ABSTRACT

When a report has several thousands lines, it is almost impossible to review on-line. Even the PROC REPORT is a subject of this "Van Neuman Curse" - rigid sequentiality.

This paper explains how to build and implement an on-line data viewer, which allows you to see an overall summary of your data, then to "drill down" on selected fields to display detail lines that make up those statements. Therefore you can target specific areas of interest without having to view or scroll an entire report.

This is a two step procedure. First, for each subtotal level in the report a DATA member is built; i.e., each level is represented with its own DATA, all in the same LIBREF. The lowest level is a window to the INFFILE. Each observation on this level is actually a set of detail observations grouped together - as many as will fill up the screen leaving room for the header and bottom lines. By paging back and forth you can see the whole INFFILE.

The next level contains first subtotals only, grouped the same way. For each line on the screen showing the given observation a pointer to the first observation (page) of the lowest level can be calculated. By marking a line on the screen of the second level and pressing PF3 you will bring up the observation from the lowest level, showing all details summarized on the selected line. This process continues until the highest level of summation is presented by observations/screens.

Second, using SAS/AF, a set of menus, programs, screens, and macros is developed to handle this hierarchical data.

The design and programming of a simple sample report is demonstrated. The formal algorithm, which determines how to calculate pointers allowing a jump from the parent-line to the child-page, is also presented and useful macros, showing how to branch between SAS/AF programs (including branching to itself), are discussed.

INTRODUCTION

In many cases too much paper is being "lasered" in printers, especially in accounting-type applications. The problem partially lies in genetically built-in sequentiality in today's Business Data Processing. SAS Institute software resolves this precisely because of the sequential nature of the DATA step.

The following is a description of a method that allows replacement of a sequential report with the truly user-friendly, on-line hierarchical presentation of the same data.

To demonstrate this method, a simple sales report from Company IBN (International Business Novelties) will be built. Since this process has two distinct steps - the first to build SAS DATA SETS representing the Data Base, and the second, to build a set of SAS/AF screens, SAS/AF programs and macros - these steps are discussed separately.

IBN has operations in four regions - North, East, South and West. Each region covers a group of states. Every state has a number of salesmen who sell Widgets and Gadgets. This structure can be as deep as desired, but for this demonstration, three levels are quite sufficient.

It is evident how the standard sales report would look. The proposed solution would provide a different view.

ABSTRACT

When a report has several thousands lines, it is almost impossible to review on-line. Even the PROC REPORT is a subject of this "Van Neuman Curse" - rigid sequentiality.

This paper explains how to build and implement an on-line data viewer, which allows you to see an overall summary of your data, then to "drill down" on selected fields to display detail lines that make up those statements. Therefore you can target specific areas of interest without having to view or scroll an entire report.

This is a two step procedure. First, for each subtotal level in the report a DATA member is built; i.e., each level is represented with its own DATA, all in the same LIBREF. The lowest level is a window to the INFFILE. Each observation on this level is actually a set of detail observations grouped together - as many as will fill up the screen leaving room for the header and bottom lines. By paging back and forth you can see the whole INFFILE.

The next level contains first subtotals only, grouped the same way. For each line on the screen showing the given observation a pointer to the first observation (page) of the lowest level can be calculated. By marking a line on the screen of the second level and pressing PF3 you will bring up the observation from the lowest level, showing all details summarized on the selected line. This process continues until the highest level of summation is presented by observations/screens.

Second, using SAS/AF, a set of menus, programs, screens, and macros is developed to handle this hierarchical data.

The design and programming of a simple sample report is demonstrated. The formal algorithm, which determines how to calculate pointers allowing a jump from the parent-line to the child-page, is also presented and useful macros, showing how to branch between SAS/AF programs (including branching to itself), are discussed.

INTRODUCTION

In many cases too much paper is being "lasered" in printers, especially in accounting-type applications. The problem partially lies in genetically built-in sequentiality in today's Business Data Processing. SAS Institute software resolves this precisely because of the sequential nature of the DATA step.

The following is a description of a method that allows replacement of a sequential report with the truly user-friendly, on-line hierarchical presentation of the same data...

