

In producing reports with SAS*, sometimes the Print procedure can't do the job. In those instances, I have had to resort to the use of "PUT" statements to create the desired effect. Calculating and coding byte positions for the headings and variables is tedious, and must be redone each time the users decide that variables need to be added, deleted, or moved. To overcome this problem, I developed two macros to handle column headings. The macros "HDRIN" and "HDROUT" (LISTING 1) eliminate the problem of having to code "PUT" statements for column and, optionally, page headings. This method requires that the column and/or page heading layout be maintained in a dataset.

Another problem with the Print procedure is in producing reports that contain only one or two variables. Utilizing only a few percent of the bytes available in a report line is a waste of paper or, if the report is being written to a dataset with RECFM=FB, a waste of disk space. What I wanted was a way to print multiple observations per line without having to code "PUT" statements. My solution was to create two macros - "BLDIT" (LISTING 2), which transforms a multi-observation dataset into a one observation arrayed dataset, and "FILEPG" (LISTING 3) which formats and prints the dataset. By using macro parameters, the layout of the report is easily controlled. A sample of output produced by this method is in LISTING 6. This output was sized so that each page would fill one TSO* screen - making it very easy to page through the report online.

The SAS program in LISTING 4 was used to produce the report in LISTING 6. This run was produced with the "MPRINT" option so that the macro generated SAS code would be listed. The "DATA" step beginning in Line 25 of LISTING 4 reads in the external file which is to be reported on. Note that this data step is responsible for creating a global macro variable "TOTOBS", which contains the number of observations in the dataset "TESTIT". This information is needed by the macro "BLDIT" in order to correctly size the arrayed version of the file. Also note that the page headings for the report are placed in macro variables "PHEAD1-PHEADn" where they can be accessed by "FILEPG". The file is sorted in Line 43 to place the observations in the correct order for the report.

The "HDRIN" macro is invoked in Line 43 to read each line of the column heading layout file "P.HDRS(MNDEN)" (LISTING 5) as a SAS variable and write that data to WORK.HDR. The third parm specifies a read length of 32 bytes. This is needed to identify the maximum length of the line being read. The "HDRIN" macro must be invoked before the "FILEPG" macro for two reasons. First, "FILEPG" incorporates the "HDROUT" macro code within itself and expects to find correctly formatted header data in the file WORK.HDR. Second, "HDRIN" places the number of lines in the column heading file into a macro variable "HDRIN_N" for use by "FILEPG" in calculating page size.

The "BLDIT" macro is called in Line 46. "BLDIT" reads the SAS dataset "TESTIT", and creates a one observation dataset named "FINAL" which will contain two variables from the original dataset - "NAME" and "MNDEN". Formatting information must be passed to "BLDIT" so that the arrays containing the variables will be built correctly. This format information indicates that "NAME" is a character variable 13 bytes long, and that "MNDEN" is 5.2 numeric.

Lastly, "FILEPG" is invoked in Line 49 to generate the report. In the example, the instructions to "FILEPG" are to read the one observation, arrayed dataset "FINAL" and write its contents to the file associated with the "OUTLIB" DDname. The report is to contain 3 observations per line with a report width of 72 bytes and 20 lines per page. The array variables to print are "NAME" and "MNDEN". To calculate the byte positions for the variables in report columns 2-N, "FILEPG" must be primed with valid byte positions for the variables in the first report column. In this case, "NAME" is to begin at byte position 1 and "MNDEN" at byte position 17.

In order to print the variables, SAS formats may be passed to "FILEPG". In the example, no format is needed for the "NAME" variable. This must be indicated by using a null format since the macro parameters are positional. A null value is indicated by the absence of data between any two commas. For "MNDEN", the format "5.2" is used.

The macro parameter data combined with the array size in the macro variable "TOTOBS", the page heading data in the macro variables PHEAD1-PHEAD5, the column heading data in WORK.HDR, and the number of lines in WORK.HDR in the macro variable "HDRIN_N" is all the information needed by "FILEPG" to produce a customized report. An examination of the "FILEPG" source code in LISTING 3, along with the macro generated SAS code in LISTING 4 should be sufficient to provide an understanding of how the macro works.

