

"How I Learned to Stop Worrying & Love (?) Macros"

Ann Stephan, Texas State Comptroller, Austin, TX

Once upon a time, some users requested a job which would allow them to enter a zip code & get a report showing taxpayer active & out-of-business locations within that zip (with their master address information). This was a simple, short job - just read one VSAM file & print a report. They used this report for several years and then they decided that they sometimes needed to be able to select & print only location information (no master information). It was easy to just copy the program, change the few lines necessary to create the different report. It was also easy to copy the PROC which they invoked to run the job. Then they decided that they also wanted to be able to request a different sort order and then Active only & locations only. When anything changed – such as a data conversion from VSAM & S2K to DB2, all of the individual programs had to be changed, but we did it.

Here is part of the original 600+ lines of code:

```
000025 PROC SQL;
000026   CONNECT TO DB2 (SSID=DSNQ);
000027   CREATE TABLE ADDRS AS
000028     SELECT * FROM CONNECTION TO DB2
000029       (SELECT ICL_CD,
000030             AD_STR_POB_TX AS LOC_ADDR, CITY_NM AS LOC_CITY,
000032             JRIS_CITY_NM AS JUR_CITY, ST_CD AS LOCSTATE,
000034             AD_ZP AS LOCZIP, AD_PLUS_FOUR_ZP AS LOCZIP4,
000036             CMPT_CNTY_CD AS LOCCNTY, TP_NTRL_ID AS TPNTRL, A.AD_ID
000039             FROM TAXPYR.VADDRESS01 A,
000040             TAXPYR.VADDRESS_PURP01 P
000041             WHERE AD_ZP = '78746'
000042                   AND P.AD_PURP_CD = 'E'
000043                   AND A.AD_ID = P.AD_ID
000044             FOR FETCH ONLY);
000046   %PUT &SQLXMSG;
000047   %PUT &SQLXRC;
000048   DISCONNECT FROM DB2;
000049   QUIT;
```

CHUNKLOC

```
000050 PROC SORT DATA=ADDRS; BY AD_ID; ** For Merge step;
000051 RUN;
000053 ***** Using AD_IDs found in above step, select the business location info;
000055 FILENAME TEMPSQL '&&TMPSQL' DISP=NEW
000056             SPACE=(CYL,(100,2)) LRECL=80 BLKSIZE=6160;
000057 FILENAME TEMP2 '&&TMPSQ2';
000059 DATA _NULL_ FILE TEMPSQL;
000060   COUNT = 1; SET = 1;
000061   CHUNK=1000;
000062   DO COUNT = 1 TO CHUNK;
000063     SET ADDRS END=LAST;
000064     IF COUNT = 1 THEN DO;
000065       PUT 'CREATE TABLE BLOCS' SET ' AS SELECT * FROM CONNECTION TO DB2';
000066       PUT '(SELECT BUS_LOC_NM AS LOC_NAME, AD_ID, '
000067             BUS_LOC_ID AS BLOCID';
000068       PUT ' FROM TAXPYR.VBUS_LOC01 WHERE AD_ID IN (';
000069             END;
000070       IF NOT LAST THEN PUT AD_ID 13.;';
000071       IF LAST THEN DO;
000072         CALL SYMPUT('NUMSETS',LEFT(PUT(SET,6)));
000073         PUT AD_ID 13.) FOR FETCH ONLY);';
000074       COUNT = CHUNK + 1;
000075       END;
000076       IF COUNT=CHUNK THEN DO;
000077         PUT AD_ID 13.) FOR FETCH ONLY);';
000078         COUNT=0;
000079         SET+1;
000080       END;
000081     END;
000082 RUN;
000084 DATA _NULL_ FILE TEMP2;
000085   %PUT &NUMSETS;
000087   DO N=1 TO &NUMSETS;
000088     DSN='BLOCS'ITRIM(LEFT(N));
000089     PUT 'PROC APPEND BASE=BLOCS DATA=' DSN ';;
000090   END;
000092 RUN;
```

```

000094 PROC SQL;
000095     CONNECT TO DB2 (SSID=DSNQ);
000096     %INCLUDE TEMPSQL ;
000098 %PUT &SQLXMSG;
000099 %PUT &SQLXRC;
000100 DISCONNECT FROM DB2;
000101 QUIT;
000102 %INCLUDE TEMP2;
000104 PROC SORT DATA=BLOCS; BY AD_ID;
000106 RUN;
000108 DATA LOCADDRS; MERGE ADDR (IN=A)
000109                      BLOCS (IN=B);
000110                      BY AD_ID;
000111 IF A=1 & B=1;
000112 RUN;
                                ** ( Code in Lines 50 - 112 is repeated to read select data from 2 more tables.);

000245 DATA CITYCODE; MERGE LOCADDRS (IN=A)
000246                      JURIS (IN=B);
000247                      BY TP_ID UNIT_NR;
000248 IF LAST.UNIT_NR;
000249 IF A=1;
000250 IF ('0' <= LOC_ADDR <= '9') THEN DO;
000251     ADDRNUM = SCAN(LOC_ADDR,1,' ');
000252     ADDRTEXT = SUBSTR(LOC_ADDR,(LENGTH(ADDRNUM)+2));
000253     END;
000254 ELSE DO;
000255     ADDRNUM = '';
000256     ADDRTEXT = LOC_ADDR;
000257     END;
000258
000259 OUTPUT;
000260 RUN;

