How To Sort Production Reports Prior to Printing

Helen-Jean Talbott, Commercial Credit Corporation, Baltimore, MD
Earl R. Westerlund, Eastman Kodak Company, Rochester, NY

ABSTRACT

Each month the Credit Policy Department of Commercial Credit produces about 25 different reports to send to the 202 district, 48 region, and 8 division managers. These reports are produced using base SAS® software in programs run in a production environment under MVS. Each district manager receives portions of the reports specific for that district. Similarly, region and division managers receive portions specific for the districts within regions and for regions within division. Due to the size of these reports and number of recipients, it would take two days for a person to sort and label them for distribution. There was a need for an automatic method to print reports in order by district so that they would not have to be sorted manually. The purpose of this poster is to show how to create SAS data sets and report modules and to use them in conjunction with the SAS macro language to produce reports sorted in district order prior to printing.

INTRODUCTION

The Credit Policy Department of Commercial Credit Corporation in Baltimore produces more than 200 large reports using base SAS software in a production environment under MVS in the first three days of each month. These reports cover many different topics related to credit policy in the 1100 Commercial Credit offices located across the United States such as summary tables showing approval (applications taken vs. loans made) and delinquency (late payment) rates and account lists showing recently made loans that did not meet company guidelines. Of these 200 reports, about 25 different reports are sent to the 202 district, 48 region, and 8 division managers. Each district manager receives portions of the reports specific for that district. Similarly, region and division managers receive portions specific for the districts within regions and for regions within division. Reports are sorted and paged by district number which is 6 characters long. The first two digits of the district number indicate the division, the first four digits indicate the region. Thus, three copies of each report are printed, with the first copy divided into districts, second copy divided into regions, and the third copy divided into divisions. This process is repeated for each report, until a complete package containing all reports is assembled for each district, region, and division. Due to the size of these reports and number of recipients, it would take two days for a person to sort and label them for distribution.

There was a need for an automatic method to print reports in order by district so that they would not have to be sorted manually. The idea was simple, but the solution was not obvious. After several unsuccessful attempts, Helen-Jean Talbott posted a question on SAS-L, an Internet listserv for SAS software users, and received several suggestions. Of these suggestions, the solution proposed by Earl Westerlund was the most successful and has been implemented in the Credit Policy MIS production system. The purpose of this paper is to show how to create SAS data sets and report modules and to use them in conjunction with the SAS macro language to produce reports sorted in district order prior to printing.

GENERAL PROCESS

For many simple SAS programs, it is common to read data from an input file, select and manipulate specific observations, and create new variables in a temporary data set that then is used in a reporting procedure (such as PROC PRINT) to produce the desired list. The approach used in our solution is quite similar.

The first program extracts the desired observations, creates new variables, and produces a permanent output data set sorted by district. Due the size of the data sets and the complexity of the data manipulations, a separate program is run for each report desired in the distribution package.

The code normally present in the PROC PRINT step (such as the VAR, LABEL, FORMAT, and TITLE statements) is stored in a separate file, or module, with a separate module for each report desired. These report modules are used by more than one program in the production schedule. Maintenance and consistency are improved by storing the code in modules.

When all of the permanent output data sets are ready, the sorting program runs. This program determines the identification numbers for the active districts and places them into macro variables such as DIST1, DIST2, etc. The next portion of the program operates within a macro which executes once for each active district. Within this macro, a PROC PRINT step (with a WHERE statement that selects data for that district) executes on the permanent data set created by the first program and uses a %INCLUDE statement to bring in the corresponding reporting module. There is a set of PROC PRINT, WHERE, and %INCLUDE statements for each report in the distribution package. When a district has no observations for a particular report, that report does not print (which was the desired result).

