Abstract

This paper is for those SAS® software users who wish to retrieve and process data residing in an IBM Database 2 (DB2) database, but who do not yet have the newly released SAS/DB2® software. We will illustrate a method to use in the interim which enables a SAS user to retrieve data from a DB2 data base and then process that data using the SAS language. This paper may be of interest to other SAS users because the mechanics of this procedure can be applied to other situations which might arise where data reside in a data base that can not be easily accessed by base SAS software.

Introduction

At the Federal Reserve Board, nonfinancial data for every depository institution in the country are stored in an IBM IMS data base. We retrieve and process data residing in this data base by a simple two step procedure. We first use an inhouse programming language, which was in use prior to the Board’s installation of the SAS System, to retrieve the desired data and write them to a temporary sequential file. We then use base SAS software to read the newly created file and to process the data.

We have used this two step procedure rather than SAS/IMS-DL/IT® because we were not familiar with that software and we could not invest the time necessary to become conversant with it, since the Board had planned for and currently is converting the data base from IMS to DB2.

The conversion of the data base has created a temporary difficulty for Board SAS users in that the Board does not yet have the newly released SAS/DB2 software. The purchase and installation of the software will be forthcoming, but in the meantime, we will have to rely on a variation of our two step procedure to retrieve and process data residing in the DB2 data base. Specifically, the data will be retrieved from the data base and written to a sequential file using IBM’s Structured Query Language (SQL) and IBM’s Query Management Facility (QMF). SQL is the language used to retrieve and manipulate data residing in the DB2 data base. QMF is an interactive facility that enables a user to retrieve selected data, using SQL, and to produce formatted reports of that data. Once the data have been written to a sequential file, SAS software, as before, can be used to read the file and process the data.

Process Description

Data can be retrieved from a DB2 data base by executing a QMF QUERY, either interactively or in batch. The QUERY, written in SQL, contains the criteria to select the desired data. To run a QUERY interactively, which is the focus of this paper, it must be submitted for execution by way of the QMF QUERY PANEL, one of four panels or screens used in QMF.

After the QUERY has run and the retrieved data can be viewed from the QMF REPORT PANEL, the data can be written to a sequential data set using the QMF EXPORT command. The EXPORT command can be executed one of two ways. If the QUERY retrieved only a few hundred observations, the user can directly execute the EXPORT command and let QMF automatically allocate the data set to which the data will be written. If several thousand observations were retrieved, the user can allocate a larger data set, from within QMF, and then export the data. The method chosen to export the data will be largely determined by how the QMF governor, which monitors execution (CPU) time and lines of output, is set.

If a data set is to be allocated, it can be easily done with QMF by entering ‘ISP’ on the command line of the QMF REPORT PANEL. This will bring up the ISP/PDF Primary Option menu from which the Utilities option (3.2) can be selected and the data set allocated. When allocating the data set, the lower level qualifier can be either ‘DATA’ or ‘REPORT’. The choice of qualifier will determine the syntax of the EXPORT command which in turn will determine how the data will be written to the data set when the command is executed. If ‘DATA’ is used, the data will be written to the data set in nonstandard form, whereas if ‘REPORT’ is used, the data will be written in standard form. The form of the data is significant in that it will determine how the file can be later read using SAS software.

After allocating the data set, exit from ISP/PDF back to the QMF REPORT PANEL. The EXPORT command can now be executed by entering on the command line of the panel either ‘EXPORT DATA TO DAT’ or ‘EXPORT REPORT TO DAT’, depending on what was used as the lower level qualifier when the data set was allocated. When the EXPORT command is executed, the lower level qualifier will automatically be appended to the data set name.

After the data have been written to the data set, the data can be edited to determine the columns in which each data field are in and to delete any extraneous data, such as header records. Once the columns have been determined, the file can be read and the data processed using the excellent data modification and report writing capabilities of the SAS System.

An Example

To further illustrate the QMF/SAS version of our two step procedure, we will examine how it can be used to generate an alphabetic listing of data retrieved from a DB2 table.

For our example, selected data for all active insured commercial banks and savings & loan associations in the states of Virginia and North Carolina are retrieved from the DB2 table CUST.SRV_ATTRIB by the QMF QUERY listed in Appendix A. The retrieved data, consisting of a few hundred observations, are then written in standard form to data set DB2DATA.REPORT by the QMF command ‘EXPORT REPORT TO DB2DATA’ listed in Appendix B. The data set is then opened to determine the columns of each data field and to delete extraneous data. The first step of the process is now complete. The desired subset of data has been retrieved from the DB2 data base and is presently resident in a sequential file that can be easily read using base SAS software.

