

The Development of a Cost Recovery / Chargeback System for Printing

Vickie L. Custer

University of Wisconsin - Milwaukee
Information Systems - Administrative Data Processing

When the University of Wisconsin - Milwaukee (UW-Milwaukee) needed to replace impact printers with a cut sheet laser printer, the Administrative Computing Services Department was required to provide a method for recovering costs of overusage of the printer to Administrative Affairs division. One of the requirements when the laser printer was obtained, was that printing would not exceed an average number of sheets printed based on the prior three years of printing on impact printers. This average was also used to determine the maintenance agreement for the laser printer. The agreement stated that if printing increased by more than 10% of the average based on three years of impact printing, the university would pay a usage fee for the excess printed. As a result of this agreement, Administrative Affairs gave approval for obtaining the laser printer and for the development of a cost recovery / charge back system for all printing done by Information Systems.

The first step in the development of the system was to determine how to charge back the excess usage. This required a thorough evaluation of the type of printing that Information Systems was providing to the user community. The printing being done included user supplied forms, cheshire labels, pressure sensitive labels, I.S. supplied forms and standard printing. Excluded from the evaluation was any type of printing being done in remote locations. Only the print being done on the mainframe printers would be evaluated. Once the type of printing was determined, the next step was to collect all historical data by department / division to determine allocation amounts. At this point, SAS* became the tool of choice. The historical data was system accounting data for the prior three years. H.W. Merrill's software package, MXG*, was used to decode the specific system accounting records. Once the data was decoded, the records were summarized by account, form, number of lines printed and number of pages printed for 1987-88, 1988-89, and 1989-90 fiscal years. The summarized yearly data was then averaged for the three years to determine the allocation for the department or division.

After the allocations were determined, the design phase began. The initial parameters for the system were that each department or division would receive a yearly allocation of dollars to spend on printing. The allocation would be decremented by the usage on a monthly cycle. A summary statement of usage and charges and a

The inclusion of inventory did not add any new fields to the database. It was agreed that only a summary by form of the pages printed and estimated pages printed would be sufficient for the Operations Manager. Later, stock available and remaining would be added. The addition of this information became important to several of the departments, since they supply their own forms and needed a monthly count to know when to reorder.

Now the fun began. As the design of each phase was completed, the programming began. The daily data had to be summarized into month-to-date data. The data that came into the system was system data that needed to be decoded. The MXG software decoded the daily data into meaningful variables. The data was then summarized by job number and account. From the daily data a merge was done by account number to get the department information added to the daily data. If the account was not found, an error file was created that would be used as input to a correction run. After the data was merged with the account file, the month to date files are created. First, the current month-to-date file is renamed to a work month-to-date. The summarized data for the day is merged with the work month-to-date creating a new current month-to-date file. The old month-to-date file is deleted and the work month-to-date file is appended to the old month-to-date file. There are now three files existing for the month, one is an old month-to-date, one is the current month-to-date and the third is the account error file. Examples of the code used for creating the three files can be seen in figure 1.

When the system was developed, several macros to handle current and previous months for data files were created. Those macros are listed in figure 2. The rest of the macros used are from Merrill's MXG software.

Two reports are generated from the daily runs. One is the report of the daily data with invalid account numbers and the second is the summary of the daily data. The report of the daily data in error gives the job name and time of run, so that the error can be tracked down. In most cases it is a TSO user that has entered the account number incorrectly. The person responsible for this job will go in and change the account number through an FSP screen. If a new account must be added to the system, maintenance to the account file is performed through an FSP screen also. Once the error has been corrected, there is a correction run that is done to add the records to the current month-to-date file. The correction run merges the account file information with the error information and then appends the corrected data to the current month-to-date. The second report, the summary of the daily data just lists the jobs, account numbers, department information and the pages and lines printed. This allows for the billing person to verify that all data has been added to the month-to-date file. It is also used to track for problems within the system.

remaining allocation balance statement had to be provided to the department / division on a monthly basis. As a part of the system, the Operations Manager desired a report that summarized the forms usage for inventory purposes. With these parameters, the first step in the design was to evaluate the current billing package to see if it would provide the necessary reports. That proposal was rejected when it became apparent that the billing system did not account for duplex print correctly.

The parameters given for the system made it easy to set up the cycle of the system. Daily data would be accumulated and summarized in a month-to-date dataset. The month-to-date data would be summarized to produce the necessary summary reports and the printer usage statements monthly. A macro was developed to pull off the month name and number from a parameter statement. The month name was appended to each month-to-date file so that a rerun of the system could be accomplished easily. The month-to-date file could be recreated from rerunning the daily jobs with the system accounting data. The previous day's month-to-date file was saved as an old month-to-date file to facilitate rerunning if necessary.

