

CHAPTER

8

FSVIEW Procedure Windows

<i>Overview</i>	105
<i>Browsing and Editing SAS Data Sets</i>	106
<i>How the Control Level Affects Editing</i>	107
<i>Editing with Record-level Control</i>	107
<i>Editing with Member-level Control</i>	108
<i>Opening Multiple FSVIEW Windows</i>	108
<i>FSVIEW Window Commands</i>	108
<i>Command Descriptions</i>	110
<i>Creating New SAS Data Sets</i>	127
<i>Opening the FSVIEW NEW Window</i>	127
<i>Using the NEW= Option</i>	128
<i>Using the NEW Command</i>	128
<i>Creating a Data Set That Is Like an Existing One</i>	129
<i>Closing the FSVIEW NEW Window</i>	129
<i>FSVIEW NEW Window Commands</i>	129
<i>Command Descriptions</i>	130
<i>Duplicating Existing SAS Data Sets</i>	131
<i>Selecting Variables for FSVIEW Operations</i>	131
<i>Defining Formulas</i>	132
<i>How Formulas Are Evaluated</i>	132
<i>Using SAS Component Language in Formulas</i>	133
<i>Defining Formulas in the FSVIEW Define Window</i>	133
<i>FSVIEW Define Window Commands</i>	135
<i>Reviewing Formula Definitions</i>	135
<i>FSVIEW REVIEW Window Commands</i>	136
<i>Customizing the FSVIEW Environment</i>	136
<i>Parameter Descriptions</i>	137
<i>FSVIEW Parameters Window Commands</i>	139
<i>Creating FSVIEW Applications</i>	139
<i>Creating and Updating Formula Entries</i>	140
<i>Loading Formula Entries</i>	140

Overview

You can use the FSVIEW procedure to perform the following tasks:

- browse and edit existing SAS data sets
- create new SAS data sets
- duplicate existing SAS data sets

- define formulas for creating computed variables or for manipulating data set variables
- customize the FSVIEW environment by setting general parameters
- create data entry or data browsing applications.

The following sections describe the windows and commands you use to accomplish these tasks.

Browsing and Editing SAS Data Sets

In the FSVIEW procedure, observations are displayed in the FSVIEW window. Display 8.1 on page 106 identifies important features of the FSVIEW window.

Display 8.1 The FSVIEW Window

Obs	style	sqfeet	bedrooms	baths
1	RANCH	1250	2	1
2	SPLIT	1190	1	1
3	CONDO	1400	2	1.5
4	TWOSTORY	1810	4	3
5	RANCH	1500	3	3
6	SPLIT	1615	4	3
7	SPLIT	1305	3	1.5
8	CONDO	1390	3	2.5
9	TWOSTORY	1040	2	1
10	CONDO	2105	4	2.5
11	RANCH	1535	3	3
12	TWOSTORY	1240	2	1
13	RANCH	720	1	1
14	TWOSTORY	1745	4	2.5
15	CONDO	1860	2	2

By default,

- the leftmost column has the heading *Obs* and contains the observation number of the corresponding observations.

Note: If the engine does not support observation numbers, the observation number column has the heading *Row* instead of *Obs*. Δ

You can use variable values instead of observation numbers for row identifier columns.

- the other column headings are the names of the corresponding data set variables. You can assign other headings to the rows.
- the FSVIEW procedure opens the input data set for browsing. Use the MODIFY command if you want to edit the data set. (If you use the PROC FSVIEW statement to start the procedure, you can add the EDIT or MODIFY option to directly open the data set for editing.) If you use the procedure to create a new data set, the FSVIEW window is opened for editing after the structure of the new data set has been defined in the FSVIEW NEW window.

Unless you use a WHERE statement in conjunction with the PROC FSVIEW statement, all observations in the data set are available for browsing or editing. The FSVIEW procedure ignores the FIRSTOBS= and OBS= system options.

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 a copy of the original data, be sure to copy the data set before you begin editing. △

How the Control Level Affects Editing

The editing behavior of the FSVIEW procedure depends on which control level you select when the data set is opened. The *control level* is the degree to which the procedure can restrict access to the data set.

The FSVIEW procedure supports two levels of control:

record	locks only the observation that is currently being edited. With this control level, you can open multiple FSVIEW windows for browsing or editing the same data set. Other users can edit the same data set simultaneously.
member	locks the entire data set. No other window or user can open the data set while this control level is in effect.

By default, the FSVIEW procedure selects record-level control when it opens a SAS data set. You can specify the control level with the MODIFY command in the FSVIEW window, or by using the CNTLLEV= data set option with the data set name in the PROC FSVIEW statement, FSVIEW command, or BROWSE or EDIT command. The CNTLLEV= data set option is described in *SAS Language Reference: Dictionary*.

Editing with Record-level Control

When record-level control is selected, you must lock an observation for editing before you can change variable values in that observation. To lock an observation, move the cursor to the line for the desired observation and press ENTER. The line is highlighted to indicate that it is locked. (You can use the HI command to control which type of highlighting is used.) Once the observation is locked, you can change values in the variable columns. Scroll right or left to display additional columns. If you have defined formulas for any data set variables, the values of the variables that have formulas are updated only when observations are locked.

The lock on the current observation is released when you lock a different observation for editing. In addition, the lock is released when any of the following commands cause the currently locked observation to scroll out of the window:

FORWARD	<i>n</i>
BACKWARD	AUTOADD
TOP	DUP
BOTTOM	

The lock is always released when you issue any of the following commands:

DELETE
 SORT
 WHERE

Editing with Member-level Control

When the MEMBER control level is selected, the FSVIEW procedure obtains exclusive control over the data set. In this case, all observations are available for editing. It is not necessary to select an individual observation before editing; simply move the cursor to the desired observation and enter the new value in the desired variable column.

Opening Multiple FSVIEW Windows

After entering the FSVIEW procedure, you can use the BROWSE, EDIT, and NEW commands to open additional FSVIEW windows to concurrently browse or edit other SAS data sets. The multiple FSVIEW windows may overlap visually, but each is completely independent of the others.

Opening additional FSVIEW windows within the current FSVIEW session by using the EDIT or BROWSE commands consumes fewer computer resources than starting additional FSVIEW sessions with the FSVIEW command.

If you invoke the FSVIEW procedure with a PROC FSVIEW statement, some of the PROC FSVIEW statement options affect the behavior of the additional FSVIEW windows:

- If you use the BROWSEONLY option, the MODIFY and EDIT commands are not permitted in any FSVIEW windows that are opened during the FSVIEW session.
- If you use the NOADD option, the AUTOADD and DUP commands are not permitted in any FSVIEW windows that are opened during the FSVIEW session.
- If you use the NODELETE option, the DELETE command is not permitted in any FSVIEW windows that are opened during the FSVIEW session.

FSVIEW Window Commands

In addition to the global commands that are discussed in Chapter 9, “SAS/FSP Software Global Commands,” on page 141, you can issue the following commands when editing a data set with the FSVIEW procedure. When browsing, you can use all the following commands except those that are identified as editing commands:

Scrolling

n

=variable

BACKWARD <*n* | HALF | PAGE | MAX>

BOTTOM

FORWARD <*n* | HALF | PAGE | MAX>

HSCROLL *n*

LEFT <*n* | MAX>

RIGHT <*n* | MAX>

SMOOTH <ON | OFF>
 TOP
 VSCROLL *n* | HALF | PAGE

Managing the Data Set

AUTOSAVE <*n*>
 BROWSE *data-set* <FORMULA=*SAS-catalog*<.*formula-entry*>>
 CANCEL
 CREATE *data-set* <REPLACE> <*variable-list* | ALL | ?>
 EDIT *data-set* <FORMULA=*SAS-catalog*<.*formula-entry*>>
 END
 MODIFY <RECORD | MEMBER>
 NEW *data-set* <LIKE=*data-set*>
 <FORMULA=*SAS-catalog*<.*formula-entry*>>
 SAVE
 WHERE <<ALSO> *expression*> <UNDO | CLEAR>

Editing the Data Set

(The commands in this group are valid only when the FSVIEW window is opened for editing.)

AUTOADD | ADD <ON | OFF>
 CURSOR <*variable*>
 DELETE | DEL <*obs* <... *obs-n*>>
 DUP <*n* <*obs*>>
 INITIAL <*obs* | CLEAR>
 PROTECT ON | OFF <*variable* <... *variable-n*>>
 SORT <ASCENDING | DESCENDING> *variable* <... <ASCENDING |
 DESCENDING> *variable-n*>

Managing Formulas

DEFINE <*variable-name* <\$>
 <(<*format*><,<*informat*><,<*label*>>)><=*formula*>>
 FORMULA <*formula-name*>
 RESET <ALL | *variable* <... *variable-n*>>
 REVIEW
 SAVE FORMULA <*formula-name*>

Changing the Display

COLOR *area color* <*highlight*>
 DROP <*variable* <... *variable-n*>>
 FORMAT *variable-list* '*format*' <... *variable-list-n* '*format-n*'>
 HI <ON | OFF> | <*color* <*highlight*>> | <RESET <ALL>>

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

MOVE *variable* | *variable-range* <*target-variable*>

RENAME *current-variable-name* *new-variable-name*

SETWSZ <CLEAR>

SHOW ID | VAR *variable* <...*variable-n*>

Other

KEYS

OBS

PARMS

Command Descriptions

Here are descriptions of the FSVIEW window commands:

n

scrolls the FSVIEW window vertically so that the specified observation occupies the top line. Type the desired observation number on the command line, then press ENTER. You receive an error message if you specify a value that is greater than the highest observation number.