To read an external or non-SAS file, the INFILE and INPUT statements must be used in a DATA step. The INFILE statement identifies the external file to be read with the INPUT statement. The INPUT statement describes the order and format of the input record and assigns values to the corresponding SAS variables used in the DATA step. Depending on the type of data to be described, column, list, formatted or named input may be used in the INPUT statement. Standard form data, stored with one digit or character per byte, can be read with all four input types. Non-standard data, such as binary or packed decimal, can be read only with formatted input.

In our SAS program listed in Appendix C, the INFILE statement identifies the external file to be read as DB2DATA. The input record, which is in standard form, is described using column input. If the data were in nonstandard form, formatted input would have to be used. If this were the case, the packed decimal (PD,W) informat would be used to read numeric data and the $CHAR informat would be used to read character data.

After the file has been read and the data are in SAS data set WORK.READIN, PROC SORT is used to sort the data, first by state (ST NUM), then by type of institution (CHRT_TYP) and finally alphabetically by institution name (NAME). Descriptive labels are then assigned to the numeric values of the variables ST NUM, CHRT_TYP and ATHR_CD using PROC FORMAT. These labels will be associated with their respective variables later when the report is printed.

After the data are properly sorted and the variable formats defined, the data are printed using PROC PRINT. We used the following
PRINT options and statements, rather than the default settings, to generate a more descriptive, customized listing. The BY statement is used to form the subgroups in which the variables named in the VAR statement are to be printed. The N option is used to count the number of observations in each BY group. The ID statement is used with the BY statement to suppress the printing of repetitive values of the BY variable and to suppress the printing of observation numbers and the BY line. The PAGEBY statement causes a new page to be printed when the BY statement variable CHRT_TYP changes value. The LABEL statement is used in conjunction with the SPLIT= option to assign descriptive labels to the variable names. The FORMAT statement associates the labels defined in the FORMAT procedure with the values of the appropriate variables and the TITLE statements provide descriptive report titles.

Appendix D contains an example of the output listing generated by our SAS program. The listing could have been generated by using SQL and QMF exclusively. However, we prefer using our two step approach because, aside from being devoted SAS users, we feel that it gives us greater flexibility to efficiently process data stored in the DB2 data base. For example, we found that when working with a large number of observations, it was generally quicker to MERGE and SORT the data using the SAS language rather than JOIN and sort(ORDER BY) the data in a QMF QUERY using SQL.

Conclusion

We have illustrated for those SAS software users who are without the benefit of SAS/DB2 software how the SAS System can be used with IBM's Query Management Facility to easily retrieve and process data residing in a DB2 data base. By describing how this can be accomplished, we have also shown the ease with which the SAS System can be used with another system to efficiently process data, regardless of the type of data base in which the data reside.

Postscript

Several weeks after this paper was submitted, the Board purchased and installed SAS/DB2 software. We found during the initial testing that the software greatly expedites the retrieval and processing of data residing in the DB2 data base. For example, a user can retrieve data from one or more tables and create a temporary or permanent SAS data set, all in one PROC. Appendix E contains an example of a SAS program that uses this PROC to generate the same output listed in Appendix D. Needless to say, we are eagerly looking forward to when the software will be placed into full production.

Footnote

1/ If the retrieved data is to be used to generate the same report repetitively, with only the values of the data changing, the entire two step procedure can be set up to run in batch. However, a discussion of how to run QMF in batch is not within the scope of this paper.

Acknowledgments

SAS® and SAS/IMS-DLI® are registered trademarks of SAS Institute Inc., Cary, North Carolina, USA.

SAS/DB2® is a trademark of SAS Institute Inc., Cary, North Carolina, USA.

I would like to thank Darrel Parke, Jackie McDaniel, Frank Fry, Sam Sowinski, and Laura Bauer for critiquing this paper.

This paper does not necessarily reflect the views of the Board of Governors of the Federal Reserve System; therefore, no official endorsement should be inferred.

References


Author Contact

Hank Leddon
Federal Reserve Board
20th and C Streets N.W.
Washington, D.C. 20551
(202) 728-5866

APPENDIX A - QMF QUERY

```sql
SELECT IB_DBBB_TYPE, BOARD_NM, CITY, STATE_CD, CHTR_TYPE_CD, CHTR_AUTH_CD FROM CUSTOM.SRV_ATTRIBUTES WHERE (DT_END=99999999 AND EST_TYPE_CD=1) AND (INSR_PRI_CD=1 AND CHTR_TYPE_CD=200) OR (INSR_PRI_CD=2 AND CHTR_TYPE_CD=310) AND (STATE_CD<>37 OR STATE_CD=51)
```

