ABSTRACT

PROC FSEDIT and FSVIEW are procedures in SAS/FSP® software which allow for interactive, real-time maintenance and modification of SAS data sets on a record level. This workshop will allow attendees to perform simple modifications of a SAS data set, including addition, update and deletion of observations. Manipulation of the default edit screen with display screen modification and use of field attributes and general screen parameters will be performed to develop application-specific screens. While screen control language (SCL) provides additional power to application development, this portion of FSEDIT will not be covered.

INTRODUCTION

This beginning level workshop on SAS/FSP introduces basic concepts of creating and exploring SAS data sets using the interactive procedures FSEDIT and FSVIEW in SAS version 6.10 for Windows. Similarities and differences between the two procedures will be highlighted.

PROC FSEDIT allows interactive data set creation, editing, and exploration one observation at a time.

PROC FSVIEW displays observations in a tabular format instead.

As you will see, these two procedures are very similar.

Adding new transactions to a data set or querying existing observations are easiest using PROC FSEDIT. The FSVIEW procedure is most useful for viewing the data set structure and contents, for example after a merge or after calculating new variables to see if the desired result was in fact obtained.

In the first half of the workshop, you will create two SAS data sets using PROC FSEDIT and PROC FSVIEW. You will learn how to:

> invoke the procedures and create data sets
> display labels in place of variable names
> specify attributes for variables
> add, duplicate and delete observations
> scroll within and between observations
> edit observations

The second portion of the workshop will include subsetting, searching and sorting before looking at basic screen customization in PROC FSEDIT. Lastly you will learn how to use formulas to modify existing variables and create new variables in an FSVIEW session.

INVOKING PROCEDURES

FSEDIT and FSVIEW can be invoked in several different ways: from the command line, using the menu bar, and by code submission. In this workshop we will submit program statements to invoke the sessions.

DATA SET CREATION

Syntax:

PROC FSEDIT | FSVIEW data=SASdataset
new=SASdataset like=SASdataset
<label>;

FSEDIT and FSVIEW both use the NEW= option to name the SAS data set to be created. The LIKE= option creates a new data set with the same variable attributes as an existing data set, but with no observations. You enter the session in the data set definition mode (New window) and can specify the
names and attributes of the variables which you want to be present. Optionally you can specify variable labels, formats and informats. After completing the variable attribute definitions, the default data entry window is displayed.

Before you can create your first permanent data set, you need to submit a LIBNAME statement to assign a libref to the SAS data library where the data set will be stored.

```sas
LIBNAME workshop 'C:\TUTORIAL\S8l7';
```

Use the libref as the first part of the two-level name of the data set specified in the PROC FSEDIT step.

**DATA SET CREATION IN PROC FSEDIT**

```sas
proc fsedit label new-workshop.first;
run;
```

The example above creates a new data set named FIRST to be stored in the SAS data library you assigned the libref WORKSHOP. Including the LABEL option in the PROC FSEDIT statement displays any labels you associate with variables in the new data set. When you submit this code, the FSEDIT New window is displayed.

```
+-------------FSEDIT New WORKSHOP.FIRST-----------+
I
Name Type Length Label Format
NAME $ 15 Subject name
WEIGHT N 8 Weight in pounds 3._
```

The New window is where you specify variable attributes. Each line in the window represents information about a single variable.

All SAS variables must have the following three attributes:

1. a name,
2. a type (character or numeric), and
3. a length.

i) Variable names must be valid SAS names (unique, with 1 to 8 characters, beginning with a letter or an underscore). If you enter an invalid variable name, you will receive an error message. If you enter a ? you will be shown a help screen.

ii) A variable's type can be character (denoted by $ or C) or numeric (denoted by N or a blank).

iii) The length of a character variable such as NAME can be up to 200 characters (the default length is 8). The length you specify for a character variable determines the width of the data entry field, so choose a length that accommodates the longest possible value that you will enter. Generally, you can use the default length of 8 for numeric variables such as the variable WEIGHT. By default, the field is 12 characters wide.

You can assign the following optional attributes to your new variables: a label to be displayed in place of the variable name, a format, and an informat. Because values for WEIGHT will not exceed 3 digits, the 3. format can be assigned. To select the format from a HELP:Formats screen, type ? in the Format field and press ENTER. A different help screen will be displayed for character variables. Select Locals->Format/Informat or use left and right function keys to toggle the display from the Format field to the Informats field. Corresponding HELP:Informats screens are also available by entering ?. Close the FSEDIT NEW window using View>End

Important! Check the attributes carefully before you exit the FSEDIT New window. Once you leave this window, you cannot return to it.