The *n* command is still valid when ID variables are used in place of observation numbers to identify observations. However, the command is not valid when the engine that is being used to read the data set does not support access by observation number. When an engine does not support observation numbers, the column label is ROW instead of OBS. The command also is not valid while a permanent or temporary WHERE clause is in effect.

=*variable*

scrolls the FSVIEW window horizontally so that the column for the specified variable is visible.

AUTOADD <ON | OFF>

ADD <ON | OFF>

(editing command; not valid while browsing a data set)

turns the autoadd feature on or off. While the autoadd feature is on, a new observation is always displayed. If observation numbers are used in the ID column, the new observation is identified with the word NEW.

Note: The AUTOADD command is not permitted if the FSVIEW session is initiated with a PROC FSVIEW statement that includes the NOADD option. Δ

By default, all values in the new observation are missing. You can use the INITIAL command to store the current contents of an existing observation; the stored values are then copied into new observations when they are displayed by the autoadd feature.

By default, the cursor is positioned on the column for the leftmost variable in the FSVIEW window whenever a new observation is displayed. You can use the CURSOR command to specify the variable on which the cursor is positioned when the new observation is displayed.

The new observation is not actually added to the displayed data set until you add values and press ENTER, or until you issue any FSVIEW command other than LEFT or RIGHT. While entering values, you can use the LEFT and RIGHT commands to scroll among the variable columns without adding the new

observation to the data set. After the observation is added, another new observation is displayed.

If the new observation has not been modified when you issue the END command, the observation is not added to the data set.

When the AUTOADD command is used without the ON or OFF arguments, it acts as a toggle, turning the autoadd feature on if it is currently off or off if it is currently on.

AUTOSAVE $\langle n \rangle$

specifies how frequently the FSVIEW procedure automatically saves the data set. The n value determines how many modifications must be made before an automatic save is performed. By default, the procedure saves the data set automatically whenever 25 observations have been modified since the last save.

To check the current value of AUTOSAVE parameter, issue the AUTOSAVE command without specifying an n value.

You can change the default AUTOSAVE value for an FSVIEW application by changing the value in the **Autosave value** field in the FSVIEW Parameters window. See “Customizing the FSVIEW Environment” on page 136 for details.

Regardless of the AUTOSAVE value, you can save the data set at any time using the SAVE command.

BACKWARD $\langle n \mid \text{HALF} \mid \text{PAGE} \mid \text{MAX} \rangle$

scrolls vertically toward the top of the window. The following scroll amounts can be specified:

n	scrolls upward by the specified number of observations.
HALF	scrolls upward by half the number of lines in the window.
PAGE	scrolls upward by the number of lines in the window.
MAX	scrolls upward until the first observation is at the top of the window.

If the scrolling increment is not explicitly specified, the window is scrolled by the amount specified in the VSCROLL parameter. The default VSCROLL amount is HALF.

BOTTOM

scrolls forward until the last observation in the data set is displayed at the bottom of the window.

You can interrupt the BOTTOM scroll operation before the last observation is reached. This feature is useful when you want to halt a scroll request while you are browsing or editing a large data set. To halt a BOTTOM scroll operation that is in progress, press the interrupt key or key combination for your system. The FSVIEW procedure displays a requestor window that gives you the options of canceling or resuming the scrolling operation.

Note: The key or key combination that is used to interrupt an active process depends on your host operating system and terminal device. For example, some systems have a key labeled BREAK or ATTENTION (or ATTN). Other systems use a combination of the CTRL key and another key. Refer to your host documentation to determine the interrupt key or key combination for your host operating system and terminal device. Δ

BROWSE *data-set* \langle *data-set-options* \rangle \langle FORMULA=*SAS-catalog* \langle *.formula-entry* \rangle \rangle

opens another FSVIEW window and displays the specified SAS data set for browsing. The current FSVIEW window remains open, but the new FSVIEW window becomes the active window. All FSVIEW procedure browsing commands are valid in the new window (except for those that are disabled with PROC

FSVIEW statement options). When you use the END command to close the new window, you return to the FSVIEW window from which the BROWSE command was issued.

The additional FSVIEW window may overlay the current one. To control where the additional window appears, use the WREGION command before issuing the BROWSE command. You can use the SWAP command to move among concurrent FSVIEW windows.

You can use the MODIFY command to change the new window from browsing to editing, unless the procedure is initiated by a PROC FSVIEW statement that includes the BROWSEONLY option.

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.

You can add the FORMULA= argument to the BROWSE command to associate a FORMULA entry with the displayed data set. The syntax for the FORMULA= argument is the same as for the PROC FSVIEW statement's FORMULA= option.

CANCEL

closes the FSVIEW window and saves the data set without recording any updated information in the associated FORMULA entry. (Changes to variable values are recorded; changes to formulas and to other general parameters are not.) If only one FSVIEW window is open in the current FSVIEW session, the CANCEL command also ends the FSVIEW procedure.

COLOR *area color<highlight>*

sets the color and highlighting attributes of different areas of the window. You can specify the following areas:

VAR	variable value columns (except for ID variables)
VARNAME	variable names that are used as column headings (except for ID variables)
ID	ID value columns (or the observation number column if ID variables are not used)
IDNAME	ID variable names that are used as column headings (or the heading of the observation number column if ID variables are not used).

The following values are valid for the *color* argument:

BLACK	CYAN	MAGENTA	RED
BLUE	GRAY	ORANGE	WHITE
BROWN	GREEN	PINK	YELLOW

The following values are valid for the *highlight* argument:

H (high intensity)	U (underlined)
R (reverse video)	B (blinking)

In addition to the areas specific to the FSVIEW procedure, you can specify colors and attributes for the following FSVIEW window areas by using the global COLOR command:

BACKGROUND	BORDER	MESSAGE
BANNER	COMMAND	

Refer to the online Help for base SAS software for more information on the COLOR command.

CREATE *data-set*<(data-set-options)> <REPLACE> <variable-list | ALL | ?>
creates a new SAS data set, using some or all of the variables from the data set that is currently displayed. The new data set duplicates both the structure and contents of the displayed data set. You can select which variables are included in the new data set in any of the following ways:

- by listing the desired variable names as command arguments.
- by using the ALL argument to select all variables in the displayed data set.
- by using the ? argument to open the Select Table Variables window. In this window you can select the desired variables from a list of all available variables in the displayed data set. Refer to “Selecting Variables for FSVIEW Operations” on page 131 for more information on using the Select Table Variables window.

Note: The Select Table Variables window is also opened if you issue a CREATE command that includes neither variable names nor the ALL or ? arguments. By default, all variables are selected in the Select Table Variables window. △

Both computed variables and data set variables can be selected. Computed variables in the displayed data set become data set variables in the created data set.

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.

By default, the CREATE command fails with an error message if the named data set already exists. If you want the new data set to replace an existing data set that has the same name, specify the REPLACE option. If you use the REPLACE option, it must precede the variable name arguments.

CURSOR <variable>

(editing command; not valid while browsing a data set)

specifies the variable column on which the cursor is positioned when a new observation is displayed. (New observations are created by the autoadd feature or by the DUP command.) By default, the cursor is positioned on the leftmost variable column in the FSVIEW window when the new observation is added.

You can specify the desired variable name as an argument for the CURSOR command, or you can type CURSOR on the command line, position the cursor on the desired variable column, and press ENTER.

DEFINE <variable-name <\$> <(<format><,<informat><,<label>>)> <=formula>>

defines a formula for a computed variable or for an existing data set variable.

When you use the DEFINE command alone or with only the *variable-name* argument, the FSVIEW Define window is opened. In this window you can enter all the information necessary to define the formula. You can also use this form of the command to open the FSVIEW Define window to edit an existing formula. See “Defining Formulas” on page 132 for more information on using the FSVIEW Define window.

If the variable name that you specify does not exist in the displayed data set, the FSVIEW procedure creates a computed variable. When creating a computed variable, the procedure assumes by default that the variable is numeric. To create

a character variable, add the \$ argument following the variable name if you are specifying the formula in the DEFINE command argument. (Do not use the \$ argument if you want to open the FSVIEW Define window.)

You can supply the formula as an argument with the DEFINE command. In this case, the FSVIEW Define window is not opened. Separate the formula from the variable name with an equal sign (=).

Formulas are not limited to SAS expressions. They can also include any valid SAS Component Language (SCL) functions or statements except those that are valid only in SAS/AF software and the following statements:

CONTROL (except for CONTROL ENTER)
 CURSOR
 ERRORON
 ERROROFF
 PROTECT
 UNPROTECT

Refer to *SAS Component Language: Reference* for details about including SCL elements in FSVIEW formulas.

You can specify a format, informat, and label for the variable in the DEFINE command. The arguments for these values must be enclosed in parentheses and must appear before the equal sign that begins the formula definition. The values must be specified in the order *format, informat, label*. You can omit any of the values as long as you include the necessary commas.

If you omit the *format* argument when defining a computed variable, the default format is BEST12. for numeric variables or \$8. for character variables. If you omit the *informat* argument, the default informat is 12. for numeric variables and \$8. for character variables.

You can also use the *format, informat, and label* arguments to change the format, informat, or label of an existing variable. For an existing variable, you can supply these arguments without supplying a formula. (The arguments must still be enclosed in parentheses.) For data set variables, these arguments change the format, informat, and label that are recorded in the FORMULA entry, but they do not affect any format, informat, or label for the variable that is recorded in the data set itself.