The advantages of this methodology are that program development and maintenance time is minimized, and paper and/or disk space is saved. The primary disadvantage is the increased CPU time that is required to process the job.

LISTING 1: SAS SOURCE FOR HDRIN AND HDROUT MACROS

```

%GLOBAL HDRIN_N ;
%MACRO HDRIN(HDRMBR,WRK_OUT,VLEN);
*****;
/* MACRO:.....: HDRIN                                     */
/* AUTHOR.....: JIM PLAXCO                               */
/* PURPOSE....: READS A HEADER FILE AND WRITES IT TO WORK.WRK_OUT. */
/* PARAMETERS: HDRMBR = RECFM=VB DATASET WITH HEADER DATA */
/*              WRK_OUT= SAS WORK FILE OF HEADER DATA TO BUILD */
/*              VLEN  = MAXIMUM NUMBER OF COLUMNS TO READ IN */
/* OUTPUTS   : HDRIN_N= MACRO VARIABLE WITH THE NUMBER OF LINES */
/*              IN THE COLUMN HEADING FILE(USED BY FILEPG); */
*****;
DATA &WRK_OUT. (DROP=LINES);
  INFILE "&HDRMBR" LENGTH=RECLEN MISSEVER END=EOFFLAG;
  RETAIN LINES 0;
  INPUT @1 HDRDATA $VARYING&VLEN.. RECLEN;
  LINES+1;
  IF EOFFLAG THEN CALL SYMPUT('HDRIN_N',LINES);
%MEND HDRIN;

%MACRO HDROUT(WRK_IN,VLEN);
*****;
/* MACRO:.....: HDROUT                                     */
/* AUTHOR.....: JIM PLAXCO                               */
/* PURPOSE....: MACRO TO GENERATE A REPORT HEADER */
/* PARAMETERS: WRK_IN=SAS WORK DATASET TO READ FROM */
/*              : VLEN = MAXIMUM NUMBER OF COLUMNS TO WRITE */
*****;
DO HDR_LINE=1 TO LASTLINE BY 1;
  SET &WRK_IN POINT=HDR_LINE NOBS=LASTLINE;
  PUT @1 HDRDATA $CHAR&VLEN..;
END;
%MEND HDROUT;

```

LISTING 2: SAS SOURCE FOR BLDIT MACRO

```

%MACRO BLDIT(DS_IN,DS_OUT,VR1,AT1,VR2,AT2,VR3,AT3,VR4,AT4);
*****;
/* MACRO:.....: BLDIT                                     */
/* FILE.....: JPLAXCO.P.SASMACS(BLDIT)                   */
/* AUTHOR.....: JIM PLAXCO                               */
/* PURPOSE....: MACRO TO BUILD A ONE RECORD FILE BY */
/*              WRITING 1 to 4 VARIABLES TO ARRAYS */
/* PARAMETERS: DS_IN = DATASET TO READ FROM */
/*              DS_OUT= DATASET TO CREATE */
/*              VR1-4 = 1-4 VARIABLES TO TURN INTO ARRAYS */
/*              ARRAY NAME IS AVR#, ELEMENT IS VR#1-N */
/*              AT1-4 = ATTRIBUTES FOR EACH OF THE ARRAYS */
/*              */
/* NOTE THAT MACRO NEEDS THE NUMBER OF OBSERVATIONS TO BE PLACED */
/* IN THE MACRO VARIABLE TOTOBBS */
*****;
DATA &DS_OUT;
  SET &DS_IN END=EOFFLAG;
  RETAIN NOX EOFFLAG 0;
  NOX + 1;
  %LET OBS=%CMPRES(&TOTOBBS);
  %DO MI=1 %TO 4;
    %IF &&VR&MI NE %THEN
      %DO; ARRAY A&&VR&MI.(&OBS.) &&AT&MI &&VR&MI..1-&&VR&MI.&&OBS ;
        RETAIN &&VR&MI..1-&&VR&MI.&&OBS ;
        A&&VR&MI.(NOX) = &&VR&MI;
      %END;
    %END;
  IF EOFFLAG THEN OUTPUT;
%MEND BLDIT;