1=HELP 2=RUN 3=END 4=PRINT 5=CHART 6=DRAW
7=BACKWARD 8=FORWARD 9=FORM 10=INSERT 11=DELETE 12=REPORT

SCROLL >>>> PAGE

1387
### APPENDIX B • QMF REPORT and EXPORT COMMAND

<table>
<thead>
<tr>
<th>REPORT ID</th>
<th>DBSP</th>
<th>BOARD</th>
<th>TYPE</th>
<th>NM</th>
<th>CITY</th>
<th>CHTR</th>
<th>CHTR AUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>5511350000</td>
<td>51</td>
<td>DOMINION BK SHNDH VLY NA</td>
<td></td>
<td></td>
<td>BRIDGEWATER</td>
<td>51</td>
<td>200</td>
</tr>
<tr>
<td>1600926600</td>
<td>37</td>
<td>FIRST FS&amp;LA OF PI</td>
<td></td>
<td></td>
<td>GREENVILLE</td>
<td>37</td>
<td>310</td>
</tr>
<tr>
<td>1600757000</td>
<td>51</td>
<td>COLONIAL S&amp;LA</td>
<td></td>
<td></td>
<td>RICHMOND</td>
<td>51</td>
<td>310</td>
</tr>
<tr>
<td>5370358000</td>
<td>37</td>
<td>REPUBLIC BT&amp;C</td>
<td></td>
<td></td>
<td>CHARLOTTE</td>
<td>37</td>
<td>200</td>
</tr>
<tr>
<td>1600253300</td>
<td>51</td>
<td>BANK OF CARROLL</td>
<td></td>
<td></td>
<td>HILLSVILLE</td>
<td>51</td>
<td>200</td>
</tr>
<tr>
<td>1600735900</td>
<td>37</td>
<td>DOMINION FS&amp;LA</td>
<td></td>
<td></td>
<td>McLEAN</td>
<td>37</td>
<td>310</td>
</tr>
<tr>
<td>5511450000</td>
<td>51</td>
<td>FIRST VIRGINIA BK SOUTH</td>
<td></td>
<td></td>
<td>HURT</td>
<td>51</td>
<td>200</td>
</tr>
<tr>
<td>1600735900</td>
<td>51</td>
<td>DOMINION FS&amp;LA</td>
<td></td>
<td></td>
<td>McLEAN</td>
<td>37</td>
<td>310</td>
</tr>
<tr>
<td>5511425000</td>
<td>51</td>
<td>FIRST VIRGINIA BK SOUTH</td>
<td></td>
<td></td>
<td>HURT</td>
<td>51</td>
<td>200</td>
</tr>
<tr>
<td>1600735900</td>
<td>51</td>
<td>DOMINION FS&amp;LA</td>
<td></td>
<td></td>
<td>McLEAN</td>
<td>37</td>
<td>310</td>
</tr>
<tr>
<td>5511379000</td>
<td>51</td>
<td>BANK OF CARROLL</td>
<td></td>
<td></td>
<td>HILLSVILLE</td>
<td>51</td>
<td>200</td>
</tr>
<tr>
<td>5511425000</td>
<td>51</td>
<td>FIRST VIRGINIA BK SOUTH</td>
<td></td>
<td></td>
<td>HURT</td>
<td>51</td>
<td>200</td>
</tr>
<tr>
<td>1600735900</td>
<td>51</td>
<td>DOMINION FS&amp;LA</td>
<td></td>
<td></td>
<td>McLEAN</td>
<td>37</td>
<td>310</td>
</tr>
<tr>
<td>5511450000</td>
<td>51</td>
<td>FIRST VIRGINIA BK SOUTH</td>
<td></td>
<td></td>
<td>HURT</td>
<td>51</td>
<td>200</td>
</tr>
</tbody>
</table>

**APPENDIX C • SAS CODE and JCL**

```sas
//BNKLIST JOB (job card information)
//STEP1 EXEC SAS
//DB2DATA DD DSN=MHAL00.DB2DATA.REPORT,DISP=SHR
//SYSIN DD *

/* THIS PROGRAM SORTS & THEN PRINTS DATA WHICH WAS RETRIEVED */
/* FROM A DB2 DATA BASE USING SQL & QMF. */

DATA READIN; INFIlE DB2DATA; INPUT DSB 1-12 NAME $ 15-42 CITY $ 45-64 ST_NUM 70-71 CHRT_TYP 77-79 ATHR_CD 84-85;
DSB=DSB/1000;

PROC SORT DATA=READIN; BY ST_NUM CHRT_TYP NAME;
PROC FORMAT;
VALUE STVAL
37='N. CAROLINA' 51='VIRGINIA';
VALUE CHRTVAL
200='COMMERCIAL BANK' 310='SAVINGS & LOAN';
VALUE ATHRVAL
-1='N/A' 1='FEDERAL' 2='STATE';
PROC PRINT DATA=READIN N SPLIT=*;
VAR DSB NAME CITY ATHR_CD;
BY ST_NUM CHRT_TYP;
PAGEx=CHRT_TYP;
LABEL DSB='BANK NUMBER' NAME='BANK NAME' ST_NUM='STATE' CHRT_TYP='ENTITY TYPE' ATHR_CD='CHARTER AUTHORITY';
FORMAT ST_NUM STVAL CHRT_TYP CHRTVAL ATHR_CD ATHRVAL;
TITLE1 'LISTING OF ACTIVE INSURED COMMERCIAL BANKS AND';
TITLE2 'SAVINGS & LOAN ASSOCIATIONS BY STATE';
```