```

```

● GETMASTS
000250 DATA CITYCODE; SET CITYCODE;
000252 TP_IDX=PUT(TP_ID,PD6.);
000253 XPNUMX =INPUT(TP_IDX,$6.);
000254 LINK NAMEIN;
000255 RETURN;
000257 NAMEIN:
000258     INFILE MASTINFO VSAM KEYGE GENKEY KEY=(XPNUMX);
000259     INPUT @;
000260     RC = _JORC_;
000261     FBC = _FDBK_;
000263     INPUT @ 1 TPNO PD6.
000264             @ 7 TPNAME $80.
000265             @ 87 TP_ADDR $50.
000266             @ 146 TPZIP $5.
000267             @ 151 TPZIP4 $4.
000268             @ 205 TPSTATE $2.
000269             @ 207 TP_CITY $30.
000270             @ 237 TPCNTY PD2.
000271             @ 285 TORG $2.
000272             @ 295 TPNTRL PD7. ;
000275 IF TP_ID = TPNO THEN OUTPUT;
000276 INPUT;
000277 RETURN;
000278 RUN;

```

```

● CRALLDS
000280 DATA ACTIVES OOB; SET CITYCODE;
000281
000282 IF OOBDATE = . THEN OUTPUT ACTIVES;
000283 ELSE DO;
000284     CKDATE = TODAY() - 4*(365); * PER FOEHNER 4/18/94;
000285     TDAY = TODAY();
000286     IF OOBDATE NE . AND OOBDATE GT CKDATE;
000287     FORMAT CKDATE TDAY OOBDATE MMDDYY8.;
000288     OUTPUT OOB;
000289     END;
000290 RUN;

```

The last 300 lines are data nulls which produce the reports. I didn't include them in this paper because I felt they would just make the paper longer without adding anything.

At this point, there were 8 programs & 8 PROCs to maintain - with only minor differences between them. When they submitted a request to be able to select by zip, city, county, special purpose district or transit authority, it was quite obvious that there was to be no end to the various parameters they were going to want, so I decided that it was finally time to do some research.

Over the years, I have attended and/or read several papers on using macros. For some reason, it just did not click for me. At SCSUG '97, I attended the hands-on workshop on "Beginning Macros" presented by Dana Raffiee (Destiny Corporation), and it began to look like something that I could use. So, when I say "discover", I really should be saying rediscover. The first thing I discovered was %INCLUDE - which allows one to store code in a PDS and then %INCLUDE it in a job. This means that several jobs with varying selection criteria can call the common code. If changes are needed in the common code, you have to change it only once, and all of the programs accessing it will use the updated code.