If the cursor is positioned on a variable column when the DEFINE command is issued, the column for the new computed variable appears immediately to the right of the column on which the cursor was positioned. Otherwise, the column for the new computed variable appears to the right of the displayed variable columns.

DELETE <obs <... obs-n>>

DEL <obs <... obs-n>>

(editing command; not valid while browsing a data set)

deletes one or more observations from the data set you are editing.

CAUTION:

The DELETE command deletes observations from the displayed data set, not just from the FSVIEW window. You cannot recover the contents of a deleted observation. Δ

Note: The DELETE command is not permitted if the FSVIEW session is initiated with a PROC FSVIEW statement that includes the NODELETE option. Δ

The behavior of the DELETE command depends on the control level at which the data set is currently opened for editing.

For member-level control

To delete a single observation, follow the DELETE command with the observation number of the observation to be deleted. You can also type

DELETE (or DEL) on the command line, position the cursor on the observation to be deleted, and press ENTER.

To delete multiple observations, follow the DELETE command with a list of observation numbers. Separate the observation numbers with at least one space. For example, issue the following command to delete observations 5 and 10:

```
delete 5 10
```

To delete a range of observations, specify the first and last observation numbers of the range, separated by a dash. For example, the following command deletes all observations between 5 and 10, inclusive:

```
delete 5-10
```

For record-level control

You can delete only one observation at a time, and you must lock the observation before you can delete it. Either follow the DELETE command with the observation number for the locked observation to be deleted or type DELETE (or DEL) on the command line, position the cursor on the locked observation to be deleted, and press ENTER.

DROP <variable <...variable-n>>

excludes one or more variables from the FSVIEW window. The DROP command affects only the display, not the actual data set.

If you issue the DROP command without arguments, the Select Table Variables window is opened for you to select the variables to be dropped. Refer to “Selecting Variables for FSVIEW Operations” on page 131 for more information on using the Select Table Variables window.

To exclude a single variable, follow the DROP command with the name of the variable to be dropped. Alternatively, you can type DROP on the command line, position the cursor on the variable column to be dropped, and press ENTER.

To exclude multiple variables, follow the DROP command with a list of variable names. Separate the variable names with at least one space. For example, the following command drops the variables X and Z from the display:

```
drop x z
```

You can indicate a range of variables by specifying the first and last variables to be dropped, separated by a dash. For example, the following command drops all the variables between X and Z, inclusive:

```
drop x-z
```

Note: At least one variable, in addition to any ID variables specified, must remain displayed. Δ

Use the SHOW command to redisplay dropped variables.

DUP <n <obs>>

(editing command; not valid while browsing a data set)

copies the specified observation *n* times and adds the new observation(s) to the displayed data set. The FSVIEW window is automatically scrolled forward so that the new observations appear at the top of the window.

Note: The DUP command is not permitted if the FSVIEW session is initiated with a PROC FSVIEW statement that includes the NOADD option. Δ

You can select the observation to copy by supplying its number as the *obs* argument in the DUP command. To specify the *obs* argument, you must also

specify the *n* argument (the number of times you want the observation duplicated). For example, the following command duplicates observation 5 one time:

```
dup 1 5
```

Alternatively, you can select the observation to copy by typing DUP on the command line, positioning the cursor on the desired observation, and pressing ENTER. By default, the observation is duplicated once. To duplicate the same observation again, leave the cursor on the command line and issue the DUP command again. Alternatively, you can follow the DUP command with the desired number of copies. For example, if the cursor is on the command line, the following command duplicates the most recently duplicated observation three times:

```
dup 3
```

If the cursor is positioned on an observation, then that observation is duplicated three times.

By default, the cursor is positioned on the leftmost displayed variable column of the first added observation. You can use the CURSOR command to select a particular variable to which the cursor is moved when the duplicate observation is displayed.

EDIT *data-set*<(data-set-options) <FORMULA=*SAS-catalog*<.formula-entry>>
opens another FSVIEW window and displays the specified data set for editing. The current FSVIEW window remains open, but the new FSVIEW window becomes the active window. All FSVIEW window commands are valid in the new window. When you use the END command to close the new window, you return to the FSVIEW window from which the EDIT command was issued.

Note: The EDIT command is not valid if the FSVIEW session is initiated by a PROC FSVIEW statement that includes the BROWSEONLY option. Δ

The additional FSVIEW window may overlay the current one. To control where the additional window appears, use the WREGION command before issuing the EDIT command. You can use the SWAP command to move among concurrent FSVIEW windows.

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.

END
saves any changes that you make to the displayed data set (if it is open for editing), updates the FORMULA entry (if a formula name or formula catalog has been specified), closes the current FSVIEW window, and returns to the previous FSVIEW window.

If only one FSVIEW window is open, the END command ends the FSVIEW procedure.

FORMAT *variable-list* 'format' <... *variable-list-n* 'format-n'
changes the format of one or more variables for display purposes only. The FORMAT command does not change any formats that are stored with the variables in the data set. Formats that are specified in the FORMAT command

take precedence over formats that have been defined in the displayed data set or that are assigned in FORMAT statements.

The format name can be any SAS format, or it can be a custom format that you have defined with the FORMAT procedure. The format name must be enclosed in quotes.

You can specify multiple variables either individually or as a range. Separate two variable names with a dash to indicate a range. For example, the following command changes the display formats of variables X and Z only:

```
format x z 'date7.'
```

The following command changes the display format of all the variables between X and Z, inclusive:

```
format x-z 'date7.'
```

You can assign multiple formats in a single FORMAT command. For example, the following command assigns different display formats to the variables X and Z:

```
format x 'date7.' z 'monyy5.'
```

Note: 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. Δ

To remove a format assignment from the FORMULA entry so that the format that is defined in the data set is used, issue a FORMAT command that specifies the variable name but no format name. For example, to cancel the format for the variable X that is stored in the FORMULA entry, issue the following command in the FSVIEW window:

```
format x
```

FORMULA <formula-name>

reads in a FORMULA entry.

The general form of the *formula-name* 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 an entry name in the current formula catalog. The entry type is assumed to be FORMULA. If no catalog has been previously specified in a FORMULA command, in the FORMULA= option of the PROC FSVIEW statement, or in the *formula-name* argument of the FSVIEW command, then the procedure looks for the specified entry in your personal PROFILE catalog (SASUSER.PROFILE, or WORK.PROFILE if the SASUSER library is not allocated.)
- If a two-level name is specified, the second level must be FORMULA. Any other value causes an error message. The name is treated as an entry name in the current formula catalog. If no formula catalog has been previously

specified, the procedure looks for the specified entry in your personal PROFILE catalog.

- If a three-level name is specified, it is treated as *libref.catalog-name.entry-name*. The entry type is assumed to be FORMULA. You receive an error message if the specified libref is undefined or if the specified catalog does not exist.
- If a four-level name is specified, the fourth level must be FORMULA. Any other value causes an error message. You receive an error message if the specified libref is undefined or if the specified catalog does not exist.

If you omit the *formula-name* argument altogether, the procedure looks for an entry that has the default name. The default formula name is the name that was used in the previous FORMULA command. If no previous FORMULA command has been issued, then the default name is the name that was specified in the FORMULA= option of the PROC FSVIEW statement or in the *formula-name* argument of the FSVIEW command when the procedure was initiated. If no formula name was specified when the procedure was invoked, then the name of the displayed data set is used as the default FORMULA entry name. The procedure looks for the default entry in the current formula catalog. If no formula catalog has previously been specified, then the procedure looks for the default entry in your personal PROFILE catalog.

If the specified FORMULA entry exists, it replaces the FORMULA entry that is currently being used for the FSVIEW window. All current formula definitions are removed.

If the specified entry does not currently exist, then no entry is read in and a warning message is issued. An entry that has the specified name is created when you issue an END command to close the FSVIEW window, or when you issue a SAVE FORMULA command without specifying another formula name.

The FORMULA command is valid even if you do not use the FORMULA= option in the PROC FSVIEW statement or the *formula-name* argument in the FSVIEW command when you initiate the FSVIEW session.

FORWARD $\langle n \mid \text{HALF} \mid \text{PAGE} \mid \text{MAX} \rangle$

scrolls vertically toward the bottom of the window. The following scroll amounts can be specified:

<i>n</i>	scrolls downward by the specified number of observations.
HALF	scrolls downward by half the number of lines in the window.
PAGE	scrolls downward by the number of lines in the window.
MAX	scrolls downward until the last observation is at the bottom of the window.

Note: See the discussion of the BOTTOM command for details about the restrictions that apply. Δ

If the scrolling increment is not explicitly specified, the window is scrolled by the amount that is specified in the VSCROLL parameter. The default VSCROLL amount is HALF.

Note: Regardless of the specified scroll increment, the FSVIEW procedure does not scroll beyond the last observation in the data set. Δ

HI $\langle \text{ON} \mid \text{OFF} \rangle$

HI $\langle \text{color} \langle \text{highlight} \rangle \rangle$

HI $\langle \text{RESET} \langle \text{ALL} \rangle \rangle$

controls the highlighting status of an individual row or of the entire window, or sets the highlighting characteristics for an individual row or for the entire window.

To turn highlighting on for an unhighlighted row or to turn it off for a highlighted row, use the HI command. Type HI on the command line, position the

cursor on the row that you want the command to affect, and press ENTER (or position the cursor on the desired row and press the function key to which the HI command has been assigned).

