Overview of the Report Writing Capabilities in Version 6 of SAS/FSP® Software
Patricia Lord Berryman, SAS Institute Inc., Cary, NC

ABSTRACT
This paper concentrates on the expanded report writing capabilities available with Version 6.03 of SAS/FSP® software. Each SAS/FSP procedure is discussed separately and in combination, with emphasis on ways to enhance report writing with the new available features. A discussion of the cut and paste facility and the SAS text editor (available with Base SAS® software) and how they complement the features available within the SAS/FSP procedures is included. Also discussed are the SAS/FSP global commands (for example, SPRINT, FORMNAME, and PRTFILE) that are useful in generating reports. A discussion of effective use of FORMS and ways to use your SASUSER.PROFILE catalog is included.

INTRODUCTION
Version 6 of SAS/FSP software provides many new features that make report writing and generation both easy and powerful. Two of these features, the enhanced SAS text editor combined with the new cut and paste facility, greatly increase report writing capabilities. This paper concentrates on the new and enhanced features of both base SAS software and SAS/FSP software that facilitate the writing of reports.

CUT AND PASTE FACILITY
The capability to cut and paste text within windows, across window boundaries, and from one window to another is a new feature in Release 6.03 of the SAS System. This facility, available in base SAS software, is documented in SAS Technical Report P-171, Changes and Enhancements to Base SAS Software for Personal Computers, Release 6.03.

The following is a list of the commands, with a short description, in the cut and paste facility:

- CUT copies marked text into a temporary paste buffer and deletes it from the copied window.
- MARK identifies text to be used with the CUT or STORE commands. It is also used with the FIND and CHANGE commands to identify the text you want searched.
- PASTE inserts text stored in a paste buffer into a window that uses the editing features of the SAS text editor.
- PCLEAR clears the contents of paste buffers.
- PLIST displays the names of all paste buffers in the LOG window.
- SMARK identifies an area on the display screen to be used with the STORE command. You can mark areas across window boundaries with the SMARK command.
- STORE copies marked text into a temporary paste buffer, unmarks the text, and leaves it in place.
- UNMARK returns marked text to normal status.

Where Can You Use the CUT and PASTE Commands?
You can mark and store information from all display manager windows and from all SAS/FSP and SAS/AF® windows. You can also use the CUT and PASTE commands in all SAS text editor windows that allow editing.

What is the Difference Between the MARK and SMARK Commands?
The MARK and SMARK commands are used to mark text to be used with the STORE command. The MARK command is also used with the CUT command and with the CHANGE and FIND commands to identify the area you want searched.

However, there are several differences in where and when you use the MARK and SMARK commands (see Table 1). You use the MARK command in all display manager windows and in SAS/FSP windows that allow editing with the SAS text editor. Use the SMARK command in all other SAS/FSP windows. Also, use the SMARK command when you want to span several windows or include a command line or border in the marked text. Another important difference is that the SMARK command takes a picture of the current display screen and cannot be used to mark text that appears on more than one screen. To do this, you must use the SMARK command more than once and use the STORE command with the APPEND option.

Table 1 Differences between the MARK and SMARK Commands

<table>
<thead>
<tr>
<th>MARK</th>
<th>SMARK</th>
</tr>
</thead>
<tbody>
<tr>
<td>mark text for STORE command</td>
<td>YES</td>
</tr>
<tr>
<td>mark text for CUT command</td>
<td>YES</td>
</tr>
<tr>
<td>mark text for FIND and CHANGE commands</td>
<td>YES</td>
</tr>
<tr>
<td>mark text that spans multiple windows</td>
<td>YES</td>
</tr>
<tr>
<td>mark across windows</td>
<td>NO</td>
</tr>
<tr>
<td>mark borders and command line</td>
<td>NO</td>
</tr>
<tr>
<td>mark text in HELP and MENU windows</td>
<td>NO</td>
</tr>
<tr>
<td>mark text in FSJLIST windows</td>
<td>NO</td>
</tr>
<tr>
<td>mark text in FSPRINT windows</td>
<td>NO</td>
</tr>
<tr>
<td>mark text in FSLETTER - letters</td>
<td>YES</td>
</tr>
<tr>
<td>mark text in FSLETTER - other windows</td>
<td>NO</td>
</tr>
<tr>
<td>mark text in PROC FSEDIT - data definition</td>
<td>NO</td>
</tr>
<tr>
<td>observation edit</td>
<td>NO</td>
</tr>
<tr>
<td>other windows</td>
<td>NO</td>
</tr>
<tr>
<td>mark text in PROC FSEDIT - screen customization</td>
<td>YES</td>
</tr>
<tr>
<td>program definition</td>
<td>YES</td>
</tr>
</tbody>
</table>
What about the CUT and STORE Commands?
The CUT command is available only in windows that allow editing with the SAS text editor, for example, NOTEPAD, PGM, FSLETTER, and FSEDIT screen customization and program definition windows. You can use the STORE command from any window.

