ABSTRACT

With the advent of macros and the resulting publicity and interest in their use, the power and efficiency of the LINK may easily be overlooked for many appropriate applications. Efficiency needs to be a concern to us all, particularly those of us using mainframe time-sharing environments. The CPU/cost savings potential of techniques such as those described below are substantial, especially when developing SAS systems to be used repetitively in a production mode. The efficiencies we found under MVS hold true for other operating systems running SAS.

This paper will describe when LINKs can, and probably should, be used instead of macros. We will present our real-life experiences as examples to demonstrate the power, similarities and differences in these two approaches as well as the relative efficiencies.

INTRODUCTION

Our continuous challenge is to develop flexible SAS systems, tailored to individual applications by screen-oriented expert user interfaces (we use SAS/AF and SAS/FSP). These systems must produce consistently accurate results within tight time frames and budget constraints. In order to meet these requirements, we rely heavily on the power of the SAS macro language, but have learned it pays to spend time analyzing and testing relative efficiencies of different coding techniques.

A CASE STUDY

A new procedure needed to be developed for analyzing selected survey data for outlier detection. Outliers are data values outside logical or normal limits (e.g., Hired Farm Worker Age < 12 or > 70). Records with especially high and/or low expanded values were to be printed with additional items for reference; and charts were to be produced showing the expanded and/or unexpanded variable value for all observations. Expanded data are data weighted by the sampling rate (number of units a sample represents). Our data covers the U.S. with differing sets of variables needing to be analyzed for each state. These variables often change from survey to survey.

Typically, we have 75 to 100 different variables to analyze for a single survey. A separate table was needed for each analysis variable. Each table had to have appropriate title information, supporting variables, tailored print formats and variable labels. The survey statisticians also wanted to designate for which states each different analysis table would be generated.

This seemed to be a natural for our generalized application approach, driven by parameters from which we generate tailored code (refer to our SUGI 16 paper). Creating macros immediately came to mind and we generated code accordingly.

The examples presented here are simplified versions of the real thing but demonstrate the concepts and techniques used.

The macro approach:

First we designed FEDIT procedure screens for the survey statisticians to use in defining analysis variables and all the accompanying information desired, including selecting the states for each table. Figure 1 shows a pair of FEDIT screens used to define a table. The information entered on these two screens becomes one observation in the resulting parameter dataset. These observations are then read by a SAS program which creates the actual SAS statements shown in Figures 2, 3, 5 and 6. The two sets of SAS code were generated from one pass through the parameter SAS dataset.

Figure 2 shows the macro code generated to prepare the actual data for input into the outlier print and plot routines. Figure 3 shows the generated code to selectively invoke the macros. Figure 4 is the "base" SAS code designed to bring in and use the generated code.
What happened next

When we executed tests using the code in figures 2 through 4, the cost appeared to be extremely high -- 80 SRUs in this example to run through one state’s data. SRUs are System Resource Units, the measure of cost by which we are charged for processing.

Analysis of the generated code

In looking at the actual SAS code generated for execution, we very quickly realized that, because the macro code was physically duplicated each time it was referenced, the executable code became VERY LONG! After exploring possible alternatives, we decided to try generating LINK routines the same way as the macros had been generated, and see if it helped.

The LINK approach

The code generating SAS program was modified to substitute LINKs and RETURNs in place of %MACROs and %MENDs -- See figures 5 and 6. The “base” code was re-ordered to place the LINKs at the end of the data step -- See Figure 7. A comparison of the code in figures 2 and 6 shows that very little has changed.

Results of the LINK approach (Are we having fun yet??)

When the revised code was executed for the same test data, the SRUs dropped from 80 to 30 -- a 62.5 percent saving! We tested both approaches with several sets of parameters and data, and found roughly the same savings. When you consider that this process will be used for all our major surveys (more than one per month) and for all states, the magnitude of savings is impressive. All this without changing the flexibility of the system or the accuracy of the results!

Subsequent elapsed time comparisons on the PC showed even more drastic differences, plus the macro code for this application caused “out-of-memory” problems when we ran more than 5 states!

The finished product

Figure 8 shows two of the output tables created from our case study code. The tables remained identical, only the method and cost changed!

ANALYSIS OF RESULTS

Because the link acts very much like a 2nd or 3rd generation subroutine, and is not physically reproduced in the executable code, the executable SAS code was much shorter. That and the elimination of the macro overhead for those calls resulted in much more efficient SAS code.