To demonstrate this method, a simple sales report from Company IBN (International Business Novelties) will be built. Since this process has two distinct steps - the first to build SAS DATA SETS representing the Data Base, and the second, to build a set of SAS/AF screens, SAS/AF programs and macros - these steps are discussed separately.

IBN has operations in four regions - North, East, South and West. Each region covers a group of states. Every state has a number of salesmen who sell Widgets and Gadgets. This structure can be as deep as desired, but for this demonstration, three levels are quite sufficient.

It is evident how the standard sales report would look. The proposed solution would provide a different view.
Again PF7 and PF8 are for browse; however there is no selection to be made, pressing PF3 will return you to the previous screen, as it would do on any level if no selection was made. Of course, all SAS AF hot keys and commands retain their functions and can be effectively used by the experienced user.

**DATA BASE CREATION**

Let us dissect the data organization for this report.

One LIBREF is assigned for the storage. It has 3 DATA members - as many as there are levels in our on-line report presentation. For the sake of clarity, certain assumptions were made as follows:

The number of states per region is arbitrarily limited to 15. 6 states per screen are allowed and the number of distributors in one state may be up to 33. Further assume that the user wishes to see no more than 6 distributors per screen. Under those assumptions, level 1 - major totals - will have only one page, or screen. You simply combine all major totals and a grand total in one observation. The problem is to find a pointer to the first observation on the next level 2 when an X is placed in the FLAG. The relations between observations on different levels for this example are shown on FIG 1.

For each line on level 1 there will be 3 observations on level 2. Two levels are allowed on level 2; the number of observations on level 2 is 1, 2, 3, 4.

### LEVEL 1 DISTRIBUTION

- **EAST**
- **WEST**
- **NORTH**
- **RIO**

with 6 states and the third with 3 states (total of 15) and one line for the region totals. Therefore the pointer from lines 1,2,3,4 on level 1 will be 1,4,7 and 10 respectively.

For each line on level 2 there are 6 observations on level 3, five of them full and the last one will have 3 distributors and a state total. Your selection of NORTH REGION - NORTH DAKOTA puts you at line 3 on level 1, then at line 2 on level 2, which results in the observation number on level 3:

\[(3-1)\times 15+(2-1)\times 6+1=187\]

This is the first observation you will FEDIT or rather FS Browse, since it is the last level and no more selections available. The general formula to calculate pointers is described in the next paragraph.

**ALGORITHM**

The variables used in this paragraph are:

- \(L_{\text{max}}\) - the number of levels, or summary breaks.
- \(i\) - current level number on which a selection was made, \(i=1,2,...,L_{\text{max}}-1\).
- \(O\) - number of observations on level \(l\) related to one line on level \(l=1,2,3,...,L_{\text{max}}\).
- \(D\) - number of detailed lines within those \(O\) observations, which were summarized into one line on level \(i-1\), \(i=2,3,...,L_{\text{max}}\).
- \(N\) - standard SAS notation for the current observation.
- \(L\) - line number which was X-ed on page (observation) \(N\) on level \(i\).

The formula for a pointer OBNO, to the next level is:

\[
\text{OBNO}_{i+1} = (\text{CEIL}(\frac{N}{O})) \times (1 \times O + (L-1) \times D) + 1.
\]

For any particular application variables \(O\) and \(D\) are constant and may be stored in the DATAPARM file. In the case of IBM they are:

- \(O=3\)
- \(D=15\)
- \(O=6\)
- \(D=33\).

Application of this formula to the case of IBM yields 7 and 187 respectively.

There is a hidden trade off, however. Assuming that \(O\) is a constant for a given level, it may create some space problems. If this problem is lethal, then pointers must be calculated during the data base creation, making the process much more complicated. The author faced this problem only once in numerous applications.

**SAS/AF PROGRAMMING NOTES**

The purpose of this article is to present a method, not to give a complete programming solution. There are too many different ways to build the DATA BASE discussed above. Usage of SAS arrays proved to be very beneficial in numerous applications, even the task of defining all variables on the SAS/AF screens was not so tedious because of the columnar structure of the screens. The SAS/AF technique however requires demonstration.