EXAMPLE AND PROGRAM CODE

This example deals with the fourth report in the distribution package. It is a list of personal loan and real estate accounts that were either refinanced (without additional cash) in the current month or were refinance within the last six months and are currently delinquent. This list is useful for identifying customers that are having difficulty making payments on their loans. Below is the source code for the first program for this report.

```
*******************************************************;
* COP.PROD.SOURCE(BCOPREFN)                           *;
*******************************************************;
* LIBRARY DD DSN=D13BP.MONTHEND.PERM.FMTLIB(+0)       *;
* IN1     DD DSN=D13BP.MONTHEND.INDV.DATA(+0)         *;
* OUT1    DD DSN=D13BP.REPORT.H117                    *;
*                                                     *;
*                                DECEMBER 14, 1996    *;
*                                HELEN-JEAN TALBOTT   *;
*                                                     *;
* NEW PROGRAM FOR JUL91 MONTHEND                      *;
*                                                     *;
* PRODUCE OUTPUT DATA SET FOR                         *;
*    CREDIT POLICY MIS REPORT H117                    *;
*                                                     *;
* PRODUCES AN ACCOUNT LIST OF ACCOUNTS                *;
* REFINANCED <= 6 MONTHS AND ARE CURRENTLY DLQ AND    *;
* ACCOUNTS REFINANCED IN THE CURRENT MONTH            *;
* PERSONAL LOAN AND REAL ESTATE                        *;
*                                                      *;
*******************************************************;
OPTIONS LS=177;

DATA ACCOUNTS(KEEP=OFFICE CNAME ACCTN SOB PRODUCT
                   SECURITY EXPOSURE NOTE_DY RSCHEDDY
                   TERM NOTE_AM AD_TRUE BALANCE
                   NOUT EVER DLQ_IND NAME DIST
                   DISTNAME);
    SET IN1.A(WHERE=(REFIN=1));

    *** EXCLUDE RB10 ACCOUNTS WITH $200+ NEW CASH ***;
    NEWCASH=SUM(AD_CASH,(FEE_TOT*-1),(RENEWAMT*-1));
    IF NEWCASH<0 THEN NEWCASH=0;
    IF SOB='S' AND NEWCASH GE 200 THEN DELETE;

    *** DEFINE DLQ INDICATOR ***;
    IF DLQ90 GE 1 THEN EVER = '90+';
    ELSE IF DLQ60 GE 1 THEN EVER = '60 ';
    ELSE IF DLQ30 GE 1 THEN EVER = '30 ';

    *** CURRENT BANKRUPTCY STATUS ***;
    IF BANK7='1' OR BANK13='1' THEN DLQ_IND = -1;

    IF INTEREST='1' THEN NOTE_AM = AD_TRUE;

    LENGTH NAME     $27
           DIST     $6
           DISTNAME $27;

    IF PUT(OFFICE,$BO2OPEN.)='OPEN' THEN DO;
       NAME=PUT(OFFICE,$BO2BRNM.);
       DIST=PUT(OFFICE,$BO2DST.);
       DISTNAME=PUT(OFFICE,$BO2DSTN.);
       END;
    ELSE DO;
       NAME=OFFICE;
       DIST='000000';
       DISTNAME='CLOSED BRANCHES';
       END;

PROC SORT DATA=ACCOUNTS OUT=OUT1.A;
    BY DISTNAME NAME;

RUN;
```

The input file, D13BP.MONTHEND.INDV.DATA, contains accounts that will appear on several exception reports. When this file is created in an earlier production program, the flag REFIN is set to 1 for those accounts which should be included on this refinance exception report.

There is a further exclusion to remove refinance accounts where there is more than $200 additional cash in the new loan (the reason for the loan was not due to the customer having financial difficulty). The new variable EVER shows the worst delinquency for the life of the loan. Variable NAME, DIST, and DISTNAME are created from the office number using permanent custom formats $BO2BRNM, $BO2DST, $BO2DSTN which provide the translation from the office number to the office name, district number, and district name, respectively. The custom format $BO2OPEN is used to determine whether the office number is for an open or closed office. This is important because the $BO2BRNM, $BO2DST, and $BO2DSTN formats do not contain the translations for closed offices.