OTHER USES AND LIMITATIONS OF LINK

The main limiting factor of LINK is that it must be used within a data step but, as demonstrated in our case study, you can often design your application within this constraint. Keep in mind that LINK routines can be made flexible in many ways, such as setting certain variable values before calling LINK and having your LINK routine vary depending on the value passed to it. Also, macro variables can be used to vary the routine from run to run.

SUMMARY (The Moral Of The Story)

Macros are extremely powerful and versatile but they are not necessarily the best answer to every situation requiring flexibility. Faster, more efficient code can result from taking the time to analyze and test alternatives. The Lowly LINK has many useful places in your SAS code!

References


SAS, SAS/AF and SAS/FSP are the registered trademarks of SAS Institute Inc., Cary, NC

For further information contact:

Barbara Ison
USDA/SID, User Services Branch
Room 4808, South Building
Washington, D. C. 20250
(202) 720-3362

Dania Ferguson
USDA/SID, System Services Branch
Room 5862, South Building
Washington, D. C. 20250
(202) 720-9248
FIGURE 1. EXAMPLE OF FSEDIT SCREENS FOR PARAMETER INPUT:

EDI T SAS DATA SET: SASIN.QASINFO

COMMAND ===> SCREEN

SETUP FOR CHARTS AND TABLES FOR POTENTIAL OUTLIER PRINTS

(Use 'm=n' to move to screen 'n' or use PF11 to shift to screen 2)

VARIABLE TO ANALYZE: B2

TITLE1: MINIMUM EXPANDED DIFFERENCE IN ACRES FOR CORN

TITLE2: MARCH VS. DECEMBER

TITLE3: DIFFERENCES THAT EXPAND TO ZERO HAVE BEEN EXCLUDED

Extra Print Variables Labels for extra Variables **
1. __ __ __ __ __
2. __ __ __ __ __
3. CROPLAND 9.0 CURRENT*CROPLAND* ACRES
4. INCNXOPL 9.0 DECEMBER* CORN* ACRES
5. CRRXOPL 9.0 CURRENT* CORN* ACRES
6. ED 9.1 ADJ * EXP * FACTOR
7. __ __ __ __ __

** Use an asterisk (*) to split each label into max. of 3 print lines. analysis variable may have only 2 print lines in its label.

Expansion type: NONE (NONE, XT, XF, XW, XTC, XFC, XNC, XWC)
This will control minimum expansions using SPS Data Listings expansion macro.

Additional sort variable for Hi-Lo print: __
Plot Character Variable Name: FRAME
Hi-Lo Type: 1 (1=high 20; 2=10 high/10 low; 3=10 high/10 low positive)

EDIT SAS DATA SET: SASIN.QASINFO

COMMAND ===> SCREEN 2

(Use 'm=n' or PF10 for screen 1, PF11 for screen 3)

Mark all types of analysis to perform for each state
'C' for CHART/HLIO 'O' for OUTLIER EDIT 'S' for PRE-SUMMARY

AL(01) C AK(02) C AZ(04) C AR(05) C CA(06) C
CO(08) C CT(09) C DE(10) C FL(12) C GA(13) C
IN(15) C ID(16) C IL(17) C IN(18) C IA(19) C
KS(20) C KY(21) C LA(22) C KY(23) C MO(24) C
WA(35) C WY(36) C WY(37) C ND(38) C OH(39) C
OK(40) C OR(41) C PA(42) C RI(44) C SC(45) C
SD(46) C TN(47) C TX(48) C UT(49) C VT(50) C
VA(51) C WA(53) C WV(54) C WI(55) C WY(56) C

Select criteria for commodity. Data values of 1 are always excluded.

Include records with 0 values (Y/N) N
Include records for which frame(s): B ('L'=List, 'A'=Area, 'B'=Both)
FIGURE 2

EXAMPLE OF GENERATED MACROS (CHMACS):

%MACRO CHL1; ** CHART/HILO FOR CROPLAND ;
IF CROPLAND GT 0 THEN DO;
IF FRAME = 'AREA' THEN PLOTCODE = 'AREA';
ELSE PLOTCODE = 'LIST'; SORTVAL = 0;
SEQ = 1; VARNAME = 'CROPLAND'; EFVALUE = XV;
XVAL1 = ; XVAL1 = 'X'; XVAL2 = 'CROPLAND'; XVAL2 = 'X';
XVAL3 = ; XVAL3 = 'X'; XVAL4 = ; XVAL4 = 'X';
XVAL5 = ; XVAL5 = 'X'; XVAL6 = ; XVAL6 = 'X';
XVAL7 = ; XVAL7 = 'X';
SYSAHILO = 1; UNEXPNVL = CROPLAND;
VALUE = CROPLAND * EFVALUE;
OUTPUT SASOUT.CALC2;
END;
%MEND CHL1; ** END OF CHL1 ;

