



CHAPTER

7

The FSVIEW Procedure

<i>Overview</i>	93
<i>FSVIEW Procedure Syntax</i>	93
<i>PROC FSVIEW Statement</i>	94
<i>FORMAT Statement</i>	97
<i>ID Statement</i>	98
<i>INFORMAT Statement</i>	98
<i>VAR Statement</i>	100
<i>WHERE Statement</i>	100
<i>FSVIEW Command Syntax</i>	101
<i>FSVIEW Command</i>	101

Overview

The FSVIEW procedure enables you to browse or edit a SAS data set, displaying the data set as a table of rows and columns. You can also use it to create a new SAS data set.

The procedure provides tools for customizing an FSVIEW application. For example, you can redesign the display by changing the size, position, and colors of the FSVIEW window. You can also add computed variables, which display values that are calculated from other variables in the data set.

Note: You can also open the FSVIEW window by issuing an FSVIEW command from any SAS System command line. Δ

CAUTION:

The FSVIEW procedure edits a SAS data set in place. The FSVIEW procedure does not leave an unedited copy of the original. If you need to preserve the original data, be sure to make a copy of the data set before you begin editing. Δ

FSVIEW Procedure Syntax

Restriction: Do not use any of the other statements when you use the NEW= option in the PROC FSVIEW statement. (The ID and VAR statements cause errors; the FORMAT, INFORMAT, and WHERE statements are ignored.)

Reminder: When you use the FORMULA= option with the PROC FSVIEW statement to load an existing formula entry, the features that are defined in the entry take precedence over the features that are specified in the FORMAT, ID, INFORMAT, and VAR statements.

```
PROC FSVIEW <DATA=data-set> | <NEW=data-set <LIKE=data-set>>
  <FORMULA=SAS-catalog<.formula-entry>>
  <options>;
```

```
FORMAT variable-list format <... variable-list-n format-n>;
```

```
ID variable <... variable-n>;
```

```
INFORMAT variable-list informat <... variable-list-n informat-n>;
```

```
VAR variable <... variable-n>;
```

```
WHERE expression;
```

The PROC FSVIEW statement is required. The other statements are optional and are used as follows:

To do this	Use this statement
Associate formats with variables in the input data set	FORMAT
Specify the variable or variables that are used to identify observations instead of using observation numbers	ID
Associate informats with variables in the input data set	INFORMAT
Select which variables are available to the procedure	VAR
Specify a condition or set of conditions that observations in the input data set must meet in order to be processed	WHERE

PROC FSVIEW Statement

Initiates the FSVIEW procedure.

Requirement: The FSVIEW procedure must have an input data set. By default, the procedure uses the most recently created data set as its input data set. You can use the DATA= option in the PROC FSVIEW statement to select a particular data set. If you do not specify a data set and none has previously been created in the current SAS session, the procedure terminates with an error message.

Tip: Use the FORMULA= option to identify a FORMULA entry in which to store formulas for computed variables and other custom features of the FSVIEW session.

```
PROC FSVIEW <DATA=data-set> | <NEW=data-set <LIKE=data-set>>
  <FORMULA=SAS-catalog<.formula-entry>>
  <AUTOADD>
  <BROWSEONLY>
  <DEBUG>
  <EDIT | MODIFY>
  <NOADD>
  <NOBORDER>
```

<NODELETE>

Options

AUTOADD

turns on the autoadd feature. A new observation is displayed when the FSVIEW window is opened. Another new observation is added automatically whenever you enter values in the previous new observation and press ENTER.

The AUTOADD option is ignored if you also specify the BROWSEONLY or NOADD option with the PROC FSVIEW statement.

BROWSEONLY

BRONLY

prevents editing of data sets during the FSVIEW session. This option disables the MODIFY command so that data sets that are opened for browsing cannot be switched to editing. It also disables the EDIT command so that additional FSVIEW windows cannot be opened for editing.

DATA=*data-set*<(data-set-options)>

names the SAS data set to edit or browse. By default, the FSVIEW procedure displays the most recently created data set.

If you specify both the DATA= option and the NEW= option in the same PROC FSVIEW statement, the DATA= option is ignored.