The output file, D13BP.REPORT.H117, is the file that will be used in the sorting program to produce Credit Policy MIS Report H117, which is the refinance exception report. This output data set is sorted by DISTNAME (district number + name) and NAME (office name) so that it is in the proper order for the sorting program. The contents of the output data set are shown below. All of the variables listed on the report are present in the data set. The variable DIST is required later for the sorting program.

```
PHYSICAL NAME:            D13BP.REPORT.H117

--ALPHABETIC LIST OF VARIABLES AND ATTRIBUTES--

VARIABLE    TYPE    LEN    LABEL
-----------------------------------------------
ACCTN       NUM       8    ACCOUNT NUMBER
AD_TRUE     NUM       5    ADVANCE
BALANCE     NUM       5    GROSS BALANCE
CNAME       CHAR     30    CUSTOMER NAME
DIST        CHAR      6    DISTRICT, 6-DIGIT
DISTNAME    CHAR     27    DISTRICT NO. + NAME
DLQ_IND     NUM       2    DLQ INDICATOR
EVER        CHAR      3    WORST DLQ OVER LIFE
EXPOSURE    NUM       8    MONTHS OPEN
NAME        CHAR     27    OFFICE NAME
NOTE_AM     NUM       5    NOTE AMOUNT
NOTE_DY     NUM       8    NOTE DATE
NOUT        NUM       8    NET BALANCE
OFFICE      CHAR      6    OFFICE, 6-DIGIT CODE
PRODUCT     CHAR      2    PRODUCT TYPE
RSCHEDDY    NUM       8    RESCHEDULED DATE
SECURITY    CHAR      1    SECURITY CODE
SOB         CHAR      1    SOURCE OF BUSINESS
TERM        NUM       8    TERM

       -----SORT INFORMATION-----

       SORTEDBY:      DISTNAME NAME
       VALIDATED:     YES
       CHARACTER SET: EBCDIC
```

The report module for the refinance exception report is stored in the file COP.PROD.SOURCE(RPTH117). The code for this module is shown below. Titles and footnote have been split across lines to fit in the space available in this paper. In the actual file, the titles normally fill 72 characters before continuing on the next line. The S=72 option is used with the %INCLUDE in the sorting program to accommodate long report titles.

```
*******************************************************;
* COP.PROD.SOURCE(RPTH117)                             *;
*******************************************************;
*                                                      *;
*                              OCTOBER 31, 1994        *;
*                              HELEN-JEAN TALBOTT      *;
* NEW MODULE FOR OCT94 MONTHEND                        *;
* DERIVED FROM BCOPREFN                                *;
*                                                      *;
* REPORT MODULE FOR BCOPREFN & BCOPDRPT.               *;
*                                                      *;
* --- MODIFICATION LIST ---                            *;
* 11/01/94 HJT - PROGRAM HEADER: ADDED NOTES ABOUT     *;
*                START DATE                            *;
*              - REVISED TITLE 2                       *;
*                                                      *;
*******************************************************;
   BY DISTNAME NAME;
   PAGEBY DISTNAME;
   VAR CNAME ACCTN SOB PRODUCT SECURITY EXPOSURE
       NOTE_DY RSCHEDDY
       TERM NOTE_AM AD_TRUE BALANCE NOUT EVER DLQ_IND;
   LABEL
       DISTNAME ='DISTRICT'
       NAME     =' --- OFFICE'
       CNAME    = 'CUSTOMER*NAME'
       ACCTN    = 'ACCOUNT*NUMBER'
       SOB      = 'SOURCE'
       PRODUCT  = 'PRODUCT'
       SECURITY = 'SECURITY'
       EXPOSURE = 'MONTHS*OPEN'
       NOTE_DY  = 'NOTE*DATE'
       RSCHEDDY = 'RESCHED*DATE'
       NOTE_AM  = 'NOTE*AMOUNT'
       TERM     = 'TERM'
       AD_TRUE  = 'ADVANCE'
       BALANCE  = 'GROSS*BALANCE'
       NOUT     = 'NET*BALANCE'
       EVER     = 'WORST*EVER*DLQ'
       DLQ_IND  = 'CURRENT*DLQ?';
   FORMAT NOTE_AM COMMA7.0
          DLQ_IND DQ.
          CNAME $22.
          SOB $SOA.
          SECURITY $SEC.
          BALANCE COMMA7.0
          AD_TRUE NOUT COMMA7.0
          NOTE_DY MMDDYY8.
          PRODUCT $PROD.;
   TITLE1 ' ';
   TITLE2 'ACCOUNTS REFINANCED WHICH ARE CURRENTLY DLQ
AND ACCOUNTS REFINANCED IN THE CURRENT MONTH';
   TITLE3 '-----------------------------------------';
   TITLE4 "PERSONAL LOAN AND REAL ESTATE -- &DTWORD";
   FOOTNOTE1 'REFINANCED ACCOUNTS INCLUDE: RBO,
ADJUSTMENT OF TERM, RESCHEDULE, PL RB10 WITH LESS THAN
$200 NEW CASH, RE RB10 WITH LESS THAN $1500 NEW CASH';
   FOOTNOTE3 'CREDIT POLICY MIS REPORT H117 - BCOPDRPT
& BCOPREFN';
 RUN;
    TITLE;
    FOOTNOTE;
 RUN;
```