SAS/FSP GLOBAL COMMANDS

The following is an overview of the SAS/FSP global printing commands that are useful in report writing. These commands are documented in the SAS/FSP Guide for Personal Computers, Version 6 Edition, and are discussed in the next sections under the individual SAS/FSP procedures.

FONT displays font information for the current form.
FORMNAME specifies form with instructions for printing pictures with the SPRINT command.
FREE releases items in print queue to a printer or closes print file.
PRRTFILE specifies file destination for printed items with the SPRINT command or letters sent from FSLETTER or FSEDIT.
SPRINT takes a picture of a window and sends it to a printer or to a file.

PROC FSLETTER: REPORT WRITING FEATURES

Important Text Editor Features for PROC FSLETTER

AUTOWRAP is a useful new feature for writing letters and reports. AUTOWRAP ON allows you to enter text continuously, without moving the cursor to the next line. It also allows you to use the INCLUDE command to bring a file with a longer line length into the text editor without truncating any text. When the INDENT command is on, you can wrap text that is indented. It can be set in the EDPARMS window and is on by default.

AUTOFLOW controls the automatic flowing of text as you bring it into the text editor with an INCLUDE or PASTE command. When text is flowed, the left and right boundaries are determined by previous executions of the INDENT and BOUNDS commands.

BOUNDS identifies the left and right boundaries of the section of the window that you want affected when text is flowed, by either the TF (text flow) line command or the AUTOFLOW command.

Searches within Marked Areas is a new feature with Version 6 and provides the ability to use the FIND and CHANGE commands within a marked area. You can restrict the area of text being searched in the FIND and CHANGE commands.

FILE and INCLUDE from external files allow for the use of the FILE and INCLUDE commands to write and read text to and from external files.

In addition, several new line commands allow you to change the case of text easily. The commands are CL (CCL) for changing all characters to lowercase and CU (CCU) for changing all characters to uppercase.

Several new line commands allow you to left justify, right justify, or center a single line or a block of text. The commands are JC (JJC) justify center, JR (JJR) justify right, and JL (JJL) justify left.

The TABS line command allows you to indicate tab settings. Tab settings are set in the EDPARMS window or with the line command. Individual settings for a letter are saved in the EDPARMS window and are available for the next editing session.

The Text Flow (TF) and Text Split (TS) line commands are very useful when writing letters. They are best used as function keys. Text Flow flows a paragraph by removing trailing blanks from each line until the next blank line. Text Split moves text following the current cursor position to another line and allows room to insert new text.

The EDPARMS Window

Default Parameters for the SAS Text Editor The appearance and behavior of the SAS text editor within FSLETTER (that is, its color and editing options) are determined by parameters stored in a screen called an EDPARMS window. Individual letters also have their own settings that are saved in an individual EDPARMS window stored within the letter entry. The default parameter settings in the EDPARMS window for PROC FSLETTER are shown in Screen 1.

**Screen 1** Default Setting: FSLETTER.EDPARMS

The default parameters for PROC FSLETTER are as follows:

- **AUTOADD ON** causes a line to be added automatically as you scroll forward.
- **AUTOWRAP ON** causes a word to be moved to the next line automatically if you type past the end of the line.
- **INDENT ON** retains indentation when text is flowed.
- **CAPS OFF** keeps text from being automatically capitalized; that is, it is left as you enter it.
- **NUMS OFF** suppresses display of line numbers.
- **TABS** suppresses tabs.
If you are happy with the default settings, you do not need to worry about the EDPARMS window. Each letter you create uses the default settings, and you never need to edit an EDPARMS window.

Changing the Default Parameters for New Letters. The default settings for all letters you create can be changed by editing FSLETTER.EDPARMS and making the changes. Then each new letter gets its default settings from your custom EDPARMS entry. Once FSLETTER.EDPARMS is edited, it is listed in your catalog directory screen. Note that changing FSLETTER.EDPARMS does not affect any letters created previously. Also, you can still have individual letters with their own settings.