```

LISTING 3: SAS SOURCE FOR FILEPG MACRO

```

%GLOBAL PHEAD1 PHEAD2 PHEAD3 PHEAD4 PHEAD5;
%MACRO FILEPG(FIN,FOUT,CLS,LS,PS,V1,C1,F1,V2,C2,F2,V3,C3,F3,
              V4,C4,F4);
%*****;
%* MACRO:.....: FILEPG                                     *;
%* FILE.....: JPLAXCO.P.SASMACS(FILEPG)                   *;
%* AUTHOR....: JIM PLAXCO                                   *;
%* PURPOSE...: MACRO TO WRITE A MULTICOLUMN REPORT FROM ARRAY *;
%*              INPUT TO A OUTPUT DATASET.                 *;
%* PARAMETERS: FIN = DATASET TO READ FROM                 *;
%*              FOUT= DATASET TO CREATE                   *;
%*              CLS  = NUMBER OF COLUMNS TO PRINT        *;
%*              LS   = NUMBER OF CHARACTERS PER LINE      *;
%*              PS   = NUMBER OF LINES PER PAGE          *;
%*              V1-4 = 1-4 VARIABLES TO WRITE TO THE FILE *;
%*              C1-4 = INITIAL COLUMN POSITIONS FOR 1ST COLUMN *;
%*              F1-4 = OUTPUT FORMATS FOR THE VARIABLES   *;
%*              *;
%* NOTE: A MACRO VARIABLE TOTOBX MUST CONTAIN THE NUMBER OF *;
%*       ARRAY ELEMENTS.                                     *;
%* NOTE: MUST HAVE PREVIOUSLY INVOKED THE MACRO HDRIN TO BUILD *;
%*       THE COLUMN HEADING FILE. HDRIN GENERATES A MACRO *;
%*       VARIABLE HDRIN_N WITH THE NUMBER OF LINES IN THE *;
%*       COLUMN HEADING DATASET. COLUMN HEADINGS ARE ASSUMED *;
%*       TO BE IN THE DATASET WORK.HDR                     *;
%* NOTE: TO PRODUCE REPORT HEADINGS, THE MACRO VARIABLES *;
%*       PHEAD1-5 ARE READ FOR HEADING DATA AND ARE THEN *;
%*       CENETERED ON THE PAGE.                             *;
%*****;
%LOCAL CBRK CEND C12 CNDX COLV ICNDX MI OBS P_CNT PLEN TESTCNT;
%LET P_CNT=0;
%DO MI=1 %TO 5;
  %IF "&&PHEAD&MI" NE "" %THEN %LET P_CNT=%EVAL(&P_CNT + 1);
%END;
%LET TESTCNT=%EVAL(&PS - &P_CNT - &HDRIN_N);
%IF &TESTCNT < 1 %THEN
%DO;
  %PUT *** FILEPG PARM ERROR: PAGE SIZE IS TOO SMALL. THE TOTAL LINE;
  %PUT ***                      COUNT FOR THE PAGE AND COLUMN HEADING;
  %PUT ***                      EXCEEDS YOUR LINES PER PAGE VALUE;
%END;
%ELSE
%DO;
  %LET CBRK=%EVAL(&LS/&CLS);
  %LET CEND=%EVAL(&CLS-1);
  DATA _NULL_;
    FILE &FOUT NOPRINT NOTITLES;
    SET &FIN END=EOFVAR;
  %LET OBS=%CMPRES(&TOTOBX);
  %LET COLS=%CMPRES(&CLS);
  %DO MI=1 %TO 4;
    %IF &&V&MI NE %THEN
      %DO; ARRAY A&&V&MI.(&OBS.) &&V&MI..1-&&V&MI.&&OBS ;
      %END;
  %END;
  %END;
  RETAIN EVAR EOFVAR 0;
  RETAIN IC1 0 ;
  %DO MI=1 %TO &CEND;
    %LET COLPT&MI =%EVAL(&MI*&CBRK);
    %LET ICNDX=%EVAL(&MI+1);
    RETAIN IC&ICNDX 0 ;
  %END;
  RETAIN EPP 0 ; /* ELEMENTS PER PAGE */
  RETAIN EPC 0 ; /* ELEMENTS PER COLUMN */
  RETAIN ENDC1 1 ; /* LAST ELEMENT OF COLUMN 1 */
  RETAIN PNDX 0 ; /* PAGE INDEX */
  EPP = INT(&CLS.* &TESTCNT.);
  EPC = &TESTCNT;