The above code is fairly typical for a PROC PRINT step. The formats DQ, $SOA, $SEC, $PROD are stored in the permanent custom format library. The PROC PRINT statement is not in this file, but is present in the sorting program which uses this module.

The code for the sorting program is located in the file COP.PROD.SOURCE(BCOPDRPT). Both the JCL and the SAS code for this program are shown below.

```
//CMISDRPT JOB 756AS2,'H TALBOTT',MSGCLASS=9,CLASS=1
/*ROUTE PRINT R1
/*JOBPARM R=07D
/*JOBPARM L=9999
// EXEC SAS,TIME=(90),
//         OPTIONS='MACRO S=72 CHARCODE DQUOTE NOSTATS'
//WORK    DD UNIT=SYSDA,
//           SPACE=(CYL,(450,100)),BLKSIZE=27648
//SASLOG  DD SYSOUT=T
//SASLIST DD SYSOUT=(T,,LUS3),COPIES=3
//LIBRARY DD DSN=D13BP.MONTHEND.PERM.FMTLIB(+0),
//           DISP=SHR
//RPT4    DD DSN=COP.PROD.SOURCE(RPTH117),
//           DISP=SHR
//RPT5    DD DSN=COP.PROD.SOURCE(RPTH209),
//           DISP=SHR
//RPT6    DD DSN=COP.PROD.SOURCE(RPTH224),
//           DISP=SHR
```

{more code here to specify additional report modules}

```
//IN0     DD DSN=D13BP.EDR.BRANCH.DATA(+0),
//           DISP=SHR
//IN4     DD DSN=D13BP.REPORT.H117,
//           DISP=SHR
//IN5     DD DSN=D13BP.REPORT.H209,
//           DISP=SHR
//IN6     DD DSN=D13BP.REPORT.H224,
//           DISP=SHR
```

{more code here to specify additional input data sets}

```
//SYSIN   DD *
*******************************************************;
* COP.PROD.SOURCE(BCOPDRPT)                           *;
*******************************************************;
* LIBRARY  DD DSN=D13BP.MONTHEND.PERM.FMTLIB(+0)      *;
* RPT4     DD DSN=COP.PROD.SOURCE(RPTH117)            *;
* RPT5     DD DSN=COP.PROD.SOURCE(RPTH209)            *;
* RPT6     DD DSN=COP.PROD.SOURCE(RPTH224)            *;
```

{more code here for additional report modules}

```
* IN0      DD DSN=D13BP.EDR.BRANCH.DATA(+0)           *;
* IN4      DD DSN=D13BP.REPORT.H117                   *;
* IN5      DD DSN=D13BP.REPORT.H209                   *;
* IN6      DD DSN=D13BP.REPORT.H224                   *;
```