You may find it convenient to create different FSLETTER.EDPARMS entries for different catalogs. If you keep letters in one catalog, reports in another, and questionnaires in another, appropriate parameter settings can be stored for each kind of text in each catalog. You can also have one default EDPARMS entry in your SASUSER.PROFILE catalog. If no EDPARMS entry exists in a catalog, any new letters use the settings in your SASUSER.PROFILE and not the default system settings. Keeping a customized default EDPARMS screen in SASUSER.PROFILE saves time and effort if you use many different catalogs for editing your letters.

Changing Parameters for Individual Letters. If you want to change the text editor environment for an individual letter, execute the EDPARMS command within the letter. The EDPARMS window overlays the text editor and shows the parameters in effect for that LETTER entry. Make any changes and press END to return to your letter. The EDPARMS entry is associated only with the letter displayed, and any changes apply only to that letter.

Another way to change the text editor settings within a letter is to use regular editing commands. For example, if you change the background color to blue or add tab settings and then use the EDPARMS command to view the EDPARMS window, the new setting is listed. Changes made with regular editing commands are saved automatically in the EDPARMS entry associated with that particular letter. See Screen 2, which shows an individual EDPARMS window.

Screen 2 Changing Parameters for Individual Letters

Use of Display Manager Windows within PROC FSLETTER

An important enhancement in Version 6 is the ability to jump to various windows from within any full-screen procedure. For example, while editing a letter, you can display any display manager window without leaving the editing session. Any text within the SAS LOG or OUTPUT windows can easily be cut and pasted within the letter you are editing. To display any display manager window, simply enter its name on the command line (or use a PF key set to the window name). To return to the full-screen window, use the NEXT command followed by the window name. For example, to return from the LOG window to the LETTER entry, issue the command NEXT FS. (Note: NEXT FS returns the active SAS/FSP procedure window.)

Other windows that are useful to view or to store/paste information about data set or catalog entries within letters are the CAT, DIR, and VAR windows. The NOTEPAD window is used to store any information you find useful while writing a report (for example, notes from a meeting or definitions for technical terms). The NOTEPAD window is displayed whenever you want. You can also arrange your letter and NOTEPAD window (using window management commands, for example, the WSHRINK command) such that the NOTEPAD is available for viewing at all times. Screen 3 shows such a window arrangement.

Screen 3 Use of NOTEPAD Window in PROC FSLETTER

Effective Use of FORMS

A FORM entry allows you to store specifications for printing output created with any SAS/FSP procedure. These entries are created in PROC FSLETTER or in PROC FSEDIT with the LETTER=n option specified. The format of your printed text (for example, which text is to be underlined and what margin settings are used) is specified within the FORM entry. This information is used when you send a letter from the FSLETTER screen or the FSEDIT catalog directory screen of PROC FSLETTER or from within a LETTER entry by issuing the command:

```
edit name.form
```

There are several panels that make up the FORM entry. The Printer Selection panel allows you to select your printer from a list of default printers. The Text Body and Margin Information panel allows you to specify page formats and margin information. The Print File Parameters panel allows you to specify print file information for the operating system the letters are printed on.

FSLETTER.FORM. A new feature in Version 6 is that all letter entries are assigned a default form when created; this form is FSLETTER.FORM. If you have a particular printer you use for most of your letters, you should edit FSLETTER.FORM and assign it to your printer. Then each letter you create uses that
form by default. You can place that form in your SASUSER.PROFILE where it will be used by default in any letter you create (unless you have a FSLETTER.FORM for that catalog). Using a customized FSLETTER.FORM eliminates the need to assign a form to each letter.

The Font Control Information Panel  This panel is new in Version 6 of SAS/FSP software. This panel allows you to use color and highlight attributes to tell your printer how to print your text. For example, text entered in magenta may tell your printer to underline, while highlighted text may signify to bold face the printed text.

The SAS System supplies default attribute definitions with the FORM in the previous screen. Once a form is assigned to a LETTER (either as FSLETTER.FORM or with the FORM command) the FONT command is issued to display the colors and attributes used for that letter. In Screen 5, the FONT screen displays the setting from the FORM in the previous screen.

Using Color to Signal Printing Instructions  With the ability to signal printing instructions by entering text with specified colors and highlight attributes, the color commands available in the SAS text editor become very important. In particular, the COLOR MAP and COLOR MTEXT commands are used to enhance reports. For example, to change all section headings in a letter to underline, mark all headings, and issue the following command:

```
   COLOR MAP 'from color' 'to color'
```

where 'from color' is original color and 'to color' is the color and highlight attribute used to signify underline text. Now all headings are underlined when printed. You can also use the COLOR MAP command when you want to change from one color to another for use in printing.