```

LISTING 3: SAS SOURCE FOR FILEPG MACRO CONTINUED

```

DO UNTIL(IC&COLS. > &OBS.);
  LINK HDR;
  IC1 = INT(1+(PNDX*EPP)); PNDX+1;
  ENDC = IC1 + EPC;
  %DO CI=2 %TO &COLS ;
  %LET CI2=%EVAL(&CI-1);
  %IF &CI = 2 %THEN
    %DO; IC2 = ENDC;
    %END;
  %ELSE
    %DO; IC&CI = EPC + IC&CI2. ;
    %END;
  %END;
  DO UNTIL( IC1 = ENDC );
  %DO CI=1 %TO %EVAL(&COLS -1) ;
  %LET CNDX=%EVAL((&CI-1)*&CBRK);
  IF IC&CI <= &TOTOBS THEN
    PUT
    %DO MI=1 %TO 4;
    %IF &&V&MI NE %THEN
      %DO; @&&C&MI.+&CNDX. A&&V&MI.(IC&CI.) &&F&MI
      %END;
    %END;
    @ ;
    %IF &CI NE &COLS %THEN
      %DO; PUT @&&COLPT&CI. '|' @ ;
      %END;
    %END;
  IF IC&COLS <= &TOTOBS THEN
    PUT
    %DO MI=1 %TO 4;
    %IF &&V&MI NE %THEN
      %DO; %LET CNDX=%EVAL((&COLS-1)*&CBRK);
      @&&C&MI.+&CNDX. A&&V&MI.(IC&COLS.) &&F&MI
      %END;
    %END;
    )
    ELSE PUT;
  %DO MI=1 %TO &COLS;
  IC&MI + 1;
  %END;
  END;
  END;
  RETURN;
  *****;
  HDR:
  %DO MI=1 %TO 5;
  %IF "&&PHEAD&MI" NE "" %THEN
    %DO; %LET PLEN=%LENGTH(&&PHEAD&MI);
    %LET COLV=%EVAL(1+(&LS-&PLEN)/2);
    PUT @&COLV "&&PHEAD&MI.";
    %END;
  %END;
  PUT ' ';
  DO HDR_LINE=1 TO LASTLINE BY 1;
  SET HDR POINT=HDR_LINE NOBS=LASTLINE;
  PUT
  %DO CI=1 %TO &COLS;
  %LET CNDX=%EVAL((&CI-1)*&CBRK);
  @1+&CNDX HDRDATA $CHAR&CBRK..
  %END;
  ;
  END;
  RETURN;
  %END;
  %MEND FILEPG;