1388
APPENDIX D - EXAMPLE OF OUTPUT

LISTING OF ACTIVE INSURED COMMERCIAL BANKS AND SAVINGS & LOAN ASSOCIATIONS BY STATE

<table>
<thead>
<tr>
<th>STATE</th>
<th>ENTITY TYPE</th>
<th>NUMBER</th>
<th>NAME</th>
<th>CITY</th>
<th>CHARTER AUTHORITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>N. CAROLINA</td>
<td>COMMERCIAL BANK</td>
<td>5370975</td>
<td>AMERICAN B&amp;TC</td>
<td>CITY POINT</td>
<td>STATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5371440</td>
<td>AVERY COUNTY BK</td>
<td>HAYWARD</td>
<td>STATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5370160</td>
<td>BANK OF BLADENBORO</td>
<td>BLADENBORO</td>
<td>STATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5371420</td>
<td>BANK OF CURRITUCK</td>
<td>MOYOCK</td>
<td>STATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5370685</td>
<td>UNITED NB</td>
<td>FAYETTEVILLE</td>
<td>FEDERAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5376281</td>
<td>VILLAGE BK</td>
<td>CHAPEL HILL</td>
<td>STATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5372230</td>
<td>NACHOVIA B&amp;TC NA</td>
<td>NINSON-SALEM</td>
<td>FEDERAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5378629</td>
<td>YADKIN VALLEY B&amp;TC</td>
<td>ELKIN</td>
<td>STATE</td>
</tr>
</tbody>
</table>

n= 67

APPENDIX E - EXAMPLE OF PROC DB2EXT

//SASDB2 JOB (job card information)
//STEP1 EXEC SASDB2
//SYSIN DD *

PROC DB2EXT OUT=BNKDATA;
  SELECT ID_DSB_TYPE, BOARD_NM, CITY, STATE_CD, CHRT_TYPE_CD, CHRT_AUTH_CD
  FROM CUST_SRV_ATTRIBUTES
  WHERE (BT_END=9999999 AND EST_TYPE_CD=1) AND
    ((INSUR_PRI_CD=1 AND CHRT_TYPE_CD=200) OR
     (INSUR_PRI_CD=2 AND CHRT_TYPE_CD=310)) AND
    ((STATE_CD=37 OR STATE_CD=51))
  RENAME 1=DSB 2=NAME 4=ST_NUM 5=CHRT_TYP 6=CHRT_TYPE;
  FMT 2=** 5=** 9=.

DATA SELECT;
  LENGTH NAME $ 28 CITY $ 20;
  SET BNKDATA;
  DSB=DSB/10000;

PROC SORT DATA=SELECT OUT=SELECT;
  BY ST_NUM CHRT_TYP NAME;

PROC FORMAT:
  VALUE STVAL
    37='N. CAROLINA'
    51='VIRGINIA';
  VALUE CHRTVAL
    200='COMMERCIAL BANK'
    310='SAVINGS & LOAN';
  VALUE ATHRVAL
    -1='N/A'
    1='FEDERAL'
    3='STATE';

PROC PRINT DATA=SELECT N SPLIT=*;
  VAR
    DSB NAME CITY ATHR_CD;
    BY ST_NUM CHRT_TYP;
    ID ST_NUM CHRT_TYP;
    PAGE BY CHRT_TYP;
  LABEL
    DSB='BANK NUMBER'
    NAME='BANK NAME'
    ST_NUM='STATE'
    CHRT_TYPE='ENTITY TYPE'
    ATHR_CD='CHARTER AUTHORITY';
  FORMAT ST_NUM STVAL. CHRT_TYP CHRTVAL. ATHR_CD ATHRVAL. DSB 8.;
  TITLE1 'LISTING OF ACTIVE INSURED COMMERCIAL BANKS AND';
  TITLE2 'SAVINGS & LOAN ASSOCIATIONS BY STATE';