To turn highlighting on for all unhighlighted rows, use the HI ON command. To turn highlighting off for all highlighted rows, use the HI OFF command.

To set default highlighting characteristics for the entire window, follow the HI command with the desired color and (optionally) an attribute. The following values are valid for the *color* argument:

BLACK	CYAN	MAGENTA	RED
BLUE	GRAY	ORANGE	WHITE
BROWN	GREEN	PINK	YELLOW

The following values are valid for the *highlight* argument:

H (high intensity)	U (underlined)
R (reverse video)	B (blinking)

To change the highlighting characteristics for an individual row, type HI on the command line, followed by the new color and (optionally) the new attribute, position the cursor on the desired row, and then press ENTER.

To reset all rows to the default color and highlighting attribute, use the HI RESET ALL command.

To reset the highlight characteristics of an individual row to the default, type HI RESET on the command line, position the cursor on the desired row, and then press ENTER.

HSCROLL *n*

sets the default horizontal scrolling amount for the LEFT and RIGHT commands. For *n*, specify the number of variable columns to scroll by. The default amount is one column.

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

changes the informat of one or more variables in the FSVIEW window. The INFORMAT command does not change any informats that are stored with the variables in the data set. Informats that are specified with the INFORMAT command take precedence over informats that are defined in the data set or assigned in INFORMAT statements.

The informat name can be any SAS informat, or it can be a custom informat that you have defined with the FORMAT procedure. The informat name must be enclosed in quotes.

You can list multiple variable names individually or as a range. Separate two variable names with a dash to indicate a range. For example, the following command selects the informats for the variables X and Z only:

```
informat x z 'date7.'
```

The following command selects the informat for all the variables between X and Z, inclusive:

```
informat x-z 'date7.'
```

You can also assign multiple informats in a single INFORMAT command. For example, the following command selects different informats for the variables X and Z:

```
informat x 'date7.' z 'monyy5.'
```

Note: When using informats 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 the values of variables. 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. Δ

To remove an informat assignment from the FORMULA entry so that the informat that is defined the data set is used, issue an INFORMAT command that specifies the variable name but no informat name. For example, to cancel the informat for the variable X that is stored in the FORMULA entry, issue the following command in the FSVIEW window:

```
informat x
```

INITIAL <obs | CLEAR>

(editing command; not valid while browsing a data set)

selects an observation whose contents are used as initial values for added observations. By default, all variables in a new observation that is added by the autoadd feature are initialized with missing values. You can use this command to record the current variable values from an existing observation for use as initial values for new observations.

The values of the selected observation are recorded in the FORMULA entry. The recorded values are then used to initialize autoadded observations. Any subsequent changes that you make to the observation designated in the INITIAL command do not affect the stored values that are used to initialize new observations.

You can identify the desired observation by specifying its observation number as the *obs* argument in the INITIAL command, or you can type INITIAL on the command line, position the cursor on the desired observation, and press ENTER.

Use the INITIAL CLEAR command to reset the initial variable values for new observations to missing values.

KEYS

opens the KEYS window for browsing and editing function key definitions for the FSVIEW window. Function key definitions are stored in catalog entries of type KEYS.

The default function key definitions for the FSVIEW window are stored in the FSVIEW.KEYS entry in the SASHELP.FSP system catalog. If you are using this default set of key definitions when you issue the KEYS command and you change any key definitions in the KEYS window, a new copy of the FSVIEW.KEYS entry is created in your personal PROFILE catalog (SASUSER.PROFILE, or

WORK.PROFILE if the SASUSER library is not allocated). The changes that you make are recorded in your personal copy of the KEYS entry.

LEFT <*n* | MAX>

scrolls the FSVIEW window to the left in increments of variable columns. The following scroll amounts can be specified:

n scrolls to the left by the specified number of variable columns.

MAX scrolls to the leftmost variable column.

If the LEFT command is used without arguments, the FSVIEW window is scrolled by the amount that is specified in the HSCROLL parameter. The default HSCROLL amount is one column.

MODIFY <RECORD | MEMBER>

UPDATE <RECORD | MEMBER>

changes the FSVIEW window from browsing to editing, or changes the control level of an FSVIEW window that is already open for editing.

Note: The MODIFY command is not permitted if the FSVIEW session is initiated with a PROC FSVIEW statement that includes the BROWSEONLY option. Δ

The data set can be opened for editing with either member- or record-level control. (Refer to “How the Control Level Affects Editing” on page 107 for more information on control levels.) You can specify a control level by using the RECORD or MEMBER arguments with the MODIFY command. If you do not use either argument, the default is record-level control (unless you use the CNTLLEV=MEMBER data set option with the data set name when you open the FSVIEW window).

When the FSVIEW window is open for editing, you can use the MODIFY RECORD and MODIFY MEMBER commands to change the current control level for the window.

The MODIFY command fails if the specified control level would cause a locking conflict. For example, you cannot specify MODIFY MEMBER if the same data set is open with record-level control in another FSVIEW window.

MOVE *variable* | *variable-range* <*target-variable*>

moves a variable column or a range of variable columns.

Note: The MOVE command affects only the order of variables in the FSVIEW window, not the actual position of the variables in the data set. Δ

To move a single variable, type MOVE on the command line, followed by the name of the variable to be moved; then place the cursor on the variable that you want the moved variable to follow and press ENTER.

To move a range of variables, specify the name of the first and last variables of the range, separated by a dash. All variables in the range must be of the same class, either ID or VAR. Indicate the variable that you want the range to follow by specifying the variable name as the *target-variable* argument. For example, the following command moves the variables VAR1 through VAR3 to the right of the variable VAR6:

```
move var1-var3 var6
```

You can also move a range of variables by typing MOVE and the variable range on the command line, positioning the cursor on the desired target variable, and pressing ENTER.

In order to move only one variable and specify the target variable as a command argument, you must supply the name of the moved variable as a range. For example, the following command moves the variable VAR2 after the variable VAR6:

```
move var2-var2 var6
```

```
NEW <(data-set-options)> <LIKE=data-set<(data-set-options)>>  
<FORMULA=SAS-catalog<.formula-entry>>
```

opens the FSVIEW NEW window for creating a new data set.

Note: An error message is returned if the data set that you specify in the NEW command already exists. You cannot use the NEW command to change variable names or attributes in an existing data set. Δ

In the FSVIEW NEW window you enter the names of your variables, the type, length, and (if desired) a label, format, and informat. See “Creating New SAS Data Sets” on page 127 for more information on using the FSVIEW NEW window. After you define the structure of the data set, an additional FSVIEW window is opened for entering values in the new data set.

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 SAS data set options.

You can add the LIKE= option to specify an existing data set whose variable names and attributes are copied into the FSVIEW NEW window when it is opened. The data set name in the LIKE= option can also be followed by a list of data set options.

Note: If you use the LIKE= option to specify the data set that is currently displayed in the FSVIEW window, and if that data set is opened with member-level control, you receive an error message, and the characteristics of the displayed data set are not copied into the FSVIEW NEW window. The variable names and attributes of a data set cannot be read while the data set is opened with member-level locking. To avoid this problem, you can use the MODIFY RECORD command to change the displayed data set to record-level control before issuing a NEW command with a LIKE= option that specifies the displayed data set. Δ

You can add the FORMULA= option to specify a formula to be associated with the new FSVIEW window that is opened for the data set. The rules for the *formula-name* argument are the same as in the FORMULA= option of the PROC FSVIEW statement. See the description of the FORMULA option in “PROC FSVIEW Statement” on page 94 for details.

OBS

displays the number of the observation on which the cursor is currently positioned. This is useful when observation numbers are not displayed because an ID variable is used. Type OBS on the command line, position the cursor on an observation, and press ENTER. Or position the cursor on the desired observation and press the function key to which you assigned the OBS command.

PARMS

opens the FSVIEW Parameters window, from which general parameters for the FSVIEW session can be reviewed and modified. Refer to “Customizing the FSVIEW Environment” on page 136 for more information on the parameters that can be set in the FSVIEW Parameters window.

```
PROTECT ON | OFF <variable <...variable-n>>
```

(editing command; not valid while browsing a data set)

prevents changes from being made to the values in one or more variable columns when the data set is being edited.

You must follow the PROTECT command with either the ON or OFF argument, depending on whether you want to turn the protection feature on or off for the column. You can specify which column you want the command to affect by specifying the variable name as an argument for the command, or by typing

PROTECT ON or PROTECT OFF on the command line, positioning the cursor on the desired column, and pressing ENTER.

Note: You cannot use the PROTECT OFF command to turn off protection for computed variables. △

You can control the protection feature for multiple columns with a single PROTECT command. You can provide a list of variable names, separated by spaces, as arguments for the command, or you can select a range of variable columns by specifying the names of the first and last variables in the range, separated by a dash. For example, the following command turns on protection for the variables X and Z:

```
protect on x z
```

The following command turns on protection for all variables between X and Z, inclusive:

```
protect on x-z
```

RENAME *current-variable-name new-variable-name*

changes the name for a specified variable column for the purposes of the FSVIEW window display only. The variable name in the data set is not changed; only the name that is used for the variable column in the FSVIEW window is changed. The new name is stored in a FORMULA entry rather than in the data set itself.

Like the original variable name, the new name is limited to 32 characters. All the other rules for SAS names are also applicable.

If the renamed variable is used in any formulas, the formulas are updated to reflect the new name.