NOTE: Macros for tables 2 through 21 not shown

%MACRO CHL22; ** CHART/HILO FOR 02 ;
IF 02 GT 0 THEN DO;
IF FRAME = 'AREA' THEN PLOTCODE = 'AREA';
ELSE PLOTCODE = 'LIST'; SORTVAL = 0;
SEQ = 22; VARNAME = '02'; EFVALUE = 1;
XVAL1 = ; XVAL1 = 'X'; XVAL2 = ; XVAL2 = 'X';
XVAL3 = ; XVAL3 = 'X'; XVAL4 = ; XVAL4 = 'X';
XVAL5 = ; XVAL5 = 'X'; XVAL6 = ; XVAL6 = 'X';
XVAL7 = ; XVAL7 = 'X';
SYSAHILO = 1; UNEXPNVL = 02;
VALUE = 02;
OUTPUT SASOUT.CALC2;
END;
%MEND CHL22; ** END OF CHL22 ;

NOTE: Macros for tables 23 through 78 are not shown

FIGURE 3

EXAMPLE OF GENERATED CODE TO USE MACROS FROM FIGURE 2 (CHSELS):

LENGTH XVAL1-XVAL7 $ 1;
SELECT (STATE);
WHEN (1) DO;
  %CHL1; ** CROPLAND; %CHL2; ** CCRNXXPL;
  %CHL3; ** CSOYXXPL; %CHL4; ** CSRXXPL;
  %CHL22; ** 02; %CHL23; ** 03;
  %CHL24; ** 04; %CHL26; ** 06;
  %CHL27; ** L2; %CHL28; ** L3;
  %CHL76; ** L4; %CHL78; ** NHGTOTL;
END;

NOTE: Generated code for states 2 through 55 not shown

WHEN (56) DO;
  %CHL1; ** CROPLAND;
  %CHL2; ** CCRNXXPL; %CHL9; ** CDEBXXPL;
  %CHL76; ** L4; %CHL78; ** NHGTOTL;
END;
OTHERWISE
END;
RETURN;

FIGURE 4

EXAMPLE OF SAS CODE INCORPORATING FIGURES 2 & 3

%* BRING IN GENERATED MACROS;
%INCLUDE CHMACS; *** CODE FROM FIGURE 2;
RUN;

DATA SASOUT.CALC2(KEEP= &BREAKIDS &HILOEFS VARNAME VALUE SORTVAL SYSAHILO EFVALUE XVAL1-XVAL7 SEQ XVAL1-XVAL7 UNEXPNVL);
LENGTH VARNAME $ 8;
SET IN.SASMSTR;
IF REPTIN GT 0 AND CRITERR LT 1;
*** CALL IN MACRO TO CALCULATE EXPANSIONS ON DATA VALUES;
%EXPAND;
*** BRING IN GENERATED CODE TO CALL MACROS;
%INCLUDE CHSELS; *** CODE FROM FIGURE 3;
FIGURE 5

EXAMPLE OF GENERATED LINK CODE (CHLINKS):

CHL1 : ** CHART/HILO LINK FOR CROPLAND;
IF CROPLAND GT 0 THEN DO;
  IF FRAME = 'AREA' THEN PLOTCODE = 'AREA';
  ELSE PLOTCODE = 'LIST'; SORTVAL = 0;
  SEQ = 1; VARNANE = 'CROPLAND'; EFVALUE = XW;
  XVAL1 =.; XVAL2 =.'XVAL2 =.'XVAL2 =.';
  XVAL3 =.; XVAL4 =.'XVAL4 =.'XVAL4 =.';
  XVAL5 =.; XVAL6 =.'XVAL6 =.'XVAL6 =.';
  XVAL7 =.; XVAL7 =.';
  SYSAHILO = 1; UNEXPNVL = CROPLAND;
  VALUE = CROPLAND * EFVALUE;
  OUTPUT SASOUT.CALC2;
END;
RETURN; ** END OF CHL1;

Note: Link code for tables 2 through 21 not shown.