When agreement on the cycle of the system and the necessary files for producing the statements and summaries was reached, the next step was to determine how much information must be kept. The task began with the daily data. The minimum amount of information that needed to be tracked included the account number, job number, date and time of the job, the printer used, form used, lines printed and pages printed. Since the 3825 produces output in both a simplex (one-sided) or duplex (two-sided) mode, it was necessary to separate the simplex from the duplex printing. An additional piece of information needed to be kept, that being total sheets printed. If total sheets printed equalled pages printed, then the output was simplex. Otherwise the output was duplex. If the output was printed a line printer, that field would have no data.

Line printer jobs presented some problems. Since users printed on various forms and developed their own programs for printing their own data, sometimes the accounting data did not accurately present the number of pages actually printed. This was very evident with programs that printed either pressure sensitive labels or cheshire labels. In one case, the number of pages printed was one and the number of lines printed was extremely large. In another case, the number of pages printed was large and the number of lines printed did not reflect the true size of the paper the job was printed on. Because of these cases, the number of lines printed was used to calculate an estimated number of pages printed based on the type of form being printed on. This created another variable of data that must be kept, estimated pages printed.

After the month has completed and the last daily run is made, a summary run is done. The summary run will summarize all of the daily data into monthly totals by department. After the summary of all totals by department, several calculations are done. The calculations take the lines printed from the impact printer fields and calculates approximate pages printed or labels produced. The reason for this is that user departments sometimes generate the labels or the impact printing and do not always use a channel one command to indicate top of page. They will count lines and print blanks to force to top of page in several instances. This condition was discovered when a department had a very low page count but over 200,000 lines printed. After the calculations have been made, then the department totals are merged with the previous month's budget file and a current status report is produced. The status report shows what the original allocation was for a department and the allocation remaining. If the allocation remaining is less than 10% of the original allocation a warning message is issued for obtaining the necessary requisition numbers from the department so that the next run will produce the necessary transactions for the charge back.

After the summary run is done, this is reviewed by the billing person to validate that the usage by various departments looks reasonable. If it appears that a significant increase in print was done, the billing person will print off detail for the department and review the detail with a designated person within the department. This allows us to be able to correct or reassign print if need be. In addition to corrections, if a department's allocation has gone negative, the billing person will acquire the necessary requisition information to charge back to the department the excess usage. The department's profile record is updated to show the requisition information, the summary run is accepted. Once the summary is accepted, the final run is made.

The final run of data is the most complex. This portion of the job uses the same information from the summary run but must create a remaining allocation balance file and produce the necessary transactions to update the Madison accounting system. The system copies the allocation remaining file from the previous month as a work file and uses this data and the summary data for the current month to create a work file that represents the current allocation remaining. This work file is then used to create the transactions for the update to the Madison accounting system. The work file is merged with department information to get the requisition information. That information and the negative balance is used to create a transaction to charge back to the department. The transactions are written to a sequential dataset that are sent via tape to the appropriate department for processing. After the transactions are written, the work file is modified and any negative balances are set to zero, so that the next month will have the correct values for calculation. A report that has the current allocation remaining for the various depart-

ments is printed out and sent to each department for verification.

A final report was requested to see what each department has done over a period of time. This has led to a historical file of usages being generated. Once the monthly summary has been done, the data is appended to a yearly file. This file contains the department name and number and a summary of print by type for a month. The file is generated by merging in the previous month file with the summary data. The print by type is kept separate by month number by appending the month number to the print type. Thus if a department is doing simplex printing only, there is a variable by month that would look like SIMPLX01, SIMPLX02, and on until SIMPLX12. A report printing all the monthly usages with a year-to-date summary is generated.

At this time, the inventory portion has not been developed. The operations manager has changed his mind on how he would like print forms inventoried. If a decision is made, the data is available to generate the files and the reports for his department.

Developing the chargeback system was not difficult once the ideas were conceptually understood. The design phase was the longest, with everyone trying to determine what information was necessary for reporting and tracking. Once the first report was generated and shared with the involved staff, the staff did request some changes to improve the reporting of the data from the system. The user of the system have accepted the product as it was designed.