**DATA SET CREATION IN PROC FSVIEW**

Now you'll learn how to create a new data set with the same structure as an existing data set using FSVIEW with the NEW= and LIKE= options. Submit a PROC FSVIEW statement, very similar to the previous PROC FSEDIT statement.

```sas
proc fsview new=workshop.second like=workshop.first;
run;
```

When you submit the program, the FSVIEW New window appears for the new data set SECOND. It is already initialized with the variable attributes from the old data set FIRST. You can choose to modify these attributes or leave them the same. Check the attributes carefully and close the New window by selecting View>End.

**DATA ENTRY**

During data entry you can add, duplicate and delete observations.

**DATA ENTRY IN PROC FSEDIT**

When you close the New window the default data entry screen is displayed with underlines showing the missing values. The new data set has 0 observations.

Select Edit>Add new record to add a new record.
A new observation is created and displayed, and the cursor moves to the first field. You can now enter data values. Character values are left-aligned, and unformatted numeric values are right-aligned in a field twelve characters wide. By default, all character values are displayed in uppercase as you enter them into a new data set. To continue adding observations, simply repeat the process. When you add another new observation, it is placed at the end of the data set and is automatically displayed.

Another way to create a new observation is to duplicate an observation. Duplicating copies the current observation to the end of the data set. Duplicating is useful when the new observation has several values in common with the original one. Simply type over any old values you want to modify.

You can also delete the current observation from the data set. When you mark this observation for deletion the fields are erased and you receive a message telling you that the observation has been deleted. If you scroll forward and backward through the data, you see that the observation is deleted from the display, but that the observations are not renumbered. The observation is not physically removed from the data set, even though you no longer have access to it. Marked observations are not physically removed from the data set until you re-create the data set. For example, you can remove deleted observations by running a DATA step or a PROC SORT step against the data set.

To delete an observation, select Edit > Delete. This opens the Delete dialog box. Type the observation number to delete, and select OK. To select more than one observation for deletion, separate each observation number with a space, or type a range such as 23-28. Just as you saw in PROC FSEDIT the observation is removed from the display but is not deleted from the data set. The observations are not renumbered until the data is sorted or a data step is invoked.

To end the FSVIEW session select File > End

SCROLLING AND EDITING

The FSEDIT and FSVIEW procedures each offer many additional features to simplify data entry, query, editing, validation, display, and retrieval. Before you start editing valuable data, you should be aware that editing is done in REAL TIME. If you are working with an irreplaceable data set, making a backup copy is strongly advised!

INVOKING FROM THE COMMAND LINE

FSEDIT and FSVIEW sessions for data set editing or viewing may be invoked from the command line by the following syntax:

FSEDIT <? |dataset<screen-name>> or
FSVIEW <? |dataset<formula-name>>

If you do not specify a data set, the last data set used in the current SAS session will be used. If there were no data sets used in the session, a selection list screen will display a list of available data sets. The ? option will display the selection list also. In PROC FSEDIT the SCREEN-NAME option is used to specify a customized screen. In PROC FSVIEW the FORMULA-NAME option will cause a computed variable(s) to be displayed with the data set.
EDITING DATA IN PROC FSEDIT

Write and submit a PROC FSEDIT statement specifying the DATA= option to name the data set to be edited. This example invokes the FSEDIT procedure to edit the data set THIRD stored in the SAS data library to which you assigned the libref WORKSHOP. The THIRD data set is a completed version of FIRST and SECOND. Again, specify the LABEL option so that any labels associated with variables in the data set are displayed.

```
00001 proc fsedit data=workshop.third label;
00002 run;
```

When the FSEDIT window appears displaying the first observation of your SAS data set notice the observation number in the title bar of the window. Also notice the justification of the numeric and character variable values.

To modify a value, simply tab to the field, type over the value, and press ENTER. Make sure the Insert key is toggled off if the computer beeps at you.

SCROLLING THE DISPLAY IN PROC FSEDIT

In the FSEDIT procedure, the following menu bar selections enable you to scroll between and within observations in the data set.

```
Selection Action
View | scrolls forward to the next observation.
View | scrolls backward to the previous observation.
View | scrolls right to the next screen of the current observation.
View | scrolls left to the previous screen of the current observation.
```

Note: The examples and exercises in this workshop do not use data sets that require more than one screen. You can also use Page Up/Down keys or function keys for left, right, forward and backward to scroll.

EDITING DATA IN PROC FSVIEW

Sometimes it is helpful to get an overall view of the data set and study relationships between observations. The FSVIEW procedure gives you the ability to edit data in the form of a table of rows and columns.