RESET <ALL | *variable <... variable-n>*>

deletes some or all formula definitions. For computed variables, resetting the formula deletes the variable column from the FSVIEW window.

You can remove the formula definitions for more than one variable with a single RESET command. You can provide a list of variable names, separated by spaces, as arguments for the command, or you can select a range of variables by specifying the names of the first and last variables in the range, separated by a dash. For example, the following command resets formula definitions for the variables X and Z:

```
reset x z
```

The following command resets formula definitions for all variables between X and Z, inclusive:

```
reset x-z
```

Use the RESET ALL command to remove all current formula definitions.

REVIEW

opens the FSVIEW REVIEW window, in which all current formula definitions are displayed. See “Defining Formulas” on page 132 for more information on using the FSVIEW REVIEW window.

If no formulas are currently defined, the FSVIEW REVIEW window is not opened. Instead, a message in the FSVIEW window’s message line indicates that no formula definitions exist.

RIGHT <*n* | MAX>

scrolls the FSVIEW window to the right in increments of variable columns. The following scroll amounts can be specified:

n scrolls to the right by the specified number of variable columns.

MAX scrolls to the rightmost variable column.

If the RIGHT command is used without arguments, the FSVIEW window is scrolled by the amount that is specified in the HSCROLL parameter. The default HSCROLL amount is one column.

SAVE

stores all changes that have been made to the data set since the last time it was saved. See also the AUTOSAVE command.

SAVE FORMULA *<formula-name>*

stores all current formula definitions, general parameter settings, and FSVIEW window characteristics in a FORMULA entry.

The general form of the *formula-name* 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 an entry name in the current formula catalog. The entry type is assumed to be FORMULA. If no catalog has been previously specified in a FORMULA or SAVE FORMULA command, in the FORMULA= option of the PROC FSVIEW statement, or in the *formula-name* argument of the FSVIEW command, then the procedure stores the specified entry in your personal PROFILE catalog (SASUSER.PROFILE, or WORK.PROFILE if the SASUSER library is not allocated).
- If a two-level name is specified, the second level must be FORMULA. Any other value causes an error message. The name is treated as an entry name in the current formula catalog. If no formula catalog has been previously specified, the procedure stores the specified entry in your personal PROFILE catalog.
- If a three-level name is specified, it is treated as *libref.catalog-name.entry-name*. The entry type is assumed to be FORMULA. If the specified libref is undefined, you receive an error message. If the specified catalog does not exist, it is created.
- If a four-level name is specified, the fourth level must be FORMULA. Any other value causes an error message. You also receive an error message if the specified libref is undefined. If the specified catalog does not exist, it is created.

If you omit the *formula-name* argument altogether, the procedure stores the entry using the default name. The default formula name is the name that was used in the previous FORMULA command. If no FORMULA command has been issued, the default name is the name that was specified in the FORMULA= option of the PROC FSVIEW statement or in the *formula-name* argument of the FSVIEW command when the procedure was initiated. If no formula name was specified when the procedure was invoked, then the name of the displayed data set is used as the default FORMULA entry name. The procedure stores the default entry in the current formula catalog. If no formula catalog has previously been specified, the procedure stores the default entry in your personal PROFILE catalog.

SETWSZ <CLEAR>

records the current size and position of the FSVIEW window in the FORMULA entry. The SETWSZ CLEAR command removes any previously recorded size and position information.

SHOW ID | VAR *variable <...variable-n>*

adds dropped variables to the FSVIEW window and controls how they are displayed. Dropped variables are variables that have been excluded from the

FSVIEW window by using the DROP command, or variables that were omitted from the VAR and ID statements when the procedure was initiated.

You must follow the SHOW command with an argument that indicates whether the variables are to be displayed as ID variables or as scrolling variables. ID variables remain on the left side of the window as you scroll the other variable columns horizontally.

New scrolling variables are added to the right of existing scrolling variable columns. The first ID variable that you add replaces the observation number column. Additional ID variables are added to the right of existing ID variable columns, before the first scrolling variable column.

Note: The combined width for all ID variables cannot exceed 53 columns. If a single ID variable exceeds this length, it is truncated to 53 characters. If multiple ID variables are used, then only variables that can be displayed in 53 columns without truncation appear in the FSVIEW window. Any additional ID variables are not visible. △

You can add more than one variable with a single SHOW command. You can provide a list of variable names, separated by spaces, as arguments for the command, or you can specify a range of variable columns by specifying the names of the first and last variables in the list, separated by a dash. For example, the following command adds the variables X and Z as scrolling variables:

```
show var x z
```

The following command adds all variables between X and Z, inclusive, as scrolling variables:

```
show var x-z
```

In addition to adding variables that are currently dropped from the window, you can use the SHOW command to change current ID variables into scrolling variables and vice versa. For example, suppose that the FSVIEW window currently displays an ID variable that is named PRICE that you want to scroll with the other variables. Use the following command to redefine it as a scrolling variable:

```
show var price
```

Similarly, to change a scrolling variable into an ID variable, use the SHOW command to redefine it as an ID variable.

You can also issue the SHOW ID or SHOW VAR commands with no variables specified to open the Select Table Variables window. The Select Table Variables window provides a selection list from which the variables to be added can be chosen. See “Selecting Variables for FSVIEW Operations” on page 131 for more information on using the Select Table Variables window.

SMOOTH <ON | OFF>

turns smooth scrolling on and off, which determines how the FSVIEW window behaves when the scrollbar is used. When smooth scrolling is turned on, the contents of the FSVIEW window are refreshed as the thumb of the scrollbar is moved. When smooth scrolling is turned off, the contents of the FSVIEW window are not refreshed until the thumb of the scrollbar is released after it is moved.

When the SMOOTH command is used without the ON or OFF arguments, it acts as a toggle, turning the smooth scrolling feature on if it is currently off or off if it is currently on.

The current state of the smooth scrolling feature is recorded in the FORMULA entry when the entry is saved. The saved state takes effect when the FORMULA entry is loaded.

SORT <ASCENDING | DESCENDING> *variable* <...<ASCENDING | DESCENDING> *variable-n*>

(editing command; not valid while browsing a data set)

sorts the data set by the specified variables and saves the data set. If you have deleted any observations from the data set, the remaining observations are renumbered sequentially when you issue the SORT command.

CAUTION:

The SORT command sorts the data set itself, not just the displayed values. The FSVIEW procedure sorts a data set in place. It does not leave an unsorted copy of the original. If you need to preserve a copy of the original data, make a copy of the data set before using the SORT command.

If the KEEP= or DROP= data set options were specified with the PROC FSVIEW statement or FSVIEW command, then only the specified variables will be retained in the data set after it is sorted. Δ

Note: The SORT command is not valid while a permanent or temporary WHERE clause is in effect. Δ

By default, the procedure sorts the data set in ascending order of each specified variable. To sort a data set in descending order of a particular variable, precede the variable name with the DESCENDING option. You must specify the DESCENDING option separately for each variable. For example, the following command sorts the data in descending order by STATE and in descending order by CITY within STATE:

```
sort descending state descending city
```

You can also use the ASCENDING option to explicitly indicate the default sorting order.

If the sort fails for any reason, the FSVIEW procedure is immediately terminated in order to preserve the contents of the data set.

If your site or SAS session uses a host sort utility rather than the sort utility that is supplied by the SAS System, you may not be able to use your system's attention key to cancel a sort that is in progress.

TOP

scrolls the FSVIEW window vertically so that the first observation in the data set appears at the top of the window.

You can interrupt the TOP scroll operation before the first observation is reached. This feature is useful when you want to halt a scroll request while browsing or editing a large data set. To halt a TOP scroll operation that is in progress, press the interrupt key or key combination for your system. The FSVIEW procedure displays a requestor window that gives you the options of canceling or resuming the scrolling operation.

Note: The key or key combination that is used to interrupt an active process depends on your host operating system and terminal device. For example, some systems have a key labeled BREAK or ATTENTION (or ATTN). Other systems use a combination of the CTRL key and another key. Refer to your host documentation to determine the interrupt key or key combination for your host operating system and terminal device. Δ

UPDATE <RECORD | MEMBER>

See the MODIFY command.

VSCROLL *n* | HALF | PAGE

sets the default vertical scrolling amount for the FORWARD and BACKWARD commands. The following scroll amounts can be specified:

n scroll by the specified number of lines.

HALF scroll by half the number of lines in the window.

PAGE scroll by the full number of lines in the window.

The default amount is HALF.

WHERE <<ALSO> *expression*> | <UNDO | CLEAR>

imposes one or more sets of conditions that observations in the data set must meet in order to be displayed. *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. Observations that do not satisfy the specified conditions cannot be displayed or edited.

The complete set of conditions is called a temporary WHERE clause. The conditions can be modified or canceled during the FSVIEW session. In contrast, the WHERE statement defines a permanent WHERE clause that cannot be changed or canceled during the FSVIEW session and which is not affected by WHERE commands. See “WHERE Statement” on page 100 for details.

The word *Where...* appears in the upper-right corner of the window whenever a temporary WHERE clause is in effect.

The WHERE command has several forms:

WHERE *expression*

applies the conditions that are specified in *expression* as the new temporary WHERE clause, replacing any clause that was previously in effect.

WHERE ALSO *expression*

adds the conditions that are specified in *expression* to the existing temporary WHERE clause.

WHERE UNDO

deletes the most recently added set of conditions from the temporary WHERE clause.

WHERE

WHERE CLEAR

cancels the current temporary WHERE clause.