The system is now up and running and has produced one year of "shadow bills" that were accepted by the user community. The actual billing cycle begins with the new fiscal year and chargeback will begin with the first actual bill in August. The unknown has finally been entered.

```

-----
OPTIONS NOSOURCE NOSOURCE2 NOMACROGEN;
%INCLUDE SOURCLIB(VMACSMF, VMAC6, VMAC26J2);
%INCLUDE SOURCLIB(MONTHMAC);
%BD
DATA
  ID (KEEP=ID SYSTEM)
  _VAR6
  _VAR26J2
  _SMF
OUTPUT ID;
  _CDE6
  _CDE26J2;
RUN;
PROC SORT DATA=TYPE6;
  BY LOCLINFO READTIME;
PROC SORT DATA=TYPE26J2;
  BY LOCLINFO READTIME;
DATA TYPE626;
  MERGE TYPE6 (IN=IN6) TYPE26J2 (IN=IN26);
  BY LOCLINFO READTIME;
  IF IN6 AND IN26;
RUN;
DATA PRNTLINE
  ( KEEP = JOB JESNR LOCLINFO READTIME ACCOUNT1
    PAGECNT SHEETPRN NRLINES FORM
    FORMTYPE OUTDEVCE);
  SET TYPE626;
  FORMAT FORMTYPE $8.;
  ACCOUNT1 = ACCT;
  IF FORM = 'LAND' OR FORM = 'PORT' THEN DELETE;
  IF OUTDEVCE NOT = 'PRT1' AND
    OUTDEVCE NOT = 'PRT2' AND
    OUTDEVCE NOT = 'PRT99'
  THEN DELETE;
  FORMTYPE = 'USER';
  IF SUBSTR(FORM,1,1) = 'A' OR
    FORM = 'STD'
  THEN FORMTYPE = 'ISFORM';
  IF SUBSTR(FORM,1,3) = 'A50'
  THEN FORMTYPE = 'CHESHIRE';
  IF SUBSTR(FORM,1,1) = 'C'
  THEN FORMTYPE = 'LABEL';
  IF OUTDEVCE = 'PRT99'
  THEN FORMTYPE = 'DUPLEX';
  IF PAGECNT = SHEETPRN AND OUTDEVCE = 'PRT99'
  THEN FORMTYPE = 'SIMPLEX';
  IF OUTDEVCE = 'PRT1'
  THEN FORMTYPE = 'DUPLEX';
  IF PAGECNT = SHEETPRN AND OUTDEVCE = 'PRT1'
  THEN FORMTYPE = 'SIMPLEX';
  IF ACCOUNT1 = '9201'
  THEN ACCOUNT1 = '9021';
RUN;

```

```

PROC SORT DATA=PRNTLINE;
  BY ACCOUNT1;
PROC SORT DATA=BILL2.ACCOUNT;
  BY ACCOUNT1;
DATA PRNTACCT;
  MERGE PRNTLINE (IN=INPRT) BILL2.ACCOUNT (IN=INACCT);
  BY ACCOUNT1;
  IF INPRT AND INACCT;
RUN;
DATA ERRACCT;
  MERGE PRNTLINE (IN=INPRT) BILL2.ACCOUNT (IN=INACCT);
  BY ACCOUNT1;
  IF INPRT AND NOT INACCT;
RUN;
PROC DATASETS LIBRARY=BILLING;
  DELETE OLD&BNAME;
PROC APPEND BASE=BILLING.OLD&BNAME DATA=BILLING=DETL&BNAME FORCE;
PROC DATASETS LIBRARY=BILLING;
  DELETE DETL&BNAME ERR&BNAME;
PROC APPEND BASE=BILLING.DETL&BNAME DATA=PRNTACCT FORCE;
PROC APPEND BASE=BILLING.ERR&BNAME DATA=ERRACCT FORCE;
PROC PRINT DATA=BILLING.ERR&BNAME;

```

Figure 1.

```

-----

%MACRO BD;
/* THIS MACRO GENERATES A MACRO VARIABLE CONSISTING OF
   THE THREE LETTER ABBREVIATION OF A MONTH DERIVED FROM
   A SYSPARM VALUE IN THE FORM OF MMYY */
DATA _NULL_;
  BNME = ' ';
  CALL SYMPUT('BNAME', PUT(MDY(%SUBSTR(PRO($PAPER ),2PASS, SPELL, NOUP, 1,
                                     %SUBSTR(PRO($PAPER ),2PASS, SPELL, NOUP, 3,
RUN;
%MEND BD;

%MACRO BNUM;
/* THIS MACRO GENERATES A MACRO VARIABLE CONSISTING OF
   A CONVERSION FROM MONTH TO NUMBER WITHIN THE FISCAL YEAR
   USING THE SYSPARM VALUE OF MM */
DATA _NULL_;
  MP = SUBSTR("PRO($PAPER ),2PASS, SPELL, NOUP",1,2);
  FORMAT MNTHNUM 2.;
  MNTHNUM = INPUT(MP,2.);
  MNTHNUM = MNTHNUM + 1;
  IF MNTHNUM < 08
    THEN DO;
      MNTHNUM = MNTHNUM + 5;
    END;
  ELSE DO;
      MNTHNUM = MNTHNUM - 07;
    END;
  CALL SYMPUT('BMONTH', PUT(MNTHNUM,Z2.));
RUN;
%MEND BNUM;

%MACRO BP;
DATA _NULL_;
  BNME = %SUBSTR(PRO($PAPER ),2PASS, SPELL, NOUP,1,2) - 1;
  %IF BNME = 0
    %THEN BNME = 12;
  CALL SYMPUT('BPREV', PUT(MDY(BNME,1,
                                     %SUBSTR(PRO($PAPER ),2PASS, SPELL, NOUP,3,2)),M
RUN;
%MEND BP;

```

Figure 2.