Your LIBREF is CAT for the SAS CATALOG and DATA for the SAS DATA. If there are more than one user then the SAS SHARE Data Server must be used. The name of the catalog is IBN.

As it customary in SAS/AF applications the highest level of the on-line program is MAIN.MENU. All other Data Names are either self-explanatory or commented.

The program below is constructed for two levels of data.

```
/* IBN SALES REPORTING AF PROGRAM */

%GLOBAL OBNO=0;
/* it is Q variable in (1). */

DATA LEVEL1; SET DAT.LEVEL1; RUN;
DATA LEVEL2; SET DAT.LEVEL2; RUN;
/* Working copies of the levels 1 and 2. They are necessary because otherwise an X which you put in the selection field would appear 'uninvited' at the next session. */

PROC FSEDIT DATA=LEVEL1 SCREEN=DAT.IBN.LEVEL1.SCREEN;
/* You made a selection above, or you did not. The next block will figure it out. */

DATA_NULL; SET LEVEL1;
ARRAY FLAGS [4] $1 FLAG1-FLAG4;
DO L=1 TO 4;
IF FLAGS (L) = 'X' THEN DO;
   Q = (L-1)\times 3 + 1;
END;
```

---

229
CALL SYMPUT('OBNO',Q);
END;
END;
RUN;

%M1: /*Macro M1 controls the flow between levels 1 and 2.
It is expanded here for easier reading*/
%MACROM1:
%IF %EVAL(&OBNO=0) %THEN %DO;
/* NO selection, return to menu*/
PROC DISPLAY C=CAT.IBN.MENU;
RUN;
%END;
%ELSE %DO;
/* Line X-ed, proceed with the LEVEL 2 */
PROC FSEDIT DATA=DAT.LEVEL2
OBS=&OBNO
SCREEN=CAT.IBN.LEVEL2.SCREEN;
RUN;
/* Reset the OBNO for the next level*/
%LET OBNO=0;
%END;
%MENDM1;

/* You would come to this point if selection was made
on level 1. Now you check if anything was marked on
level 2. Formula (1) is used below in the calculation of Q with actual
parameters. Since in the IBN example L = 2, this formula
yields 187 */

DATA _NULL_; SET LEVEL2;
ARRAY FLAGS (6) $1 FLAG1-FLAG6;
DO L=1 TO 6;
   IF FLAGS {L} NE ' ' THEN DO;
      Q = (CEIL(7/2)-1)x15 + (L-1)x6 + 1;
      CALL SYMPUT('OBNO',Q);
   END;
END;
RUN;

%M2: /*Macro M2 controls the flow after level 2.
It is expanded here for the easier reading*/
%MACROM2:
%IF %EVAL(&OBNO=0) %THEN %DO;
/* NO selection, CALL THE PROGRAM AGAIN*/
PROC DISPLAY C=CAT.IBN.SALES.PROGRAM;
RUN;
%END;
%ELSE %DO;
/* Line X-ed, proceed with the LEVEL 3,
note that FSXWROKE used instead of FSEDIT */
PROC FSXWROKE DATA=DAT.LEVEL3
OBS=&OBNO
SCREEN=CAT.IBN.LEVEL3.SCREEN;
RUN;
/* And return to the same program again*/
PROC DISPLAY C=CAT.IBN.SALES.PROGRAM;
RUN;
%END;
%MENDM2;

Conclusion

It seems that because of the high level of formalization of the proposed
method it could be developed with substantial but not overwhelming
efforts to the PROC level with a lot of advantages for the mainframe
programmers.

Acknowledgements

The author would like to express his gratitude to Fredrica Lake of
TRANSCO whose help in the debugging of severe cases was instrumental.
SAS, SAS/AF,SAS/SHARE are registered trademarks of SAS Institute Inc., in the USA and other countries. © indicates USA registrat

For more information please contact Alex Ostrovsky, TRANSCO
ENERGY COMPANY, 2800 Post Oak Boulevard, Houston, TX
77055-6106. Tel (713)-439-4267.