CHL22 : ** CHART/HILO LINK FOR D2;
IF D2 GT 0 THEN DO;
  IF FRAME = 'AREA' THEN PLOTCODE = 'AREA';
  ELSE PLOTCODE = 'LIST'; SORTVAL = 0;
  SEQ = 22; VARNANE = 'D2'; EFVALUE = 1;
  XVAL1 =.; XVAL2 =.'XVAL2 =.'XVAL2 =.';
  XVAL3 = CROPLAND; XVAL3 =.'XVAL4 =.'XVAL4 =.';
  XVAL5 =.'XVAL5 =.'XVAL5 =.'XVAL5 =.';
  XVAL7 =.; XVAL7 =.';
  SYSAHILO = 1; UNEXPNVL = D2;
  VALUE = D2;
  OUTPUT SASOUT.CALC2;
END;
RETURN; ** END OF CHL22;

Note: Link code for tables 23 through 78 not shown.

NOTE: As you can see, the code remains very similar to that in Figures 2, 3, and 4. The links must be positioned physically within the data step instead of preceding it as the macros do.

FIGURE 6

EXAMPLE OF GENERATED CODE TO USE LINKS FROM FIGURE 5 (CHSELS):

LENGTH XAVAL1-XAVAL7 $ 1;
SELECT (STATE) ;
WHEN (1) 00;
  LINK CHL1 ; ** CROPLAND;
  LINK CHL2 ; ** CCRNXXPL;
  LINK CHL3 ; ** CS0YXXPL;
  LINK CHL4 ; ** CSRGXXPL;
  LINK CHL22 ; ** D2 ; LINK CHL23 ; ** D3;
  LINK CHL24 ; ** D4; LINK CHL26 ; ** D6;
  LINK CHL74 ; ** L2; LINK CHL75 ; ** L3;
  LINK CHL76 ; ** L4; LINK CHL78 ; ** NHOTGTOI;
END;

RETURN;

When (56) DO;
  LINK CHL1 ; ** CROPLAND;
  LINK CHL2 ; ** CCRNXXPL; LINK CHL9 ; ** CDEBXXPL;
  LINK CHL76 ; ** L4; LINK CHL78 ; ** NHOTGTOI;
END;
OTHERWISE;
END;
RETURN;

FIGURE 7

EXAMPLE OF SAS CODE INCORPORATING FIGURES 5 AND 6:

DATA SASOUT.CALC2(KEEP=&BREAKIDS &HILOEFS VARNAME VALUE SORTVAL SYSAHILO EFVALUE XAVAL1-SAVAL7 SEQ XVAL1-XVAL7 UNEXPNVL);
LENGTH VARNAME $ 8;
SET IN.SASMSTR;
IF REPTIN GT 0 AND CRITERR LT 1;
*** CALL IN MACRO TO CALCULATE EXPANSIONS ON DATA VALUES;
%EXPAND;
*** BRING IN GENERATED CODE TO CALL MACROS:
%INCLUDE CHSELS; *** CODE FROM FIGURE 7;
RETURN;
*** BRING IN GENERATED LINKS;
%INCLUDE CHLINKS; *** CODE FROM FIGURE 6;
**EXAMPLE OF PRINT GENERATED USING LINK OR MACRO CODE**

**MARCH 1992 AG SURVEY -- POTENTIAL OUTLIER PRINTS**

MINIMUM EXPANDED ACRES FOR CROPLAND

REPORTS THAT EXPAND TO ZERO HAVE BEEN EXCLUDED

---

**ATLANTIS (01)**

<table>
<thead>
<tr>
<th>PLOT CODE</th>
<th>STRATA CODE</th>
<th>SEGMENT CODE</th>
<th>OL/HOL EXPN CODE</th>
<th>TRACT FACT</th>
<th>FARM EXPN CODE</th>
<th>FARM FACT</th>
<th>CROPLAND EXPN CODE</th>
<th>CROPLAND FACT</th>
<th>CORN EXPN CODE</th>
<th>CORN FACT</th>
<th>EXPN CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA 3103</td>
<td>250</td>
<td></td>
<td>714.3 1.00</td>
<td></td>
<td>25</td>
<td>145</td>
<td></td>
<td>0</td>
<td>145</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AREA 1304</td>
<td>900</td>
<td></td>
<td>649.4 1.00</td>
<td>35</td>
<td>35</td>
<td>120</td>
<td></td>
<td>32</td>
<td>115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AREA 2001</td>
<td>918</td>
<td></td>
<td>1179.8 1.00</td>
<td>98</td>
<td>98</td>
<td>99</td>
<td></td>
<td>30</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AREA 1312</td>
<td>24</td>
<td></td>
<td>651.1 1.00</td>
<td>138</td>
<td>203</td>
<td>220</td>
<td></td>
<td>140</td>
<td>150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AREA 1303</td>
<td>1027</td>
<td></td>
<td>126.5 1.00</td>
<td>389</td>
<td>490</td>
<td>500</td>
<td></td>
<td>400</td>
<td>470</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MARCH 1992 AG SURVEY -- POTENTIAL OUTLIER PRINTS**