If you edit values in an observation so that it no longer meets the conditions of the WHERE clause, that observation can still be displayed and edited. However, a warning message is printed whenever the observation is displayed, indicating that the observation no longer meets the WHERE conditions.

If you use the ADD or DUP command to add a new observation and then enter values that do not meet the WHERE conditions, that observation cannot be displayed or edited once you scroll to another observation.

Creating New SAS Data Sets

You can use the FSVIEW procedure to create new SAS data sets. You name the variables and specify their attributes in the FSVIEW NEW window. After you exit the FSVIEW NEW window, the data set is created. It contains one observation, which is filled with missing values. An FSVIEW window is then automatically opened for editing so that you can enter values in the new data set.

Opening the FSVIEW NEW Window

You can open the FSVIEW NEW window in two ways:

- by using the NEW= option in the PROC FSVIEW statement when you invoke the procedure
- by using the NEW command after you enter the procedure.

Using the NEW= Option

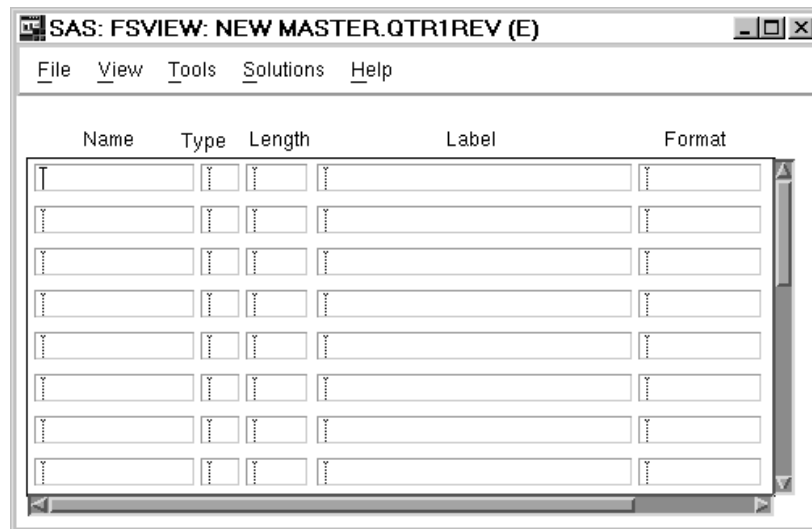
When you invoke the FSVIEW procedure with the NEW= option rather than with the DATA= option in the PROC FSVIEW statement, the FSVIEW NEW window is opened when the FSVIEW procedure is initiated.

For example, the following statements initiate an FSVIEW session and open the FSVIEW NEW window to create the new data set MASTER.QTR1REV:

```
proc fsview new=master.qtr1rev;
run;
```

Display 8.2 on page 128 shows the initial FSVIEW NEW window display.

Display 8.2 The FSVIEW NEW Window



You define the variables in the fields provided. When you issue the END command, the data set is created, and the FSVIEW window is automatically opened for editing so that you can add observations and enter variable values.

Note: After you issue the END command to close the FSVIEW NEW window, you cannot return to that window to make changes to information in the data set. Δ

Using the NEW Command

You can issue the NEW command from an FSVIEW window to define a new data set within an FSVIEW session.

For example, the following command opens the FSVIEW NEW window to define the data set MASTER.QTR1REV:

```
new master.qtr1rev
```

The display in this case is the same as shown in Display 21.2. However, when you use the END command to close the FSVIEW NEW window and create the new data set,

an additional FSVIEW window is opened. You can use that window to add observations and enter values for the new data set. The new FSVIEW window may overlay the window from which the NEW command was issued.

Creating a Data Set That Is Like an Existing One

If you want to create a new SAS data set that is identical or similar in structure to an existing data set, you can save time by letting the SAS System do some of the work. Instead of entering all the variable information, use the LIKE= option to identify an existing SAS data set. When the FSVIEW NEW window is opened, the variable names and attributes of the data set that you specified in the LIKE= option are automatically displayed.

The display contains all the information necessary to create a SAS data set that has exactly the same structure as the specified data set. (Only the structure of the specified data set is copied, not the contents.) You have the option of making changes to the variable names and attributes before creating the new data set. You can also delete a variable entirely by blanking out its name.

You open the FSVIEW NEW window in one of two ways: either with the NEW= option when you invoke the FSVIEW procedure or with the NEW command when you open an additional FSVIEW window. You can use the LIKE= option with both of these methods.

In the PROC FSVIEW statement, the LIKE= option must always be used in conjunction with the NEW= option. For example, the following statements initiate an FSVIEW session, open the FSVIEW NEW window to create the data set MASTER.QTR2REV, and fill in the fields of the FSVIEW NEW window with the names and attributes of the variables in the data set MASTER.QTR1REV:

```
proc fsview new=master.qtr2rev
    like=master.qtr1rev;
run;
```

The following command performs the same tasks, but from within an active FSVIEW session:

```
new master.qtr2rev like=master.qtr1rev
```

Closing the FSVIEW NEW Window

To close the FSVIEW NEW window, create the SAS data set, and open the FSVIEW window so that you can enter values for each observation, issue the END command. To close the FSVIEW NEW window without creating a data set, issue the CANCEL command.

FSVIEW NEW Window Commands

In addition to the global commands that are discussed in Chapter 9, “SAS/FSP Software Global Commands,” on page 141, you can use the following commands in the FSVIEW NEW window to scroll information, to duplicate selected lines, or to exit with the choice of creating a data set or canceling it.

Scrolling

```
BACKWARD <HALF | PAGE | n>
```

```
BOTTOM
```

FORWARD <HALF | PAGE | *n*>

LEFT

RIGHT

TOP

Exiting

CANCEL

END

Command Descriptions

Here are descriptions of the FSVIEW NEW window commands:

BACKWARD <*n* | HALF | PAGE>

scrolls vertically toward the top of the window. The following scroll amounts can be specified:

n scrolls upward by the specified number of lines.

HALF scrolls upward by half the number of lines in the window.

PAGE scrolls upward by the number of lines in the window.

If you do not explicitly specify an amount to scroll, the default increment is HALF.

BOTTOM

scrolls downward until the last line that contains a variable definition is displayed.

CANCEL

closes the FSVIEW NEW window and ends the FSVIEW session. The new data set is not created.

END

closes the FSVIEW NEW window, creates the SAS data set that was defined in the window, and opens an FSVIEW window for editing observations in the newly created data set.

FORWARD <*n* | HALF | PAGE>

scrolls vertically toward the bottom of the window. You can scroll forward only if you have filled the last blank variable-definition line that is currently displayed or if there are more variables to be displayed. The following scroll amounts can be specified:

n scrolls downward by the specified number of lines.

HALF scrolls downward by half the number of lines in the window.

PAGE scrolls downward by the number of lines in the window.

If you do not explicitly specify an amount to scroll, the default increment is HALF.

LEFT

displays the FORMAT column when the INFORMAT column is displayed or vice versa.

RIGHT

displays the INFORMAT column when the FORMAT column is displayed or vice versa.

TOP

scrolls upward until the first variable-definition line is displayed.

Duplicating Existing SAS Data Sets

The FSVIEW procedure enables you to create a new data set that duplicates both the structure and the contents of the data set that is currently displayed in the FSVIEW window. You use the CREATE command to create new data sets.

The new data set can incorporate some or all of the variables from the currently displayed data set. (Dropped variables cannot be used.) Computed variables in the displayed data set become data set variables in the new data set. The current computed values of the variables are stored as the values of the new variables. If a temporary or permanent WHERE clause is in effect, then only observations from the displayed data set that satisfy the WHERE conditions are copied to the new data set.

There are two ways to specify which variables in the displayed data set are used in the new data set:

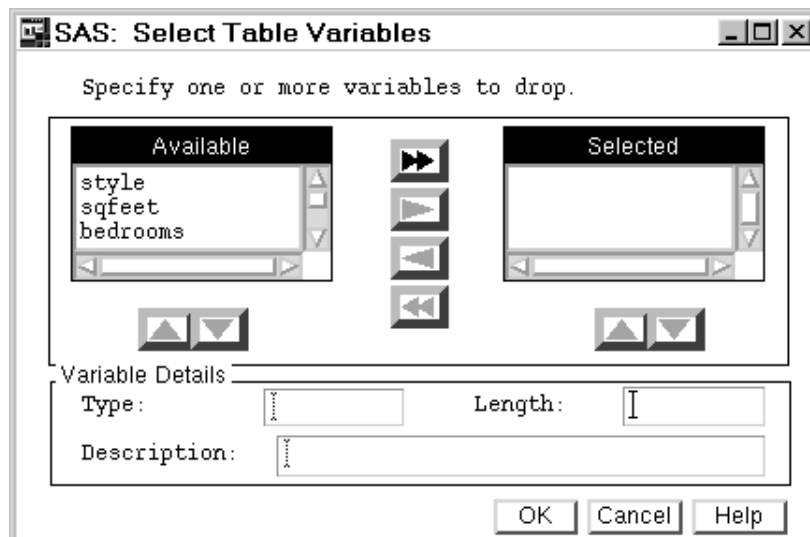
- You can list the desired variable names as arguments in the CREATE command (or use the ALL argument to copy all variables).
- You can issue a CREATE command with no variable names listed or with the ? argument. This form of the command opens the Select Table Variables window, from which you can select the variables to include in the new data set. By default, all variables in the displayed data set are selected in the Select Table Variables window. The Select Table Variables window is explained in the following section.