You can add data set options following the data set name. The options must be enclosed in parentheses. The FIRSTOBS= and OBS= options are ignored; all other data set options are valid. Refer to *SAS Language Reference: Dictionary* for a list and descriptions of data set options.

DEBUG

turns on the SAS Component Language (SCL) source-level debugger. The debugger is useful for finding and correcting errors in formulas. See *SAS Component Language: Reference* for information on using the debugger.

EDIT

MODIFY

opens the initial FSVIEW window for editing the data set. By default, the initial FSVIEW window is opened for browsing.

This option is ignored if the BROWSEONLY option is also used. If the data set cannot be opened for editing, it is opened for browsing instead.

FORMULA=*SAS-catalog*<.formula-entry>

names a SAS catalog or specific FORMULA entry that contains formulas for computed variables and other FSVIEW session parameters to be associated with the data set, or in which the procedure can store formulas and parameter settings that are defined during the current FSVIEW session.

The general form of the *SAS-catalog* value is

<libref.>catalog-name

If the specified catalog does not already exist, it is created.

If you specify only a one-level name, it is treated as a catalog name in the default library, WORK. You must use a two-level catalog name if you want to specify a FORMULA entry name.

The general form of the *formula-entry* value is

entry-name<.FORMULA>

If you specify a two-level entry name with anything other than FORMULA as the second level (entry type), the specified type is ignored; FORMULA is used instead.

When an entry name is provided, the procedure attempts to load the specified entry when the initial FSVIEW window is opened. If the entry is not found, the FSVIEW session is initiated with default FSVIEW window characteristics. A FORMULA entry that has the specified name is created when you end the FSVIEW session or when you use the SAVE FORMULA command.

Note: If any variable names that are specified in the FORMULA entry do not match those in the data set to be displayed, then the FORMULA entry is not loaded and an error message is displayed. Δ

LIKE=*data-set*<(data-set-options)>

names an existing SAS data set whose structure is copied when a new SAS data set is created. (This option must be used in conjunction with the NEW= option.) When the FSVIEW NEW window is opened, the variable names and attributes of the specified data set are displayed.

You can add data set options following the data set name. The options must be enclosed in parentheses. Refer to *SAS Language Reference: Dictionary* for a list and descriptions of data set options.

MODIFY

See the EDIT option.

NEW=*data-set*<(data-set-options)>

creates a new SAS data set. If a data set that has the specified name already exists, the FSVIEW procedure terminates with an error message.

When this option is used, the procedure begins by opening the FSVIEW NEW window, in which the names and attributes of the variables in the new data set are defined. Use the LIKE= option in conjunction with the NEW= option to initialize the FSVIEW NEW window with the variable names and attributes of an existing data set. After the structure of the new data set is defined, the FSVIEW window is opened so that observations can be added. For details, see “Creating New SAS Data Sets” on page 127.

If you specify both the NEW= option and the DATA= option in the same PROC FSVIEW statement, the DATA= option is ignored.

You can add data set options following the data set name. The options must be enclosed in parentheses. Refer to *SAS Language Reference: Dictionary* for a list and descriptions of data set options.

NOADD

prevents the addition of observations to any data set that is edited during the current FSVIEW session. This option disables the AUTOADD and DUP commands so that no new observations can be added.

NOBORDER

suppresses the sides and bottom of the FSVIEW window’s border in a character-based display environment.

Note: This option is ignored in graphical windowing environments. Δ

When this option is used in a supported display environment, values can appear in the columns and row that the border normally occupies.

When the NOBORDER option is used in conjunction with the FORMULA= option, the window size that is specified in the FORMULA entry is ignored. The FSVIEW window always occupies the maximum possible number of rows and columns when the NOBORDER option is specified.

**NODELETE
NODEL**

prevents the deletion of observations from any data set that is edited during the current FSVIEW session. This option disables the DELETE command so that observations cannot be deleted.

FORMAT Statement

Associates formats with variables in the input data set. *Formats* are patterns that the SAS System uses to determine how variable values are displayed. The formats can be either SAS formats or custom formats that you have defined with the FORMAT procedure.

Reminder: The FORMAT statement is ignored when you use the NEW= option in the PROC FSVIEW statement.