```

LISTING 4: EDITED SASLOG FOR PROGRAM USING THE REPORT MACROS

```

23  FILENAME OUTLIB 'JPLAXCO.P.DATA(PLANRPT)' DISP=OLD; RUN;
24  ***** STEP 1 - READ IN DATA *****
25  DATA TESTIT (KEEP=NAME CLASS MNDEN);
26    INFILE 'JPLAXCO.P.DATA(PLAN)' END=EODDATA;
27    LENGTH NAME $ 11 CLASS $ 3; RETAIN OBSCNT 0;
28    INPUT @1 TCHAR $1 @;
29    IF TCHAR = '*' THEN DELETE;
30    ELSE DO; INPUT NAME $ CLASS $ RADIUS MNORB MNDEN ;
31      OBSCNT + 1; OUTPUT;
32    END;
33  IF EODDATA THEN
34    DO; /*TOTOBS IS USED BY BLDIT AND PHEAD1-3 IS USED BY FILEPG*/
35      CALL SYMPUT('TOTOBS',OBSCNT);
36      CALL SYMPUT('PHEAD1','RANK OF PLANETS/MOONS BY DENSITY');
37      PTEMP='TOTAL ENTRIES= '|| LEFT(OBSCNT);
38      CALL SYMPUT('PHEAD2',PTEMP);
39      CALL SYMPUT('PHEAD3','UNKNOWN DENSITIES ARE SET TO ZERO');
40    END;
41  PROC SORT DATA=TESTIT; BY DESCENDING MNDEN;
42  RUN;
43  %HDRIN(JPLAXCO.P.HDRS(MNDEN),HDR,32)
(HDRIN): DATA HDR (DROP=LINES);
(HDRIN): INFILE "JPLAXCO.P.HDRS(MNDEN)" LENGTH=RECLEN
(HDRIN): MISSEVER END=EOFFLAG;
(HDRIN): RETAIN LINES 0;
(HDRIN): INPUT @1 HDRDATA $VARYING32. RECLEN;
(HDRIN): LINES+1;
(HDRIN): IF EOFFLAG THEN CALL SYMPUT('HDRIN_N',LINES);
44  RUN;
45  ***** STEP 2 - RUN THE BUILD MACRO *****
46  %BLDIT(TESTIT,FINAL,NAME,$13,MNDEN,5.2)
(BLDIT): DATA FINAL;
(BLDIT): SET TESTIT END=EODFLAG;
(BLDIT): RETAIN NDX EODFLAG 0;
(BLDIT): NDX + 1;
(BLDIT): ARRAY ANAME(70) $13 NAME1-NAME70 ;
(BLDIT): RETAIN NAME1-NAME70 ;
(BLDIT): ARRAY AMNDEN(70) 5.2 MNDEN1-MNDEN70 ;
(BLDIT): RETAIN MNDEN1-MNDEN70 ;
(BLDIT): ANAME(NDX) = NAME;
(BLDIT): AMNDEN(NDX) = MNDEN;
(BLDIT): IF EODFLAG THEN OUTPUT;
47  RUN;
48  ***** STEP 3 - SIMULATE THE PRINT COLS BY PAGE MACRO *****
49  %FILEPG(FINAL,OUTLIB,3,72,20,NAME,1,,MNDEN,17,5.2)
(FILEPG) DATA _NULL_;
(FILEPG) FILE OUTLIB NOPRINT NOTITLES;
(FILEPG) SET FINAL END=EOFVAR;
(FILEPG) ARRAY ANAME(70) NAME1-NAME70 ;
(FILEPG) ARRAY AMNDEN(70) MNDEN1-MNDEN70 ;
(FILEPG) RETAIN EVAR EOFVAR 0;
(FILEPG) RETAIN IC1 0 ;
(FILEPG) RETAIN IC2 0 ;
(FILEPG) RETAIN IC3 0 ;
(FILEPG) RETAIN EPP 0 ;
(FILEPG) RETAIN EPC 0 ;
(FILEPG) RETAIN ENDC1 1 ;
(FILEPG) RETAIN PNDX 0 ;
(FILEPG) EPP = INT(3* 14);
(FILEPG) EPC = 14;
(FILEPG) DO UNTIL(IC3 > 70);
(FILEPG) LINK HDR;
(FILEPG) IC1 = INT(1+(PNDX*EPP));
(FILEPG) PNDX+1;
(FILEPG) ENDC = IC1 + EPC;
(FILEPG) IC2 = ENDC;
(FILEPG) IC3 = EPC + IC2 ;
(FILEPG) DO UNTIL( IC1 = ENDC );
(FILEPG) IF IC1<=70 THEN PUT @1+0 ANAME(IC1) @17+0 AMNDEN(IC1) 5.2 @;
(FILEPG) PUT @24 '|' @ ;