I looked at the code & broke it into pieces which could be shared. There were still eight separate programs and PROCs to maintain, but by making a change in one PDS member, the change would be in effect in every program which used it. This looked okay, until I realized that I needed to get the user-provided parameters into the code which was being "included". So, after attending numerous classes and presentations over the years, I finally I saw a need for the dreaded "M" word (macros). I had won a "SAS Macro Language Reference" manual at a conference and started reading it. One of the first items discussed is %LET.

On line 20, the %LET statement gives the variable ZIP the value of the zip code which the user enters via the proc. Whenever SAS reads an "&ZIP", the &ZIP is replaced with the value from line 20. In the following code the &ZIP can be seen on line 40. It also is in the titles in the PDS member on line 54. The Users also requested that the job name appear on the report, so this was easily accomplished with the %LET on line 21 (the &job is also used in the PDS member on Line 54).

```

000020 %LET ZIP = '%%%%%%%%';
000021 %LET JOB = 'ZIPALLA';
000024 PROC SQL;
000025     CONNECT TO DB2 (SSID=DSNQ);
000026     CREATE TABLE ADDRS AS SELECT * FROM CONNECTION TO DB2
000028         (SELECT ICL_CD, AD_STR_POB_TX AS LOC_ADDR, CITY_NM AS LOC_CITY,
000031             JRIS_CITY_NM AS JUR_CITY,
000032             ST_CD AS LOCSTATE, AD_ZP AS LOCZIP,
000034             AD_PLUS_FOUR_ZP AS LOCZIP4,
000035             CMPT_CNTY_CD AS LOCCNTY, TP_NTRL_ID AS TPNTRL,
000037             A.AD_ID
000038         FROM TAXPYR.VADDRESS01 A,
000039             TAXPYR.VADDRESS_PURP01 P
000040         WHERE AD_ZP = &ZIP
000041             AND P.AD_PURP_CD = 'E'
000042             AND A.AD_ID = P.AD_ID
000043             FOR FETCH ONLY);
000045 %PUT &SQLXMSG;
000046 %PUT &SQLXRC;
000047 DISCONNECT FROM DB2;
000048 QUIT;
000049
000050 %INCLUDE PDS(CHUNKLOC)
000051     PDS(ACTIVEDS)
000052     PDS(ICLFMT)
000053     PDS(SORTADRA)
000054     PDS(ACTLOCS);

```

This was better but not ideal. The PDS member CHUNKLOC contains the SQL pass-through code to read several DB2 tables. When a change was required, it meant that I had to make the change only once & not 8 times. So I kept playing with it. It has now evolved into one

PROC and one program with several macros. The user enters five parameters via the PROC, and the %LETs provide the info to the rest of the program

The special characters in lines 15 - 19 are supplied by the user when they invoke the PROC. Since they still want the job name to appear in the headings, it is created in line 20 by putting "LOC" in front of the SELBY variable from line 15.

```

000012 * The %let items are filled in by the user when they invoke the RPF TRA.LOCMASTR ;
000015 %LET SELBY = '---';          * ZIP, CNTY,CITY;
000016 %LET SELON = '%%%';          * ZIP CODE, COUNTY CODE, OR JRIS_CITY_NAME;
000017 %LET SORTORD = '&&';          * TP, ADDR, OR LOCN (LOCATION NAME) ORDER;
000018 %LET ACTX = '???';            * ACT (ACTIVE ONLY) OR ALL (ACTIVE & OOB);
000019 %LET MASTX = '@@@';           * LOC (LOCATIONS ONLY) OR MAST (LOCS & MASTS);

000020 %LET JOB = "LOC"&SELBY;       * PUTS "ZIP", "CNTY", OR "CITY" IN HEADINGS;