Note: If you load an existing FORMULA entry, any formats that are recorded in the FORMULA entry take precedence over the formats that are specified in the FORMAT statement.

FORMAT *variable-list format <... variable-list-n format-n>;*

Arguments

At least one pair of the following arguments is required:

variable-list

consists of one or more variable names from the input data set.

format

is the SAS format or user-defined format to associate with the specified variable or variables.

Using the FORMAT Statement

You can use a single FORMAT statement to assign the same format to several variables or to assign different formats to different variables. You can use any number of FORMAT statements with each PROC FSVIEW statement.

Formats that are specified in a FORMAT statement take precedence over formats that are defined in the data set itself. Formats that are assigned in a FORMAT statement remain in effect only for the duration of the procedure; the FORMAT statement does not affect any format assignments that are stored in the data set. You can use the FORMAT command in the FSVIEW window to override formats that were specified in the FORMAT statement.

Format widths determine the width of the columns for the associated variables in the FSVIEW display.

Be aware that the format you assign a variable affects the informats you can assign with the INFORMAT statement. For example, suppose the data set that is displayed by the FSVIEW procedure contains a variable AMOUNT that is assigned the format DOLLAR10.2 but an informat of 10.2. Because of the format, values in the AMOUNT column are displayed with commas and a leading dollar sign, so the value 1250 would be displayed as \$1,250.00. However, if you edit this value (for example, changing it to \$1,150.00) and press ENTER, an error condition occurs. The 10.2 informat does not

permit the dollar sign (\$) or comma characters in entered values. An appropriate informat for this variable is COMMA., which does permit these characters.

Refer to *SAS Language Reference: Dictionary* for more detailed information on formats and on the FORMAT statement. See the description of the FORMAT procedure in *SAS Procedures Guide* for information on defining your own formats with the FORMAT procedure.

ID Statement

Specifies one or more variables whose values identify observations in the FSVIEW window.

Restriction: If you use the ID statement in conjunction with a PROC FSVIEW statement that includes the NEW= option, the FSVIEW procedure terminates with an error message.

Note: If you load an existing FORMULA entry, the ID variables that are recorded in the FORMULA entry take precedence over the ID variables that are specified in the ID statement.

ID *variable* <... *variable-n*>;

Argument

variable is the name of a variable from the displayed data set.

Using the ID Statement

By default, the FSVIEW procedure uses observation numbers to identify observations. Observation numbers are not displayed when ID variables are used. The values of the ID variables appear at the beginning of each row and remain on the left side of the window when the other variable columns are scrolled horizontally.

The FSVIEW procedure does not show more than the first 53 characters of ID variables. If the ID variable column exceeds this width, it is truncated.

If you want to know an observation's number when it is identified by an ID variable, use the OBS command. See "FSVIEW Window Commands" on page 108 for details.

Use the COLOR command to select the color for ID variables and for the ID variable column heading. See "FSVIEW Window Commands" on page 108 for details.

INFORMAT Statement

Associates informats with variables in the input data set. *Informats* are patterns that the SAS System uses to determine how values that are entered in variable columns are interpreted. The

informats can be either SAS informats or custom informats that you have defined with the FORMAT procedure.

Reminder: INFORMAT statements are ignored if you use the NEW= option in the PROC FSVIEW statement.

Note: If you load an existing FORMULA entry, any informats that are recorded in the FORMULA entry take precedence over the informats that are specified in the INFORMAT statement.

INFORMAT *variable-list informat <... variable-list-n informat-n>;*

Arguments

At least one pair of the following arguments is required:

variable-list

consists of one or more variable names from the input data set.

informat

is the SAS informat or user-defined informat to associate with the specified variable or variables.

Using the INFORMAT Statement

You can use a single INFORMAT statement to assign the same informat to several variables or to assign different informats to different variables. You can use any number of INFORMAT statements with each PROC FSVIEW statement.

Informats that are specified in an INFORMAT statement take precedence over informats that are defined in the data set itself. Informats that are assigned in an INFORMAT statement remain in effect only for the duration of the procedure; the INFORMAT statement does not affect any informat assignments that are stored in the data set. You can use the INFORMAT command in the FSVIEW window to override informats that were specified in the INFORMAT statement.

