FSLETTER FOR A VERY DETAILED REPORT

Kimberly J. LeBouton, Lexus, Torrance, CA

ABSTRACT

When I was given a new assignment for a very detailed report that was basically seven reports on one page, I was determined not to use "PUT" statements. After some research which ruled out PROC REPORT, I decided that PROC FSLETTER would solve my problems. This paper will serve as a guide to using PROC FSLETTER: preparing your data to be "one" detailed record; setting up a catalog to save, modify, and print your report; and using SAS formats to get around some of the quirks of PROC FSLETTER.

INTRODUCTION

Before you start with PROC FSLETTER, you need to have some basic interactive SAS skills in place. A working knowledge of SAS Display Manager is vital to create the report layout. This step cannot be done in batch or a non-windowing interactive mode. You will also need to be able to create and modify permanent SAS catalogs. And finally, some SAS/FSP or SAS/AF experience is helpful, but not necessary.

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Pos</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>COUNT</td>
<td>Num</td>
<td>8</td>
<td>22</td>
<td>Frequency of observations in category</td>
</tr>
<tr>
<td>2</td>
<td>SEATBELT</td>
<td>Char</td>
<td>3</td>
<td>7</td>
<td>Seatbelt usage</td>
</tr>
<tr>
<td>3</td>
<td>SPEED</td>
<td>Char</td>
<td>5</td>
<td>10</td>
<td>Speed of car prior to impact</td>
</tr>
<tr>
<td>4</td>
<td>STATUS</td>
<td>Char</td>
<td>7</td>
<td>15</td>
<td>Driver injury status</td>
</tr>
</tbody>
</table>

Figure 1 Contents of data set

PROC FSLETTER works well if you have a very detailed individual record (e.g., an individual survey) or a report that has information spread out all over the page. This paper will provide a cookbook approach to PROC FSLETTER using the latter scenario.

Let's begin with describing the data and a mockup of the report. I used the SASUSER.HIGHWAY data set provided with SAS/ASSIST. Figure 1 shows the information from PROC CONTENTS describing the data set.

Figure 2 Report Mock-up

Figure 2 is the "mock-up" of the report. Since I am generally the recipient of someone else's report design versus being the developer of the report, I like to have an ASCII file of the mock-up but the report could be entirely "mocked-up" in the FSLETTER window.

SAS PROGRAM

The sample SAS program used, for this example, is displayed in Figure 3. Three separate PROCs are used to develop the report. Once the report is created and tested, only the second PROC FSLETTER needs to run. Each step will be described below.

```sas
/* ** FILE ALLOCATIONS ***/
LIBNAME DETAIL 'TOC48K\SAS\FSLETTER\SASDATA';
FILENAME PRTRFILE 'TOC48K\SAS\FSLETTER\PRTRFILE';
/* ** USE THE FOLLOWING TO CHANGE THE REPORT LAYOUT ***/
PROC FSLETTER LETTER=DETAIL,REPORTS,EXAMPLE,LETTER;
RUN;
/* ** USE THE FOLLOWING TO WRITE THE REPORT ***/
PROC FSLETTER DATA=SASUSER.HIGHWAY
LETTER=DETAIL,REPORTS,EXAMPLE,LETTER
PRINT=PRTRFILE;
RUN;
/* ** USE THE FOLLOWING IN DISPLAY MANAGER TO VIEW OUTPUT ***/
PROC FSLETTER FILEREF=PRTRFILE;
RUN;
```

Figure 3 SAS Program

For this example, one SAS library and one filename is allocated. The library "DETAIL" is a SAS catalog that will contain the report layout and the FORM attached to the report (See below for discussion of FORM). PRTRFILE is an output file where I like to have my reports printed.

The first PROC FSLETTER has only "LETTER=" specified. This PROC will take us to the screen displayed in Figure 4a. A FORM must be specified at this point, as seen on the command line. This report will use a form called "V90" which is a vertical page with 90 characters per line and 80 lines per page. This form does not designate a printer. (See FORMNAME or FSFORM in the appropriate SAS Companion for your operating system. Most of my FORMs are in my SASUSER.PROFILE while running under MVS.) Once the form is entered, the margins will be adjusted to the above specifications.

Figure 4a Form Specification

```
FSLETTER: EXAMPLE.LETTER (E)
Command ===> FORM V90
00001
00002
```
The next step is to bring in a "mock-up" with an INCLUDE statement or to design the report while in this screen. One possible problem of bringing in a report is the file, used to create the mock-up, may not match the above 90 character specification. That is, if you created a report layout with a file allowing for 132 characters per line, there will be some wrapping of text or a lot of blank lines when it is brought into the FSLETTER screen.