```

The SELECTBY macro selects the addresses for the zip code, county code, or city name which was entered via the PROC. The CALL SYMPUTs also create macro variables which are then available later in the program. Since macro processing is done before program compile & execution, and SYMPUT assigns its value during program execution, you cannot create the macro variable with SYMPUT and reference it within the same step. The DATA _NULL_ steps (e.g. line 25 below) solved this problem for me.

```

000023 %MACRO SELECTBY;
000024     %IF &SELBY = 'CNTY' %THEN %DO;
000025         DATA _NULL_;
000026             CALL SYMPUT ('CNTYN',TRIM(LEFT(PUT(&SELON,5))));
000027             CALL SYMPUT ('TITLENM', 'COUNTY');
000028         RUN;
000029         PROC SQL;
000030             CONNECT TO DB2 (SSID=DSNQ);
000031             CREATE TABLE ADDR5 AS SELECT * FROM CONNECTION TO DB2
000033                 (SELECT ICL_CD, AD_STR_POB_TX AS LOC_ADDR,
000035                     CITY_NM AS LOC_CITY, JRIS_CITY_NM AS JUR_CITY,
000037                     ST_CD AS LOCSTATE, AD_ZP AS LOCZIP,
000039                     AD_PLUS_FOUR_ZP AS LOCZIP4,
000040                     CMPT_CNTY_CD AS LOCCNTY, TP_NTRL_ID AS TPNTRL,
000042                     A.AD_ID
000043                 FROM TAXPYR.VADDRESS01 A,
000044                     TAXPYR.VADDRESS_PURP01 P
000045                 WHERE CMPT_CNTY_CD = &CNTYN
000046                     AND P.AD_PURP_CD = 'E'
000047                     AND A.AD_ID = P.AD_ID
000048                     FOR FETCH ONLY);
000050             %PUT &SQLXMSG;
000051             %PUT &SQLXRC;
000052             DISCONNECT FROM DB2;
000053             QUIT;
000054         %END;
000055     %ELSE %IF &SELBY = 'ZIP' %THEN %DO;
000056         DATA _NULL_;
000057             CALL SYMPUT ('TITLENM', 'ZIP');
000059         RUN;
000060         PROC SQL;
000061             CONNECT TO DB2 (SSID=DSNQ);
000062             CREATE TABLE ADDR5 AS SELECT * FROM CONNECTION TO DB2
000063                 (SELECT ICL_CD,
000065                     AD_STR_POB_TX AS LOC_ADDR,
000066                     CITY_NM AS LOC_CITY,
000067                     JRIS_CITY_NM AS JUR_CITY,
000068                     ST_CD AS LOCSTATE,
000069

```

```

000070          AD_ZP      AS LOCZIP,
000071          AD_PLUS_FOUR_ZP AS LOCZIP4,
000072          CMPT_CNTY_CD  AS LOCCNTY,
000073          TP_NTRL_ID    AS TPNTRL,
000074          A.AD_ID
000075      FROM TAXPYR.VADDRESS01 A,
000076          TAXPYR.VADDRESS_PURP01 P
000077      WHERE AD_ZP = &SELON
000078          AND P.AD_PURP_CD = 'E'
000079          AND A.AD_ID = P.AD_ID
000080          FOR FETCH ONLY);
000082      %PUT &SQLXMSG;
000083      %PUT &SQLXRC;
000084      DISCONNECT FROM DB2;
000085      QUIT;
000086      RUN;
000087
000088          %END;
000089      %ELSE %IF &SELBY = 'CITY' %THEN %DO;
000090          DATA _NULL_;
000091          CALL SYMPUT ('TITLENM', 'CITY');
000092      RUN;
000093      PROC SQL;
000094          CONNECT TO DB2 (SSID=DSNQ);
000095          CREATE TABLE ADDR5 AS SELECT * FROM CONNECTION TO DB2
000096          (SELECT ICL_CD,
000097              AD_STR_POB_TX AS LOC_ADDR,
000098              CITY_NM      AS LOC_CITY,
000099              JRIS_CITY_NM AS JUR_CITY,
000100              ST_CD       AS LOCSTATE,
000101              AD_ZP       AS LOCZIP,
000102              AD_PLUS_FOUR_ZP AS LOCZIP4,
000103              CMPT_CNTY_CD AS LOCCNTY,
000104              TP_NTRL_ID  AS TPNTRL,
000105              A.AD_ID
000106          FROM TAXPYR.VADDRESS01 A,
000107              TAXPYR.VADDRESS_PURP01 P
000108          WHERE JRIS_CITY_NM = &SELON
000109              AND P.AD_PURP_CD = 'E'
000110              AND A.AD_ID = P.AD_ID
000111              FOR FETCH ONLY);
000112      %PUT &SQLXMSG;
000113      %PUT &SQLXRC;
000114      DISCONNECT FROM DB2;
000115      QUIT;
000116      RUN;
000117
000118          %END;
000119
000120 %MEND SELECTBY;