Printer Control Language Panel  The Printer Control Language panel is useful if you need to change the font, page size, font size, or other tasks that you cannot specify on the font control screen. Default values are given for all supported printers that require it.

SAS/FSP Global Commands and PROC FSLETTER  The PRTFILE command allows you to send individual letters in one FSLETTER session to multiple print files. This is particularly important for users that either do not have a printer attached or who want to route their letters to different printers. This feature is new in Version 6. Now, instead of requiring that you specify the PRTFILE option in the PROC statement, you can specify it within the letter session.

The FONT command allows you to view, at any time in your letter session, the color and highlight attributes of your form.

Use of Fields to Personalize Your Letters  Without PROC FSLETTER you can enter the text of your letter once and then produce an unlimited number of individual letters by entering the information that differs from letter to letter in special fields. Fields are areas where you put in the information that changes from one letter to the next, for example, name and address. This feature has not changed from Version 5.

Sending Reports with Values from SAS Data Sets  There are several methods of sending letters for observations in a SAS data set. The method you choose depends on which is most convenient for you at a particular time.

New Features in PROC FSLETTER for Version 6  You can send letters from PROC FSLETTER in batch mode. Use the DATA= option and specify the letter with the LETTER= option in the PROC FSLETTER statement as shown below:

```
   proc fsletter lett=ag.let.data.letter data=ag.data; run;
```

You can also send letters interactively using a SAS data set without using the FSEDIT procedure. From the letter or the catalog directory screen, use the SEND command with the DATA= option. A letter is sent with each observation in the SAS data set.

PROC FSEDIT: REPORT WRITING FEATURES  Use PROC FSEDIT with PROC FSLETTER to customize reports with information from a SAS data set. You can also design a customized display screen that can be printed to produce reports.
Cut and Paste Facility and SAS/FSP Global Commands

An easy way to get a printed copy of an observation using PROC FSEDIT is by using the SAS/FSP global commands. The SPRINT command allows you to print a quick copy of the FSEDIT screen. Use the FORMNAME command in order to use a particular form for printing the screen. (Note: the form must exist in the SCREEN= catalog, in your SASUSER.PROFILE catalog, or in the default SASHELP catalog.) You can also use the SMARK command to put contents of the screen into a paste buffer.

PROC FSEDIT Combined with the LETTER= Option

By combining the features of the FSEDIT and FSLETTER procedures, you can reduce the time and effort to produce reports. For example, suppose you are adding entries in a SAS data set with PROC FSEDIT and want to get a report of the values. By specifying the LETTER= option when you enter PROC FSEDIT, you have all the features of PROC FSLETTER ready to create your report. The following statement, when submitted to the SAS System, displays Screen 6.

```
PROC FSEDIT DATA=my.data SCREEN=my.report LETTER=my.report;
RUN;
```

Screen 6 PROC FSEDIT with Modified Screen

To go to the FSLETTER catalog directory screen use the LETTER command (in Version 5 you used the CAT command, which in Version 6 displays the CATALOG window). Now, edit a new letter and use field attributes for the value in the SAS data set (see Screen 7). If you forget the names of the variables, just display the VAR window; you can also cut and paste information from the VAR window. If you want the letter to look like the FSEDIT screen, use the SMARK command to mark the portion of the FSEDIT screen you want and paste it into the letter.

Screen 7 Letter with Fields from Data Set

Add any information you want, and then issue the SEND command and the values from the SAS data set are automatically filled in for you (see Screen 8).

Screen 8 Letter with Fields Filled

Now return to the FSEDIT screen. For the next observation you want a report on, you need to issue the SEND command from the FSEDIT screen and you are automatically in the letter you specified with the values filled in.

PROC FSPRINT: REPORT WRITING FEATURES

The FSPRINT procedure has many new features in Version 6 that make it an excellent report generator. Now that you have so much control over the display in the FSPRINT procedure, you can take advantage of it as a report generator. Use FSPRINT commands to display the variables you want, in the order you want. Then you can use the global SPRINT command to print a report or the global cut and paste facility to construct a report.

Highlight on New Features

The FSPRINT procedure for Version 6 is no longer just a browsing procedure; it has all the features of a full interactive editing environment. You can create a new data set, sort values in a data set, and add and delete observations. In addition to the new editing features, you also have a full range of commands to alter the
display of the variables on the screen. This makes the FSPRINT procedure a valuable interactive report generator.

Changing the Display of Variables

All the commands listed below, with the exception of the SORT command, affect only the display of the variables, not the actual data set. These commands are very useful in creating reports with the variables in just the configuration you want.