{more code here for additional input data sets}

```
*                                                     *;
*                                 OCT 28, 1997        *;
*                                 HELEN-JEAN TALBOTT  *;
*                                                     *;
* NEW PROGRAM FOR OCT94 MONTHEND                      *;
* (DEVELOPED UNDER SAS RELEASE 6.08)                  *;
*                                                     *;
* PROGRAM PRODUCES A COMBINED REPORT OF ALL REPORTS   *;
* PROCESSED AT MONTHEND BY CREDIT POLICY FOR          *;
* DISTRIBUTION TO DISTRICTS AND ABOVE                 *;
*                                                     *;
*******************************************************;
OPTIONS LS=177;

*****************************************;
***   DEFINE GENERIC MACRO VARIABLES   ***;
***      FOR MONTH-END PRODUCTION      ***;
*****************************************;

DATA _NULL_;
   TODAY=DATE();

   *** MONTH AND YEAR OF "CURRENT MONTH" ***;
   MO=MONTH(TODAY)-1;
   IF MONTH(TODAY)=1 THEN MO=12;
   YR=YEAR(TODAY);
   IF MONTH(TODAY)=1 THEN YR=YEAR(TODAY)-1;

   *** FIRST DAY OF "CURRENT MONTH" ***;
   FIRSTDAY=MDY(MO, 1, YR);

   *** TITLE DATES IN THE FORM "MONTH YYYY" ***;
DTWORD=UPCASE(TRIM(PUT(FIRSTDAY,WORDDATE9.))) ?/?/ ' '
?/?/ PUT(YR,4.);

CALL SYMPUT('DTWORD',DTWORD);

RUN;
******* END GENERIC MACRO VARIABLE DEFINITION *******;

        %LET DTWORD=%TRIM(&DTWORD);

***    CREATE DISTRICT DATA SET      ***;
*** CONTAINING ONLY ACTIVE DISTRICTS ***;
PROC SORT DATA=IN0.A(KEEP=DIST) OUT=DISTLIST NODUPKEY;
   BY DIST;
   WHERE DIST NE ' ';

RUN;

*===================================================*;
* MACRO WHICH WILL CONTAIN DISTRICT VALUES AND WILL  *;
* USED IN LOOPING PROCESS TO CREATE SINGLE LARGE     *;
* REPORT                                             *;
* -------------------------------------------------- *;
* CODE PROVIDED BY EARL R. WESTERLUND THROUGH SAS-L  *;
* ON INTERNET)                                       *;
*===================================================*;

*** PLACE DISTRICT VALUES INTO MACRO VARIABLES ***;
*** - AND COUNT THEM                           ***;

DATA _NULL_;
   SET DISTLIST END=LAST;
   BY DIST;

   IF FIRST.DIST THEN DO;
      I + 1;
      CALL SYMPUT('DSTN' ?/?/ LEFT(PUT(I,3.)),
                   DIST);
      END;
   IF LAST THEN CALL SYMPUT('NDST', PUT(I, 3.) );
RUN;


***** PRINT REPORTS BY DISTRICT *****;
%MACRO BYDIST;
    %DO I = 1 %TO &NDST;
```

```
        PROC PRINT DATA=IN4.A UNIFORM NOOBS SPLIT='*';
        WHERE DIST="&&DSTN&I";
        *** INCLUDE PRINT MODULE ***;
        %INCLUDE RPT4/SOURCE2 S2=72;

        PROC PRINT DATA=IN5.A UNIFORM NOOBS SPLIT='+' N;
        WHERE DIST="&&DSTN&I";
        *** INCLUDE PRINT MODULE ***;
        %INCLUDE RPT5/SOURCE2 S2=72;

        PROC PRINT DATA=IN6.A UNIFORM NOOBS SPLIT='*';
        WHERE DIST="&&DSTN&I";
        *** INCLUDE PRINT MODULE ***;
        %INCLUDE RPT6/SOURCE2 S2=72;

                    {more code here for remaining
                     report modules and input data sets}

        %END;
     %MEND;

%BYDIST;

RUN;
```

The above code shows the JCL and SAS code for three of the 25 reports. This example continues with how report H117 is produced.