MINIMUM EXPANDED DIFFERENCE IN ACRES FOR CORN

MARCH VS. DECEMBER

DIFFERENCES THAT EXPAND TO ZERO HAVE BEEN EXCLUDED

---

**ATLANTIS (01)**

<table>
<thead>
<tr>
<th>PLOT CODE</th>
<th>STRATA CODE</th>
<th>SEGMENT CODE</th>
<th>TRACT FACT</th>
<th>FARM FACT</th>
<th>CROPLAND FACT</th>
<th>CORN FACT</th>
<th>EXPN CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA 1312</td>
<td>24</td>
<td>138</td>
<td>203</td>
<td>220</td>
<td>150</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>LIST 70</td>
<td>790529320</td>
<td></td>
<td>927</td>
<td>800</td>
<td>350</td>
<td>500</td>
<td>49.8</td>
</tr>
<tr>
<td>AREA 1308</td>
<td>9008</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>LIST 70</td>
<td>770139290</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIST 70</td>
<td>790392320</td>
<td></td>
<td>447</td>
<td>380</td>
<td>60</td>
<td>190</td>
<td>49.8</td>
</tr>
<tr>
<td>LIST 70</td>
<td>790482400</td>
<td></td>
<td>0</td>
<td>0</td>
<td>110</td>
<td>0</td>
<td>49.8</td>
</tr>
<tr>
<td>LIST 66</td>
<td>857594550</td>
<td></td>
<td>0</td>
<td>0</td>
<td>62</td>
<td>0</td>
<td>49.8</td>
</tr>
<tr>
<td>LIST 70</td>
<td>770334330</td>
<td></td>
<td>0</td>
<td>0</td>
<td>105</td>
<td>0</td>
<td>49.8</td>
</tr>
<tr>
<td>AREA 1303</td>
<td>1027</td>
<td>30</td>
<td>30</td>
<td>275</td>
<td>240</td>
<td>80</td>
<td>120</td>
</tr>
<tr>
<td>LIST 70</td>
<td>770302510</td>
<td></td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>49.8</td>
</tr>
<tr>
<td>LIST 66</td>
<td>770908860</td>
<td></td>
<td>197</td>
<td>140</td>
<td>75</td>
<td>20</td>
<td>87.9</td>
</tr>
<tr>
<td>LIST 70</td>
<td>770157340</td>
<td></td>
<td>423</td>
<td>363</td>
<td>160</td>
<td>250</td>
<td>49.8</td>
</tr>
<tr>
<td>AREA 2003</td>
<td>1147</td>
<td>147</td>
<td>276</td>
<td>218</td>
<td>50</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>LIST 71</td>
<td>711598810</td>
<td></td>
<td>0</td>
<td>0</td>
<td>120</td>
<td>0</td>
<td>27.0</td>
</tr>
<tr>
<td>AREA 1312</td>
<td>9072</td>
<td>10</td>
<td>10</td>
<td>40</td>
<td>38</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>LIST 66</td>
<td>790155560</td>
<td></td>
<td>70</td>
<td>30</td>
<td>35</td>
<td>0</td>
<td>87.9</td>
</tr>
<tr>
<td>LIST 66</td>
<td>790712950</td>
<td></td>
<td>130</td>
<td>128</td>
<td>65</td>
<td>100</td>
<td>87.9</td>
</tr>
<tr>
<td>AREA 2008</td>
<td>9128</td>
<td>7</td>
<td>115</td>
<td>133</td>
<td>125</td>
<td>60</td>
<td>20</td>
</tr>
<tr>
<td>AREA 1311</td>
<td>23</td>
<td>86</td>
<td>550</td>
<td>280</td>
<td>170</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>AREA 1303</td>
<td>1027</td>
<td>389</td>
<td>490</td>
<td>500</td>
<td>470</td>
<td>80</td>
<td>110</td>
</tr>
</tbody>
</table>