```

The AD_IDs selected in the SELECTBY are then used to read the other DB2 tables and format the report(s) in the WHICHINC macro:

```

000122 %MACRO WHICHINC;
000123     %IF &ACTX = 'ALL' AND &MASTX = 'MAST' %THEN %DO;
000124         /* ACTIVE & OOB LOCATIONS & THEIR MASTERS */
000125
000126         %INCLUDE PDS(CHUNKLOC)
000127             PDS(GETMASTS)
000128             PDS(CRALLDS)
000129             PDS(ICLFMT)
000130             PDS(SORTALL)
000131             PDS(PRACTLM)
000132             PDS(PROOBLM);
000133     %END;

```

```

000134      %ELSE %IF &ACTX = 'ALL' AND &MASTX = 'LOC' %THEN %DO;
000135                                     /* ACTIVE & OOB LOCATIONS */
000136
000137      %INCLUDE PDS(CHUNKLOC)
000138                PDS(CRALLDS)
000139                PDS(ICLFMT)
000140                PDS(SORTALL)
000141                PDS(PRACTLOC)
000142                PDS(PROOBLOC);
000143
000144                                     %END;
000145
000145      %ELSE %IF &ACTX = 'ACT' AND &MASTX = 'MAST' %THEN %DO;
000146                                     /* ACTIVE LOCATIONS ONLY - WITH THEIR MASTERS */
000147      %INCLUDE PDS(CHUNKLOC)
000148                PDS(GETMASTS)
000149                PDS(CRACTDS)
000150                PDS(ICLFMT)
000151                PDS(SORTACT)
000152                PDS(PRACTLM);
000153
000154                                     %END;
000155
000155      %ELSE %IF &ACTX = 'ACT' AND &MASTX = 'LOC' %THEN %DO;
000156                                     /* ACTIVE LOCATIONS ONLY */
000157      %INCLUDE PDS(CHUNKLOC)
000158                PDS(CRACTDS)
000159                PDS(ICLFMT)
000160                PDS(SORTACT)
000161                PDS(PRACTLOC);
000162
000163                                     %END;
000165 %MEND WHICHINC;

```

The macros are invoked by a *%macroname* statement. A *%macroname* statement does not have to end in a semicolon.

```

000167 %SELECTBY
000169 %WHICHINC

```

I realize that this is probably not the most elegant way to do this, especially if someone is comfortable using macros, but I tried to keep it as simple as possible. This program will probably continue to evolve.

References:

Thomas J. Winn, Jr., "The SAS Macro Language: An Overview", Proceedings of SCSUG '91, pp 267 - 272.

SAS Macro Language, Reference, First Edition.

I would also like to thank the SAS-L gurus. I read most of the postings and so many answers are "try this macro". Seeing so many examples to various problems helped to make macros more easily understood.

Ann Stephan
P O Box 12103
Austin, TX 78711
(512) 463-4430
ann.stephan@cpa.state.tx.us