Selecting Variables for FSVIEW Operations

The CREATE, DROP, and SHOW commands in the FSVIEW window can open the Select Table Variables window so that you can select which variables from the displayed data set are included in the specified operation. The Select Table Variables window is a convenient alternative to entering variable names as command arguments.

Display 8.3 on page 131 shows a typical Select Table Variables window.

Display 8.3 The Select Table Variables Window



The Select Table Variables window includes lists of the available and selected variables, along with control objects for moving either individual variables or all variables between the lists. Use the single-arrow controls to move individual variables, and use the double-arrow controls to move all variables.

Below each list are control objects that you can use to change the relative position of variables within the list (for commands like SHOW where the order of variables in the list is significant). To change the order of the variables within one of the lists, select the variable you want to move. Then select the corresponding up-arrow control to move the variable toward the top of the list, or select the down-arrow control to move it toward the bottom. As an added feature, the type, size, and label of the selected variable are displayed in the Variable Details section whenever you select a variable in either list.

After you have moved all the desired variables to the **Selected** list, select **OK** to complete the operation. You can select **Cancel** to cancel the operation.

Defining Formulas

The FSVIEW procedure enables you to define formulas that perform computations and otherwise manipulate variables. *Formulas* are SAS expressions that calculate values for variables in the FSVIEW window. You can define formulas for variables in the data set, or you can create computed variables to display the formula results.

The effect of assigning a formula to a data set variable depends on whether the FSVIEW window is open for browsing or editing. If the window is open for browsing, then the result of computations performed by the assigned formula is displayed in the variable's column in the FSVIEW window, but the variable value in the data set is not affected. If the window is open for editing, then the result of the formula computations replaces the value that is stored in the data set as well as the value in the FSVIEW window.

Computed variables exist only in the FSVIEW window to show the result of computations that are performed by a formula; they are not added to the displayed data set. The values of computed variables cannot be edited. However, within the FSVIEW window computed variables can be manipulated just like data set variables. The only exception is that you cannot use the SORT command to sort the data set according to the values of a computed variable.

Use the DEFINE command to define formulas. The FSVIEW procedure provides two ways to use the DEFINE command:

- Use the DEFINE command alone or with only the *variable-name* argument to open the FSVIEW Define window. In this window you can enter all the information necessary to define the formula. You can also use this form of the command to open the FSVIEW Define window to edit an existing formula. See “Defining Formulas in the FSVIEW Define Window” on page 133 for more information on using the FSVIEW Define window.
- Supply the formula as an argument with the DEFINE command. Separate the formula from the variable name with an equal sign (=).

Formulas are stored in SAS catalog entries that have the type FORMULA. See “Creating FSVIEW Applications” on page 139 for details.

How Formulas Are Evaluated

Formulas are evaluated in the order in which the corresponding variables appear in the FSVIEW window. It is the order of the variables, not the order in which the formulas are defined, that determines the evaluation order. Formulas are evaluated

first for any ID variables, then for the remaining variables in left-to-right order. After the formulas for displayed variables are evaluated, any formulas assigned to variables that are dropped from the FSVIEW window are evaluated, in the order in which the variables are dropped.

If you use a computed variable in the formula definition for another variable, the variable that is named in the formula must appear in the FSVIEW window to the left of the variable for which the formula is defined. Otherwise, the value of the variable in the formula is missing when the formula is evaluated.

When the FSVIEW window is displayed or redrawn, the formulas are evaluated for all displayed observations. When an observation is edited, formulas are evaluated for the edited observation only.

Using SAS Component Language in Formulas

Formulas can also include any valid SAS Component Language functions or statements except those that are valid only in SAS/AF software and the following statements:

CONTROL ALWAYS	ERROROFF	PROTECT
CURSOR	ERRORON	UNPROTECT

The MODIFIED, ERROR, and CUROBS functions are valid.

Each formula that you define is treated as a block of SCL code. When you define a formula, the FSVIEW procedure internally adds a label for the formula and a RETURN statement so that each formula can be treated as a separate SCL block. (The label is the same as the variable name.) The SCL code segments are compiled individually when the formulas are defined.

For example, suppose you define a computed variable that is named MODSTAT with the following formula:

```
modstat=modified(begdate)
```

The SCL block for the MODSTAT variable would be stored as follows:

```
MODSTAT: modstat=modified(begdate);RETURN;
```

The blocks are executed in the order in which the corresponding variables appear in the FSVIEW window.

Refer to *SAS Component Language: Reference* for details about including SAS Component Language in FSVIEW formulas.

Defining Formulas in the FSVIEW Define Window

When you issue a DEFINE command in the FSVIEW window without specifying a formula for the variable, the FSVIEW Define window is opened so that you can enter the formula definition and other variable attributes. Display 8.4 on page 134 shows the FSVIEW Define window that is opened for a variable named ENDDATE:

Display 8.4 The FSVIEW Define Window

SAS: FSVIEW: Define command

Name: ENDDATE
 Type: NUMERIC CHARACTER
 Format: _____
 Informat: _____
 Label: _____

Enter the formula below:

ENDDATE = _____

Compile Ok Cancel Help

The FSVIEW Define window includes the following fields:

Name

is the name of the variable with which this formula is associated. You can change the name in this field while the FSVIEW Define window is open to change the variable for which the formula is defined. If you specify the name of a variable that has an existing formula definition, the remaining fields in this window are filled with values from the current definition.

Type

is the variable type. To select a type, position the cursor on the corresponding term and press ENTER. The default is NUMERIC.

Format

is the format that determines how the variable is displayed. The format also determines the width of the variable's column in the FSVIEW window. If you do not specify a format, a default is used:

- BEST12. for numeric variables
- \$8. for character variables.

Informat

is the informat that determines how values that are entered in the variable's column in the FSVIEW window are interpreted. If you do not specify an informat, a default is used:

- 12. for numeric variables
- \$8. for character variables.

Label

is a descriptive label up to 256 characters long. (The FSVIEW procedure does not display variable labels.)

At the bottom of the window are four lines for entering the formula. The FSVIEW Define window supplies the variable name and the equal sign that begin the definition.

Formulas that you define in the FSVIEW Define window can consist of more than one statement. If you use multiple statements, separate the statements with semicolons. The definition can include as many statements as you can fit in the space provided.

The four lines of text in the FSVIEW Define window are seen as a single line of text when the formula is compiled. Spaces are not automatically inserted between the end

of one line in the window and the beginning of the next. If you do not leave spaces between words at the line breaks, you may encounter syntax errors when the formula is compiled.

The following section describes the commands you can use in the FSVIEW Define window.

FSVIEW Define Window Commands

In addition to the global commands that are discussed in Chapter 9, “SAS/FSP Software Global Commands,” on page 141, you can use the following commands in the FSVIEW Define window:

CANCEL

closes the FSVIEW Define window without recording the formula or variable attributes that were entered in the window. Once you open the FSVIEW Define window, you must use this command if you want to close the window without entering a formula definition.

COMPILE

compiles the current formula and reports whether it is a valid SAS expression.

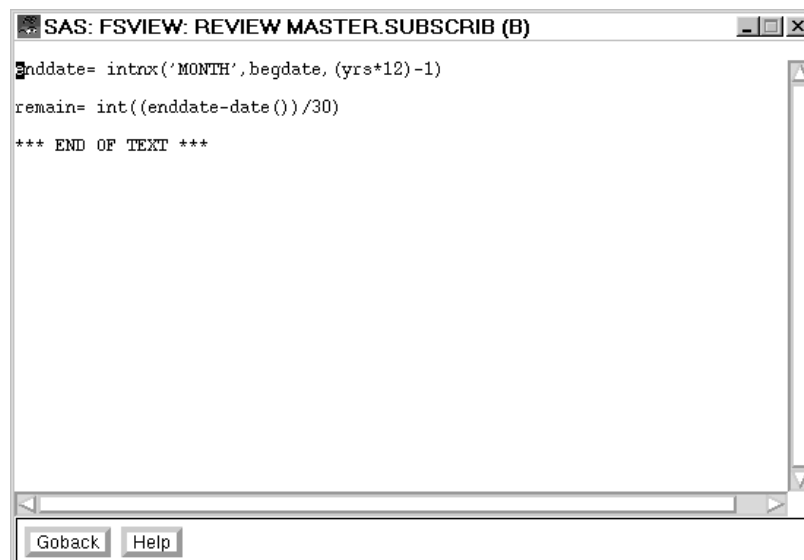
END

compiles the current formula. If the formula is valid, the FSVIEW Define window is closed, the formula definition is recorded in the FORMULA entry, and you return to the FSVIEW window. If the formula contains errors, an error message is displayed and the FSVIEW Define window remains open.

Reviewing Formula Definitions

You can browse existing formula definitions in the FSVIEW REVIEW window. Use the REVIEW command in the FSVIEW window to open the FSVIEW REVIEW window. Display 8.5 on page 135 shows a FSVIEW REVIEW window that contains two formula definitions.

Display 8.5 Viewing Formulas in the FSVIEW REVIEW Window



The FSVIEW REVIEW window displays the formulas for browsing only. You cannot make changes to the formulas in this window. The formula definitions are listed in the order in which they are entered.