The JCL specifies DD statements for the permanent custom format library (D13BP.MONTHEND.PERM.FMTLIB(+0)), which is followed by the DD statements for the report modules (COP.PROD.SOURCE(RPTH117) for report H117), which are then followed by the DD statements for the input data sets. The first input data set, D13BP.EDR.BRANCH.DATA(+0) contains information on all the open branches and their assignment to districts. This data set is used to determine the active districts later in the program. The statement /*JOBPARM L=9999 is present in this program to override the normal limit on the number of pages that can be printed. Three copies of the entire output will be printed.

After the header block of the SAS code, is a section called “GENERIC MACRO VARIABLES” used to determine the number and name of the current month. Our production programs run at the beginning of the month and use data for the month that has just completed. Therefore, the “current month” is really the calendar month prior to the month in which the programs are run, so one month is subtracted from the date. The date is determined from the computer system clock. By using the automatic SAS macro variable DATE(), along with several useful date functions such as MONTH, YEAR, and MDY, this code runs smoothly from month to month without requiring manual monthly changes to date-related code.

The next section creates a temporary data set called DISTLIST which is ordered by district number (DIST) and contains only active districts. Then the observations in DISTLIST are read in the DATA _NULL_ step in order to put the values of the district numbers into macro variables. The first district number is 010101 and it is stored in macro variable DIST1. The second district number is 010102 and it is stored in macro variable DIST2. This step also counts the number of districts and places that number into the macro variable NDST so that the program knows later when the last district is processed.

Now it is time to start printing the reports. The macro BYDIST is executed once for each district through the statement
%DO I = 1 %TO &NDST;

In this example the first step encountered within the macro is
PROC PRINT DATA=IN4.A UNIFORM NOOBS SPLIT=‘*’;

This executes a PROC PRINT on the referred top by the JCL DD statement IN4 which is the data set for report H117, the refinance exception report.

The first time through the macro, the next statement
WHERE DIST=“&&DSTN&I”;
first resolves to WHERE DIST=“&DSTN1”;
which then resolves to WHERE DIST=“010101”
so data from the first district (010101) are selected from the input data set (IN4.A) for processing.

The %INCLUDE statement executes the statements stored in the file referenced by JCL DD statement RPT4 which contains the report module for report H117. The options SOURCE2 causes the SAS log to show the source statements that are being included in the SAS program. Remember that this prints each for each district for each report, so consider removing this option if a complete log is not required. The S=72 option specifies that a record length of 72 should be used for input during the %INCLUDE. For long titles, the characters fill up 72 characters and are continued immediately on the next line of the program. When the title prints, the title appears normally without any break.

After the PROC PRINT is completed for this input data set and report module, the next PROC PRINT is encountered. It executes, on the same district (010101) and uses the next input data set and the next report module. This process continues until all the reports for a particular district are printed. Then the program returns to the top of the module and repeats the process with the next district, until all of the districts have been processed. The end result is a very large stack of paper containing three copies of output with all the reports pertaining to each district printed together from the first district (on top) to the last district (on the bottom), plus one ecstatic person that no longer has to sort the reports manually and a multitude of happy district, region, and division managers that now receive their reports earlier in the month.

CONCLUSIONS

This paper demonstrates how to write a SAS code to create SAS data sets and report modules and to use them in conjunction with the SAS macro language to produce reports sorted in district order prior to printing. The example tracks the production of report H117, the refinance exception report produced by the Credit Policy department of Commercial Credit as a part of monthly production under MVS. The approach is fairly easy to implement and has saved at least two person-days per month at Commercial Credit since its development.

REFERENCES

SAS Institute Inc. (1990), SAS Language, Cary, NC: SAS Institute Inc., 1042 pp.

SAS Institute Inc. (1987), SAS Guide to Macro Processing, Cary, NC: SAS Institute Inc., 233pp.

ACKNOWLEDGMENTS

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

Helen-Jean Talbott
Commercial Credit Corporation
300 St. Paul Place, BSP07D
Baltimore, MD 21202

email: HJTalbott@Worldnet.Att.Net

Earl R. Westerlund
Eastman Kodak Company
1669 Lake Ave, MC 24608
Rockester, NY 14652-4608

email: EarlW@Kodak.Com