When using INFORMAT statements with the FSVIEW procedure, you should make sure the informats that you assign to variables are compatible with the formats for those variables. That is, the output that the format produces should be valid input for the informat. Otherwise, you complicate the process of editing variable values. For example, suppose the data set that is displayed by the FSVIEW procedure contains a variable AMOUNT that is assigned the informat 10.2 and the format DOLLAR10.2. Because of the format, values in the AMOUNT column are displayed with commas and a leading dollar sign, so the value 1250 would be displayed as \$1,250.00. However, if you edit this value (for example, changing it to \$1,150.00) and press ENTER, an error condition occurs. The 10.2 informat does not permit the dollar sign (\$) or comma characters in entered values. An appropriate informat for this variable is COMMA., which does permit these characters.

Refer to *SAS Language Reference: Dictionary* for more detailed information on informats and on the INFORMAT statement. See the description of the FORMAT procedure in *SAS Procedures Guide* for information on defining your own informats with the FORMAT procedure.

VAR Statement

Selects which variables to display and what order to display them in.

Restriction: The FSVIEW procedure terminates with an error message if you use the VAR statement in conjunction with a PROC FSVIEW statement that includes the NEW= option.

Note: If you load an existing FORMULA entry, the variables and variable order that are recorded in the FORMULA entry take precedence over the list and order of variables that are specified in the VAR statement.

Tip: When more than one variable name is supplied, any valid form of variable list can be used. See *SAS Language Reference: Concepts* for details about variable lists.

VAR *variable* <... *variable-n*>;

Argument

variable is the name of a variable from the displayed data set.

Using the VAR Statement

By default, the FSVIEW procedure displays all the variables from the data set in the FSVIEW window and arranges the variable columns in the order in which the variables appear in the data set (except for any variables that are used as ID variables). You can use the VAR statement to control which variables appear in the FSVIEW window and in what order they appear. Columns of variable values appear in the order that is specified in the VAR statement. The number of columns that are visible at any given time depends on the width of the variable values and on the current width of the FSVIEW window.

If the same variable name appears in both the ID and VAR statements, the variable is used only as an ID variable.

During an FSVIEW session, use the SHOW and DROP commands to control the display of variables. The SHOW command can add variables to the display even if they are not listed in the VAR statement. See “FSVIEW Window Commands” on page 108 for details.

Use the COLOR command to select the colors for displayed variables and column headings. See “FSVIEW Window Commands” on page 108 for details.

WHERE Statement

Defines criteria that observations in the input data set must meet in order to be displayed by the procedure.

Reminder: The WHERE statement is ignored if you use the NEW= option in the PROC FSVIEW statement.

WHERE *expression*;

Argument

expression is any valid WHERE expression that includes one or more of the variables in the input data set. Refer to the description of the WHERE statement in *SAS Language Reference: Dictionary* for details about the operators and operands that are valid in WHERE expressions.

Using the WHERE Statement

By default, the FSVIEW procedure displays all the observations in the data set. The WHERE statement is useful when you want to view only a subset of the observations in a SAS data set. For example, to view only observations for which the value of the variable YEAR is less than 5, follow the PROC FSVIEW statement with this statement:

```
where year<5;
```

The FSVIEW procedure displays only observations that meet the specified condition(s). Observations that do not satisfy the condition(s) are not shown and cannot be edited.

Conditions that are imposed by a WHERE statement (or, equivalently, by a WHERE= data set option) are called permanent WHERE clauses because they remain in effect for the duration of the FSVIEW session and cannot be canceled or modified while the procedure is active.

When you use a WHERE statement in conjunction with the PROC FSVIEW statement, the behavior of the FSVIEW procedure is affected in the following ways:

- The word *Subset* appears in parentheses following the FSVIEW window title to indicate that a permanent WHERE clause is in effect.
- Observation numbers might not be sequential because some observations might be excluded.
- You cannot use the SORT command to sort the displayed observations.
- You cannot scroll a particular observation to the top of the FSVIEW window by typing the corresponding observation number on the command line.