FSVIEW REVIEW Window Commands

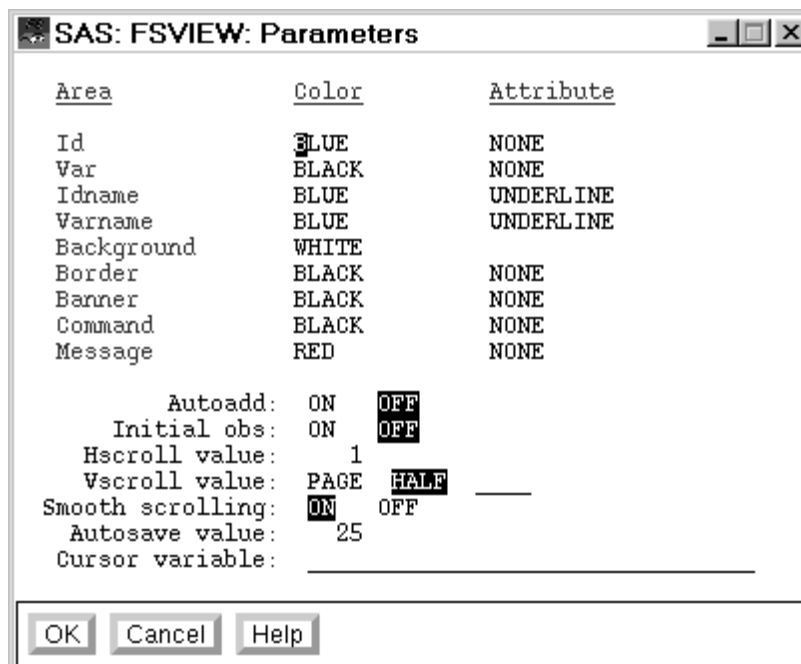
The FSVIEW REVIEW window uses the SAS text editor. All text editor browsing commands are valid in this window. Refer to the online Help for base SAS software for more information on the commands that can be used in text editor windows.

Use the END command to close the FSVIEW REVIEW window and return to the FSVIEW window.

Customizing the FSVIEW Environment

You can control some of the characteristics of the FSVIEW session by specifying values for parameters in the FSVIEW Parameters window. Use the PARMS command to open the FSVIEW Parameters window. Display 8.6 on page 136 shows the FSVIEW Parameters window with the default parameter settings.

Display 8.6 The FSVIEW Parameters Window



You can change the current parameter settings in the FSVIEW Parameters window. Some of the parameter settings in the FSVIEW Parameters window are choice groups. To turn on these parameters, move the cursor to the term for the desired parameter status and press ENTER. The selected term is highlighted. Other parameter settings are fields in which you enter the desired parameter values. The following section describes the parameters that are available in the FSVIEW Parameters window.

If you create a FORMULA entry for the FSVIEW session, all the parameter settings from the FSVIEW Parameters window are recorded and will be in effect the next time that FORMULA entry is loaded.

Parameter Descriptions

The following parameters can be controlled by using the FSVIEW Parameters windows:

Color and Attribute

specify the color and highlighting attributes of the following areas of the FSVIEW window:

ID	controls the color of observation numbers or variable values in the ID columns.
Var	controls the color of variable values in the variable columns.
IDname	controls the color of the column headings for the ID columns.
Varname	controls the color of the column headings for the variable columns.
Background	controls the background color of the FSVIEW window.
Border	controls the color of the window border in character-based display environments.

Note: This parameter has no effect in graphical windowing environments. Δ

Banner	controls the color of the Command===> text at the left of the command line.
--------	---

Note: This parameter has no effect if a menu bar is displayed in place of a command line. Δ

Command	controls the color of the text that you type on the command line.
---------	---

Note: This parameter has no effect if a menu bar is displayed in place of a command line. Δ

Message	controls the color of text that is displayed in the window's message line.
---------	--

The following values are valid in the **Color** fields:

BLUE	GREEN	WHITE	ORANGE
RED	CYAN	GRAY	
PINK	YELLOW	BROWN	

If a specified color is not available on a user's device, the procedure substitutes the available color that most closely matches the specified color. Some devices do not support changing the background color; for these devices, the background color parameter is ignored.

The following values are valid in the **Attribute** fields. (You can abbreviate the values by entering only the first letter; the procedure fills in the complete value when you press ENTER.)

HIGHLIGHT	high intensity
BLINKING	blinking
UNDERLINE	underlined
REVERSE	reverse video

If a parameter specifies a highlighting attribute that is not available on the device, the parameter is ignored. You cannot specify a highlighting attribute for the background.

If you use the COLOR command in the FSVIEW window, the colors and highlighting attributes you specify using that command appear in these parameters.

Autoadd

controls whether new observations are automatically added to the displayed data set. This parameter reflects the current status of the autoadd feature, which can also be controlled with the AUTOADD command in the FSVIEW window.

Initial obs

controls whether initial values have been recorded to fill added observations. This parameter is set on when you use the INITIAL command in the FSVIEW window. The INITIAL command records values from an existing observation so that they can be copied into new observations that are added by the autoadd feature.

Hscroll value

controls how many columns are scrolled when a LEFT or RIGHT command does not specify an explicit scroll amount. The default value is 1.

If you use the HSCROLL command in the FSVIEW window, the value that you specify with that command appears in the field for this parameter.

Vscroll value

controls how many observations are scrolled when a FORWARD or BACKWARD command does not specify an explicit scroll amount. The following default scroll amounts can be specified:

<i>n</i>	scroll by the specified number of lines. Enter the desired value in the space provided; then press ENTER to select this value. The value is highlighted in reverse video when it is selected.
HALF	scroll by half the number of lines in the window.
PAGE	scroll by the number of lines in the window.

The default value is HALF.

If you use the VSCROLL command in the FSVIEW window, the value that you specify with that command is shown in the field for this parameter.

Smooth scrolling

controls whether smooth scrolling is available. When smooth scrolling is turned on, the contents of the FSVIEW window are refreshed as the thumb of the scrollbar is moved. When smooth scrolling is turned off, the contents of the FSVIEW window are not refreshed until the thumb is released. If you use the SMOOTH command in the FSVIEW window, the scrolling status that you specify appears in this parameter.

Autosave value

controls how frequently the data set is automatically saved. An automatic save is performed whenever the number of modifications (additions, deletions, or changes)

specified in this parameter have been made since the last save. The default value is 25.

If you use the AUTOSAVE command in the FSVIEW window, the value that you specify with that command appears in the field for this parameter.

Cursor variable

controls which variable column the cursor is positioned on when a new observation is added by the autoadd feature or by the DUP command.

If you use the CURSOR command in the FSVIEW window, the variable that you specify with that command appears in the field for this parameter.

FSVIEW Parameters Window Commands

In addition to the global commands that are discussed in Chapter 9, “SAS/FSP Software Global Commands,” on page 141, you can use the following commands in the FSVIEW Parameters window:

CANCEL

closes the FSVIEW Parameters window without recording any changes to the previous parameter settings.

END

closes the FSVIEW Parameters window and applies the parameter settings that it contains to the FSVIEW window.

Creating FSVIEW Applications

If you are an applications developer, you can use the FSVIEW procedure as the basis for data entry applications and editing applications. The FSVIEW procedure enables you to customize the application environment to suit the needs of your users.

Customization can include

- redesigning the display
- defining formulas for creating computed variables or for manipulating data set variables
- setting general parameters that control the behavior of the FSVIEW session.

The feature of the FSVIEW procedure that makes this customization possible is the FORMULA entry. *Formula entries* are SAS catalog entries of type FORMULA that are created by the FSVIEW procedure to record the following information about the FSVIEW session:

- the names and order of variables in the FSVIEW window.
- any formula definitions for the variables.
- the format and informat of variables for which specific formats or informats are assigned, using the FORMAT or INFORMAT commands in the FSVIEW window.

Note: Formats and informats that are specified in the data set or by using FORMAT and INFORMAT statements in conjunction with the PROC FSVIEW statement at procedure invocation are not recorded in the FORMULA entry. △

The format and informat for computed variables are always stored in the FORMULA entry. If a FORMAT or INFORMAT command for a computed variable does not specify a valid format or informat, it is ignored.

- the current size and position of the FSVIEW window. (These values can be reset with the SETWSZ command.)

- the current FSVIEW window colors.
- the initial values for variables in added observations. (These values can be reset with the INITIAL command.)
- the current settings of all general parameters: Autoadd, Autosave value, Hscroll value, Vscroll value, and Cursor variable.

Creating and Updating Formula Entries

If you specify a catalog name or a complete formula name when you open an FSVIEW window (by using the FORMULA= option in the PROC FSVIEW statement, the *formula-name* argument in the FSVIEW command, or the FORMULA= option in the BROWSE, EDIT, and NEW commands in the FSVIEW window), a FORMULA entry is created automatically when the FSVIEW window is closed. You can also create a FORMULA entry by issuing a SAVE FORMULA command in the FSVIEW window. If you specify only a catalog name, the FSVIEW procedure uses the name of the displayed data set as the formula name.

Loading Formula Entries

To load an existing FORMULA entry when you invoke the FSVIEW procedure, use the FORMULA= option in the PROC FSVIEW statement or the *formula-name* argument in the FSVIEW command. You can also use the FORMULA= option in the BROWSE, EDIT, and NEW commands in the FSVIEW window to load a FORMULA entry for additional FSVIEW windows that are opened during the FSVIEW session. You can use the FORMULA command in an open FSVIEW window to change the FORMULA entry that is used for the window or to load a FORMULA entry if none was used when the window was originally opened. (The FORMULA command is valid even if you do not specify a formula catalog when you invoke the FSVIEW procedure.)

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.