```

LISTING 4: EDITED SASLOG FOR PROGRAM USING THE REPORT MACROS CONTINUED

```
(FILEPG) IF IC2<=70 THEN
(FILEPG) PUT @1+24 ANAME(IC2) @17+24 AMNDEN(IC2) 5.2 @;
(FILEPG) PUT @48 ' ' @;
(FILEPG) IF IC3<=70 THEN PUT @1+48 ANAME(IC3) @17+48 AMNDEN(IC3) 5.2;
(FILEPG) ELSE PUT;
(FILEPG) IC1 + 1; IC2 + 1; IC3 + 1;
(FILEPG) END; END; RETURN;
(FILEPG) HDR: PUT @20 "RANK OF PLANETS/MOONS BY DENSITY";
(FILEPG) PUT @28 "TOTAL ENTRIES= 70 ";
(FILEPG) PUT @20 "UNKNOWN DENSITIES ARE SET TO ZERO";
(FILEPG) PUT ' ';
(FILEPG) DO HDR_LINE=1 TO LASTLINE BY 1;
(FILEPG) SET HDR_POINT=HDR_LINE NOBS=LASTLINE;
(FILEPG) PUT @1+0 HDRDATA $CHAR24. @1+24 HDRDATA $CHAR24.
(FILEPG) @1+48 HDRDATA $CHAR24. ;
(FILEPG) END; RETURN;
50 RUN;
```

LISTING 5: DATA IN HEADER FILE

```
OBJECT      MEAN
NAME        DENSITY
=====
```

LISTING 6: SAMPLE REPORT OUTPUT

RANK OF PLANETS/MOONS BY DENSITY					
TOTAL ENTRIES= 70					
UNKNOWN DENSITIES ARE SET TO ZERO					
OBJECT NAME	MEAN DENSITY	OBJECT NAME	MEAN DENSITY	OBJECT NAME	MEAN DENSITY
=====					
EARTH	5.52	DEIMOS	1.70	MIMAS	1.17
MERCURY	5.43	TITANIA	1.68	WATER/REF.	1.00
VENUS	5.25	ARIEL	1.66	SATURN	0.69
MARS	3.95	NEPTUNE	1.64	METIS	0.00
IO	3.57	OBERON	1.58	ADRASTEIA	0.00
MOON	3.34	UMBRIEL	1.51	AMALTHEA	0.00
EUROPA	2.97	DIONE	1.44	THEBE	0.00
TRITON	2.07	MIRANDA	1.35	LEDA	0.00
PLUTO	2.03	JUPITER	1.33	HIMALIA	0.00
PHOBOS	2.00	RHEA	1.33	LYSITHEA	0.00
CHARON	2.00	URANUS	1.29	ELARA	0.00
GANYMEDE	1.94	TETHYS	1.26	ANANKE	0.00
TITAN	1.88	ENCELADUS	1.24	CARME	0.00
CALLISTO	1.86	IAPETUS	1.21	PASIPHAE	0.00
RANK OF PLANETS/MOONS BY DENSITY					
TOTAL ENTRIES= 70					
UNKNOWN DENSITIES ARE SET TO ZERO					
OBJECT NAME	MEAN DENSITY	OBJECT NAME	MEAN DENSITY	OBJECT NAME	MEAN DENSITY
=====					
SINOPE	0.00	CRESSIDA	0.00		
ATLAS	0.00	DESDEMONA	0.00		
PROMETHEUS	0.00	JULIET	0.00		
PANDORA	0.00	PORTIA	0.00		
EPIMETHEUS	0.00	ROSALIND	0.00		
JANUS	0.00	BELINDA	0.00		
TELESTO	0.00	PUCK	0.00		
CALYPSO	0.00	1989N6	0.00		
HELENE	0.00	1989N5	0.00		
HYPERION	0.00	1989N3	0.00		
PHOEBE	0.00	1989N4	0.00		
CORDELIA	0.00	1989N2	0.00		
OPHELIA	0.00	1989N1	0.00		
BIANCA	0.00	NEREID	0.00		

* SAS is a registered trademark of SAS Institute, Inc., Cary, NC, USA.
* ISPF is a registered trademark of IBM Corp., Armonk, NY, USA.