To invoke PROC FSVIEW, submit a PROC FSVIEW statement, very similar to the PROC FSEDIT statement. The LABEL option is not used for the FSVIEW procedure.

```
proc fsview data=workshop.third;
run;
```

When you submit the PROC FSVIEW statement, the FSVIEW window appears, as shown here. Note that only actual variable names, not labels, can be displayed in the FSVIEW window. Labels can be seen by using the HELP key. The (B) displayed in the title bar indicates that the data set is open for browsing.

```
 KLEPP 02/18/17 169 1 1
 COLETTE 06/02/06 179 1 2
 GALVIN 02/14/00 186 2 3
```

The FSVIEW window opens in browse mode by default. Type MODIFY in the command bar or Select Edit>Update and click on OK to change to edit mode with record level locking. RECORD level locking enables you to edit one observation at a time. Lock the observation you want to edit by placing the cursor on it and pressing ENTER, or clicking with a mouse. The observation is highlighted to indicate that it is locked. At this point, you can modify values for the locked observation.

SCROLLING THE DISPLAY IN PROC FSVIEW

The FSVIEW window displays as many observations and variables at a time as it can fit in the display. To view data that extend beyond the display area, you can scroll forward, backward, right, or left using scroll bars, Page Up/Down or function keys. You can scroll to an observation by entering its number, to a specific variable column by entering a "= VAR_NAME =" command or scroll to the "TOP" or "BOTTOM" by entering those commands.

****** end of first part ******
In the second part you will learn commands in both procedures for tasks such as:
- subsetting observations interactively
- querying and sorting
- customizing default edit screens
- arranging variables
- modifying existing variables
- creating new variables

SUBSETTING, SEARCHING AND SORTING

SUBSETTING
You may already know how to select observations in other SAS procedures with a permanent WHERE condition using the WHERE statement. Both FSEDIT and FSVIEW procedures enable you to subset the data interactively using temporary WHERE conditions. You can use a single WHERE clause or augment it with additional WHERE ALSO... clauses during the FSEDIT or FSVIEW session to create temporary subsets of your data. Using WHERE clauses to create a subset can save you considerable time compared to scrolling through every observation in a data set.

Select Search>Where... to open the WHERE dialog box. The WHERE clause can use any valid SAS expression as the where-expression to specify a condition for selection. WHERE clauses must specify the actual variable name. If labels appear in place of variable names, you can determine the variable name using the HELP command assigned to a function key. Note that you do not include the WHERE keyword.

\[ \text{e.g.}> \ \\
\text{weight}>170 \ \\
\text{race}='6' \ \\
\text{weight}>165 \text{ and } \text{race}='2' \ \\
\text{name contains } 'EPpt' \ \\
\]

As shown in these examples, a where-expression typically specifies a variable, an operator, and a value for comparison. As in the permanent WHERE statement, character values must be enclosed in quotes and must appear in the same case as in the data set.

The following menu bar selections enable you to add and remove WHERE clause conditions interactively: Search>Where also... ; Search>Undo last where; Search>Where... (do not type a WHERE clause, select OK... clears all WHERE conditions at once)

The word Where... in the title bar of the window indicates that a WHERE condition is in effect.

SEARCHING
PROC FSEDIT (but not PROC FSVIEW) supports field value searches using commands such as find, locate and search.

\[ \text{example}> \ \\
\text{Find weight}=155 \ \\
\text{find name}= 'JONES' \ \\
\]

However, because the WHERE clause is so powerful and is applicable to both FSVIEW and FSEDIT we will not examine these other commands further in this workshop.

SORTING
PROC FSVIEW (but not PROC FSEDIT) allows you to sort the data set in place from within the procedure by one or more variables.

Select Edit>Sort... to open the Sort window. The physical data set is sorted. The sort will not proceed if a WHERE clause is in effect or if the dataset is indexed.

SCREEN CUSTOMIZATION IN SAS/FSP

SCREEN MODIFICATION IN PROC FSEDIT

In addition to selecting and reordering variables with a VAR statement, PROC FSEDIT allows modification of the default data entry screen to create a custom data entry display to match the layout of a questionnaire or other data collection form. Modifications to the screen should be saved in a screen catalog otherwise they will be lost when the session is ended.