FSVIEW Command Syntax

Tip: The FSVIEW command provides an easy way to initiate an FSVIEW session from any SAS System command line.

```
FSVIEW <? | data-set <formula-name>>
```

FSVIEW Command

Initiates an FSVIEW session.

```
FSVIEW <? | data-set <formula-name>>
```

Arguments

You can specify the following arguments with the FSVIEW command:

?

opens a selection window from which you can choose the data set to be displayed by the FSVIEW procedure. The selection list in the window includes all data sets in all SAS data libraries that have been identified in the current SAS session (all data libraries that have defined librefs).

To select a data set, position the cursor on the desired data set name and press the ENTER key, or enter the desired data set name in the **Member Name** field.

data-set

specifies the data set to be displayed by the FSVIEW procedure. The general form of the argument is

< libref.>data-set-name< (data-set-options)>

If you omit the libref, then the default library, WORK, is assumed.

If you specify a data set that does not exist, a selection window is opened showing all available data sets. An error message in the selection window indicates that the specified data set does not exist.

If you omit this argument and do not specify ? for a selection window, the most recently created data set (the one identified in the `_LAST_` system option) is selected. If no data set has previously been created in the current SAS session, a selection window is opened showing all available data sets.

You can add a list of data set options following the data set name. The options must be enclosed in parentheses. The `FIRSTOBS=` and `OBS=` options are ignored; all other data set options are valid. Refer to *SAS Language Reference: Dictionary* for a list and descriptions of data set options.

formula-name

specifies a FORMULA entry that is associated with the FSVIEW session. The general form of the argument is

< libref.>catalog-name<.entry-name<.FORMULA>>

You can specify a one-, two-, three-, or four-level name:

- If a one-level name is specified, it is treated as a catalog name in the default library, WORK. If the specified catalog does not already exist in the WORK library, it is created. The procedure does not attempt to load a FORMULA entry when the FSVIEW session is initiated. When you end the FSVIEW session, a FORMULA entry is created, using the data set name as the FORMULA entry name.

Remember that all catalogs in the WORK library are erased when you end your SAS session.

- If a two-level name is specified, it is treated as *libref.catalog-name*. If the specified catalog does not already exist in the specified library, it is created. The procedure does not attempt to load a FORMULA entry when the FSVIEW session is initiated. When you end the FSVIEW session, a FORMULA entry is created in the specified catalog, using the data set name as the FORMULA entry name. (If an entry that has that name already exists in the catalog, it is replaced without warning.)

For example, suppose you initiate an FSVIEW session with the following command:

```
fsview master.subscrib master.custom
```

When you end the FSVIEW session, the FORMULA entry MASTER.CUSTOM.SUBSCRIB.FORMULA is created.

- If a three-level name is specified, it is treated as *libref.catalog-name.entry-name*. The entry type is assumed to be FORMULA. The procedure attempts to load the specified FORMULA entry when the FSVIEW session is initiated. If the specified FORMULA entry is not found, the FSVIEW session is initiated without an associated FORMULA entry. A FORMULA entry that has the specified name (and an entry type of FORMULA) is created when you end the FSVIEW session.
- If a four-level name is specified, the fourth level (entry type) should be FORMULA. Any other value for the type is ignored; FORMULA is used instead. The procedure attempts to load the specified FORMULA entry when the FSVIEW session is initiated. If the specified FORMULA entry is not found, the FSVIEW session is initiated without an associated FORMULA entry. A FORMULA entry that has the specified name is created when you end the FSVIEW session.

If any of the variable names that are specified in the FORMULA entry do not match those in the data set to be displayed, then the FORMULA entry is not loaded, and an error message is displayed.

Using the FSVIEW Command

The FSVIEW command always opens the data set for browsing. Use the MODIFY command to edit the displayed entries.

When you end an FSVIEW session that was initiated with the FSVIEW command, you return to the window that was active when the command was issued. Thus, the command is particularly useful when you want to view the contents of a data set while another interactive windowing procedure is executing.

The FSVIEW command does not support procedure options such as BROWSEONLY and NOADD. You must use the PROC FSVIEW statement rather than the FSVIEW command if you want to invoke the procedure with these options, or if you want to establish default procedure characteristics with the FORMAT, ID, INFORMAT, VAR, or WHERE statements.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/FSP® Software Procedures Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/FSP® Software Procedures Guide, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-517-5

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.