND7		3	CGR H		0	
95	2	RXIND8		3	CGR H		0	
96	2	RXIND9		3	CGR H		0	
97	2	RXIND10		3	CGR H		0	
98	99	END OF FILE						

Figure 5

```
***; %MACRO MODSCR;
DATA &SDS;
  SET &SDS;
  ARRAY _TY _NUMERIC_;
  ARRAY _TX _CHARACTER_;
  _I_=1;
  ARRAY TEXT(_J_) $80 TEXT1-TEXT10;
  ARRAY SAVE(_K_) $80 SAVE1-SAVE10;
  DROP TEXT1-TEXT10 SAVE1-SAVE10 TY1000 _J_ _K_
    MAXROW START BLKSZE NSCREENS LROW COL1 COL2
    VAR ROW CNT FIRSTROW TEXTSZE HEADSZE ROWS1
    TOTROWS ROWSX A END I J NEWROW NEWVAR DONE
    MAXOBS;
  RETAIN TY1000 _J_ TEXT1-TEXT10 SAVE1-SAVE10 MAXROW
    START BLKSZE ROWS1 ROWSX NSCREENS LROW TEXTSZE
    MAXOBS;
  LENGTH MAXROW NSCREENS COL1 COL2 MAXOBS 2;
  TEXTSZE=%LENGTH(&BYGROUP);
  IF _TY=1000 THEN DO;
     MAXROW=INPUT(SUBSTR(_TX,17,2),2.)-4;
     TY1000=_TX;
     START=MAXROW;
     LROW=MAXROW;
  END;
  ELSE IF _TY > 1000 THEN DO;
      _J_=MOD(_TY,1000)-4;
     TEXT=_TX;
  END;
  ELSE IF _TY=0 THEN DO;
     VAR=SUBSTR(_TX,1,6);
     ROW=INPUT(SUBSTR(_TX,14,2),2.)-4;
     COL1=INPUT(SUBSTR(_TX,17,2),2.);
     COL2=INPUT(SUBSTR(_TX,23,2),2.);
     CNT+1;
     IF CNT < START AND
        (ROW-LROW > 1 OR TEXTSZE=0)
        THEN DO;
        START=CNT;
        FIRSTROW=ROW;
        BLKSZE=_J_-FIRSTROW+1;
        IF TEXTSZE > 0 THEN DO;
           HEADSZE=ROW-LROW-1;
           TEXTSZE=FIRSTROW-HEADSZE-1;
           DO _J_=TEXTSZE+1 TO FIRSTROW-1;
              IF TEXT='' THEN DO;
                   TEXTSZE=TEXTSZE+1;
                   HEADSZE=HEADSZE-1;
              END;
              IF TEXT NE '' THEN _J_=FIRSTROW-1;
           END;
        END;
        ELSE HEADSZE=FIRSTROW-1;
        ROWS1=(MAXROW-FIRSTROW+1)-
          MOD(MAXROW-FIRSTROW+1,BLKSZE);
        %IF %LENGTH(&MAXOBS)=0 %THEN %DO;
            CALL SYMPUT('MAXOBS',PUT(ROWS1,2.));
            MAXOBS=ROWS1;
        %END;
        %ELSE %DO;MAXOBS=&MAXOBS;%END;
        TOTROWS=MAXOBS*BLKSZE;

        IF TOTROWS>ROWS1 THEN DO;
           ROWSX=(MAXROW-HEADSZE)-
             MOD(MAXROW-HEADSZE,BLKSZE);
           NSCREENS=1+CEIL((TOTROWS-ROWS1)/ROWSX);
           _TY=1000;
           SUBSTR(TY1000,14,2)=PUT(NSCREENS,2.);
           _TX=TY1000;
           OUTPUT;
        END;
        ELSE DO;
           NSCREENS=1;
           _TY=1000;
           _TX=TY1000;
           OUTPUT;
        END;
        DO _J_=1 TO FIRSTROW-1;
           IF TEXT NE '' THEN DO;
              _TY=1000+_J_+4;
              _TX=TEXT;
              OUTPUT;
           END;
        END;
        DO A=1 TO NSCREENS;
           IF A=1 THEN
             END=MIN(ROWS1/BLKSZE,MAXOBS);
           ELSE DO;
              END=MIN(ROWSX/BLKSZE,MAXOBS-DONE);
              DO _J_=TEXTSZE+1 TO TEXTSZE+HEADSZE;
                 _TY=A*1000+_J_-TEXTSZE+4;
                 _TX=TEXT;
                 OUTPUT;
              END;
           END;
           DONE+END;
           DO _J_=FIRSTROW TO FIRSTROW+BLKSZE-1;
              _TX=TEXT;
              DO I=1 TO END;
                 _TY=1000*A+_J_+4+(I-1)*BLKSZE-
                   (A>1)*TEXTSZE;
                 OUTPUT;
              END;
           END;
        END;
        _TY=0;DONE=0;
        DO _K_=1 TO START-1;
           _TX=SAVE;
           OUTPUT;
        END;
     END;
     LROW=ROW;
     IF CNT < START THEN DO;
        _K_+1;
        SAVE=_TX;
        RETURN;
     END;
     DO A=1 TO NSCREENS;
        IF A=1 THEN END=MIN(ROWS1/BLKSZE,MAXOBS);
        ELSE END=MIN(ROWSX/BLKSZE,MAXOBS-DONE);
        DONE+END;
        DO I=1 TO END;
           J+1;
           NEWROW=ROW+(I-1)*BLKSZE+4-(A>1)*TEXTSZE;
           NEWVAR=COMPRESS(VAR!!PUT(J,2.));
           _TX=PUT(NEWVAR,$8.)!!PUT(A,4.)!!
               PUT(NEWROW,3.)!!PUT(COL1,3.)!!
               PUT(NEWROW,3.)!!PUT(COL2,3.);
           OUTPUT;
        END;
     END;
     J=0;DONE=0;
  END;
  ELSE IF _TY=1 THEN DO;CNT=0;DONE=0;OUTPUT;END;
  ELSE IF _TY=2 THEN DO;
     CNT+1;
     IF CNT<START THEN DO;OUTPUT;RETURN;END;
     VAR=SUBSTR(_TX,1,6);
     DO A=1 TO NSCREENS;
        IF A=1 THEN END=MIN(ROWS1/BLKSZE,MAXOBS);
        ELSE END=MIN(ROWSX/BLKSZE,MAXOBS-DONE);
        DONE+END;
        DO I=1 TO END;
           J+1;
           NEWVAR=COMPRESS(VAR!!PUT(J,2.));
           SUBSTR(_TX,1,8)=NEWVAR;
           OUTPUT;
        END;
     END;
     J=0;DONE=0;
  END;
  ELSE IF _TY=99 THEN DO;OUTPUT;STOP;END;
RUN;
***; %MEND MODSCR;
```