1. Submit:
   \[ \text{proc fsedit new=workshop.third} \ \\
   \text{screen=workshop.demoscr;} \ \\
   \text{run;} \ \\
   \]
2. Select Locals>Modify screen.
3. Click on OK to close password window.
4. The Main Menu with 5 choices will appear
5. Select Screen Modification and Field Identification option 2. A screen similar to the default data entry screen will appear.
6. Select Edit>Options>Numbers to turn on numbers and tabs by typing tabs6 over line number 00001. Select Edit>undo to remove the tab markers from the screen (tabs remain set).
7. Use the text editor to customize the screen. Each field must have a space before and after it. Use underscores to create data entry fields. Delete text that is not wanted.
8. To exit and save select File>End.
9. When asked if you created any computed fields, respond N. These are created using SCL.
10. When prompted to identify a field by variable name place the cursor anywhere on that field and
press Enter or Unwanted. When you are finished the message "All fields are identified." will appear.
11. Select View>End to return to the Modification Mode Menu.
12. Select option 4, Assign Special Attributes to Fields, for initial values, limits, required fields etc. Use color and blinking attributes for erroneous data.
13. Select View>End to return to the Modification Menu.
14. Select option 5, Modification of General Parameters, to set screen size, color and override options and set defaults for text searches.
15. Return with View>End to the Menu and close this screen to enter data entry mode.

Your customized FSEDIT data entry display will appear on your screen.

+------------FSEDIT WORKSHOP.THIRD Obs 1------------+
| SUBJECT KLEPP     ID 1 |
| BIRTHDATE 02/18/17  RACE 1 = WHITE |
|                     2=BLACK |
| WEIGHT 169 POUNDS 3=OTHER |
+---------------------------------------------------+

ARRANGING VARIABLES IN PROC FSVIEW
In addition to selecting and reordering variables with a VAR statement, PROC FSVIEW enables you to interactively drop and show variables and to display selected variables in the FSVIEW window as ID variables that do not scroll. Dropping variables affects only the display, not the actual data set.

CREATING NEW VARIABLES IN PROC FSVIEW
A powerful new feature of PROC FSVIEW, which does not exist in PROC FSEDIT, is the ability to use formulas interactively to create new variables and modify existing variables during the FSVIEW session without using SCL. The formulas are stored in a catalog.

1. Submit:
   proc fsview data=workshop.third;
   run;
2. Select Locals> Define formula to open the Define command screen.
3. Type the name of an existing variable to modify its values or specify a new variable whose values are the result of the calculations. (The new variable is added only to the display, not to the data set.) Press Enter
   4. Select the variable type.
5. Optionally specify a format, an informat, and a label.
6. Enter the formula in the four lines available.
7. Select Compile to check the syntax of the formula.
8. Errors are displayed by selecting Window>Message
9. To save the formula select File>Save as>Save formula as... and specify a name such as LB_KG for the formula. The formula is saved by default in the SASUSER PROFILE catalog.
10. Review the formulas created by selecting Locals> Review formula, this selection opens the FSVIEW REVIEW window, displaying any formulas used during your FSVIEW session.
11. To open an FSVIEW session using the saved formula submit:
    proc fsview data=workshop.third;
    formula=sasuser.profile.lb_kg;
    run;

As shown in the example below, the FSVIEW Define command screen supplies the variable name and the equal sign that begin the definition.

+-----FSVIEW: Define command-------------------------+
  Name: KGWGT
  Type: NUMERIC CHARACTER
  Informat: 6.
  Label: WEIGHT IN KILOGRAMS
  Enter the formula below:
  KGWGT = weight / 2.2

You can specify one or more statements (as many as will fit on the four lines), separating multiple statements with semicolons. This formula converts PROC FSVIEW character input to uppercase.

+-----FSVIEW: Define command-------------------------+
  Name: UP
  Type: NUMERIC CHARACTER
  Format: 8.
  Informat: 8.
  Label: Changes FSVIEW entries to uppercase
  Enter the formula below:
  UP = "; array upp [*] name sex city;
  do i=1 to 3;
    upp[i] = upcase(upp[i]);
  end;

The formula you created is automatically applied to the data set. The new variable appears at the far right of the display. To save the computational fields as
part of the data set, you cannot overwrite the current file. Select File>New>Create... and select the desired variables from the VARLIST window displayed.

APPLYING EXISTING FORMULAS IN FSVIEW
In later FSVIEW sessions, you may want to reuse a stored formula. To perform the calculations specified in an existing formula, you can read in a formula during the session by selecting File>Open>Read a formula... To read in a formula simply specify its location. When you select OK the previously stored formula is applied.

CONCLUSION
The SAS/FSP full-screen interactive information processing procedures FSEDIT and FSVIEW are indispensable tools for creating, editing and efficiently querying and managing your data.

BIBLIOGRAPHY
Additional guidance may be found in the following excellent materials, some of which served as sources for the material in this workshop.


ACKNOWLEDGEMENTS
SAS and SAS/FSP are registered trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

CONTACT
Marge Scerbo mscerbo@umab.umb.edu
Alan C. Wilson awilson@umdnj.edu