![Table](image)

**Figure 4b Variable Specification**

Once the mock-up is in place as shown in Figure 4b. Variable names must be assigned to variable fields. Variables start with an ampersand ( & ), and is located in front of the proper SAS variable name. The variable name cannot exceed the length of the variable shown on the screen. Alias names will have to be assigned to variables when the names are longer than the variable length. Variables that are longer than the name will end with underscores ( _ ) until proper length is reached. Variables must have a space before and after another variable or text. See DATA MANIPULATION section to get around this space limitation.

The final step of this report layout is to assign alias names to variable names which are too long for the report. Enter ATTR on the command line and the ATTRIBUTE screen shown in Figure 4c will appear. This ATTRIBUTE screen is similar to the ATTRIBUTE screens in either FSEDIT or AF. The options are limited primarily to letter writing. Enter the appropriate variable name next to the alias name.

![Screen](image)

**Figure 4c Attribute Screen**

The second PROC FSLETTER brings in the data and runs the report. The output is sent to PRTFILE. Nothing appears on the screen due to the third PROC statement included in the program. PROC FSLIST will display the PRTFILE to the screen.

Once the report is running correctly, you will probably want to comment out the first PROC FSLETTER. This only needs to be run if the report needs to be modified. Speaking of modifications, PROC FSLETTER is much easier to modify than PUT statements. With CUT and PASTE capabilities available in DISPLAY MANAGER (limited in the mainframe world, but not too bad), you can move blocks of text all over the place. Modifying reports are not as burdensome as having to redo all of your PUT statements in the event of a major report modification.

PROC FSLIST could also be commented out once the report is in place. You may direct the output to a printer instead of a print file.

### DATA MANIPULATION

All of the data manipulation must occur before you use FSLETTER. As mentioned in the introduction, FSLETTER works very well on reports that are using data sets with very detailed records. If your report has a column look to it, you will want to consider other PROCs or even use PUT statements with DATA step programming.

In order to create "one" detailed record, you can use many different SAS programming techniques. Two examples are listed below to help achieve the desired detailed record data set.

#### Read a "flat" file as one record

```sas
data detail;
  input q1 $ x1/
  q2 $ x2/ q3 $ x3;
datalines;
  1 99.8
  2 97.3
  3 95.4
run;
```

#### Use arrays on SAS data sets

```sas
data detail;
  input q $ x;
datalines;
  1 99.8
  2 97.3
  3 95.4
run;
```

```sas
data one_rec(keep=qquest1-qquest3 score1-score3);
array quest(3) $ qquest1-qquest3;
array score(3) score1-score3;
do i=1 to 3;
  set detail;
  quest(i)=q; score(i)=x;
  end;
runch
```

PROC TRANSPOSE is also a very good option in turning many records into one detailed record.

To get around "space" limitations, while in the FSLETTER editor, the PICTURE option with PROC FORMAT is invaluable. The following PROC FORMAT step allowed me to fit in a perfect index score into a tight space and also close a parenthesis right next to some accompanying text.
CONCLUSION

When I first presented this paper, the only written information I could find recommending the use of FSLETTER as a report writing tool was one sentence in the FSLETTER section of the FSP manual: "... Applications are not limited to letters. You can use the same techniques to create personalized questionnaires, reports, and other documents." The remainder of the chapter talks about using FSLETTER as a letter writing tool. Of course, I thought my approach to FSLETTER was "state-of-the-art" thinking until I received an edition of Observations, it contained a two page discussion of using FSLETTER as a report writing tool. With my paper and the Observation article, you should have enough background to get started.

Always keep in mind that using FSLETTER as a report writing tool works best for very detailed records. If your report has a tabular look to it, you'll want to use other SAS procedures or even the "dreaded" PUT statements.

FOOTNOTES


2FSLETTER is discussed in Observations, Third Quarter, 1994., pp. 64-65.

ACKNOWLEDGEMENTS

I wish to thank Lorraine Geesey and Al LeBouton for reviewing this paper.

COPYRIGHT INFORMATION

SAS, SAS/AF, SAS/FSP, and Observations are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. MVS is a registered trademark or trademark of International Business Machines. * indicates USA registration.

AUTHOR

Kim LeBouton
3431 Yellowtail Drive
Rosswoor, CA  90720

e-mail: 0007426353@mcmail.com