**COLOR fieldtype**

sets the color and highlight attributes of different fields in the window. You can change the variables values (fieldtype VAR), values of ID variables (fieldtype ID), variable names (fieldtype VARNAME), and ID variable names (fieldtype IDNAME). The color commands are very useful when used in combination with the SAS/FSP global commands FORMNAME and SPRINT (see below).

**DROP variables**

drops the variable from being displayed. To redisplay a dropped variable, use the SHOW command.

**FORMAT variables**

changes the format of a variable for display purposes only. It does not alter the format stored with the variable in the data set.

**MOVE variables**

moves the display of a single variable column or a range of variable columns.

**SHOW VAR/ID variables**

controls which variables are displayed and how they are displayed. You can display variables that were previously excluded or dropped from the display. You can also use the SHOW command to redefine a variable as either a VAR or ID variable.

**SORT variables**

or **DESCENDING variables**

sorts the data set by the specified variables and saves the data set.

**Use of the SPRINT Command in PROC FSPRINT**

The following screens show an example session in PROC FSPRINT using various commands to format the display and change colors to create a finished report.

**Screen 9 Initial FSPRINT Screen**

First, drop the variables that are not needed for the report. Drop the variable NUMBER. Then make DATE an ID variable by using the SHOW command (see Screen 10).

**Screen 10 Using the SHOW ID Command**

Now change the format (only display is changed, format in data set remains the same) of several variables.

```
COLOR IDNAME VARNAME MAGENTA
COLOR VAR GREEN
COLOR ID YELLOW
```

You can also use the CMENU commands to change colors as shown in Screen 11.
GLOBAL FSPRINT SCREEN
Sets FSPRINT specific colors and attributes.

<table>
<thead>
<tr>
<th>DATE</th>
<th>DESC</th>
<th>ACDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 24, 1988</td>
<td>Belk Legget clothes</td>
<td></td>
</tr>
<tr>
<td>January 21, 1988</td>
<td>Advance Auto car main</td>
<td></td>
</tr>
</tbody>
</table>

Valid fields:
- ID, Ver, Name, Account
- Valid colors:
  - Blue, White, Red, Black, Green, Magenta, Cyan, Gray, Pink, Brown, Yellow, Orange, Real
- Valid attributes: (optional)
  - Highlight, Blinking, Underline, Reverse

Field Color Attribute

Screen 11 Use of CMENU and COLOR Commands

Now assign the form to the FSPRINT session with the FORMNAME command, and check to see if the colors are correct with the FONT command (see Screen 12).

Screen 12 FSPRINT and FONT Windows

Finally, use the SPRINT command with the NOCMD and NOBORDER options to print your report (see Screen 13).

Screen 13 Final Report Using PROC FSPRINT and the SPRINT Command

Cut and Paste Facility and PROC FSPRINT

Another way you can use FSPRINT for reports is to use the cut and paste facility. Remember, you must use the SMARK command (the MARK command will not work) and store the desired screen to a paste buffer. Then, paste the report into a letter to generate your reports.

PROC FSLIST: REPORT WRITING FEATURES

The FSLIST procedure displays an external file in full-screen browse mode. All the SAS text editor commands that apply to a browse window are available. Use the SPRINT command to print a copy of the display screen, or use the SMARK command to mark the area of text for insertion in an editing window, for example, within a letter.

CONCLUSION

Version 6 SAS/FSP software contains many new features that enhance report writing. This, in combination with the many new features in base SAS software, such as the cut and paste facility and the enhanced SAS text editor, make reports easy and fun to produce. The following is a summary of the new features discussed in this paper:

- A cut and paste facility that allows you to capture text in any display manager or SAS/FSP procedure window and insert it in any window that uses the SAS text editor for editing.
- An enhanced version of the SAS text editor.
- New printing features in SAS/FSP procedures, including taking pictures of windows and routing them directly to a printer or to a print file, using instructions for printing that you specify and store in a form.
• Ability to send a letter for each observation in the data set from the FSLETTER procedure either interactively or in batch.

• Enhanced features in the FSPRINT procedure that give you control over the display of the data set and allow you to create reports interactively.

• Forms that allow you to take advantage of the special features of your printer.

• Ability to set text editor parameters that control the environment of your letter sessions for an entire catalog, as well as to store editing parameters for each individual letter.

SAS, SAS/AF, and SAS/FSP are registered trademarks of SAS Institute Inc., Cary, NC, USA.

IBM is a registered trademark of International Business Machines.