

## Paper 73-25

**Controlling Print Setup Options from within a Windows® SAS® Program**

Delayne Stokke, GuideOne Insurance, West Des Moines, IA

**ABSTRACT**

This paper discusses the issues a SAS® programmer needs to consider when creating printed output from a SAS program in the Windows® environment. Background information regarding how PAGESIZE and LINESIZE options affect your printed output is included, along with a discussion of how these options relate to the controls accessed through the Print Setup dialog. A macro approach to controlling and coordinating print setup – including LINESIZE and PAGESIZE settings, font selection, and font size – is introduced.

This paper is intended for SAS programmers who are working on a Windows platform. A basic understanding of SAS Macro Language is required. Base SAS statements, Display Manager (DM) commands, and Macro Language statements are used.

The ideas presented here were developed and tested using SAS Release 6.12 TS Level 0020 as well as SAS Release 8.00 TS Level 00D1 running under Windows NT version 4.0.

**INTRODUCTION**

I've written many programs in SAS to produce reports. One of the problems I've had is keeping track of the appropriate Print Setup for each report. In the past, I've documented the appropriate Print Setup in program comments, but sometimes I've failed to change the Print Setup before running my report. When I didn't get the results that I wanted, I had to change the Print Setup, and re-run the report. Recently I decided that there must be a better way. I've researched ways to control the Print Setup from inside a SAS program, and have condensed my findings into a SAS macro that's called %SETPRINT. Using this macro, I can control the page orientation, the font size, and the font used for my printed output. This macro helps to ensure consistent results every time I run a report.

**CONTROLLING PAGESIZE & LINESIZE****THE OPTIONS STATEMENT**

Most SAS programmers have probably used the LINESIZE and PAGESIZE options at one time or another. PAGESIZE specifies the number of lines that can be printed per page of SAS output (page length) while LINESIZE controls the line width. Specifically, these options control the way print image outputs are formatted. By "print image output", I mean any output that is directed to the LOG or OUTPUT windows. LINESIZE affects output directed to both the LOG and OUTPUT windows, while PAGESIZE only affects output directed to the OUTPUT window. The most important thing to note here is that these controls only affect the way the output is formatted and displayed in the LOG or OUTPUT windows – they do NOT affect the print setup options at all.

**THE PRINT & PAGE SETUP DIALOGS**

The Print & Page Setup dialogs allow you to interactively control the printer settings. Within these dialogs, you can:

- select a font, from a list of available fonts
- set the font size
- set the page orientation
- select a printer, set page margins, and change other settings

When you interactively change any settings through the SETUP

dialogs, SAS automatically re-calculates the LINESIZE and PAGESIZE values.

When SAS prepares output for the OUTPUT window, it uses the LINESIZE and PAGESIZE options that are currently in effect to format the pages. The current settings of these options could have been set explicitly through an OPTIONS statement, or they could have been set implicitly through the SETUP dialogs – whichever instruction was most recently received.

**WARNING:**

If the LINESIZE and PAGESIZE options that were in effect at the time the report was created have values greater than the LINESIZE and PAGESIZE calculated by the SETUP dialogs, there may be truncation or overflow of the printed report.

On the Windows platform, I recommend the following process for creating reports:

- change the printer setup BEFORE the program creates the report
- DO NOT include an OPTIONS statement with explicit LINESIZE and PAGESIZE settings in your program

**A PROGRAM STATEMENT TO SET THE FONT AND FONT SIZE**

As discussed above, you can interactively select a font and set the font size using the Print Setup dialog. You can also do this by using the SYSPRINTFONT= option. This option can be coded on an OPTIONS statement as follows:

```
SYSPRINTFONT='fontname' <pointsize>
```

Where...

*fontname* is the name of the font to use for printing. This must be a valid, case-sensitive font face name (for example, 'SAS Monospace' or 'Courier'). This is a required argument.

And...

*pointsize* is the base point size to use for printing. This must be an integer from 1 to 7200, inclusive. If you omit this argument, SAS uses 10 points by default. Throughout this paper, I refer to this option with the term "font size" (or "fontsize").

This option is valid for releases 6.12 and 8.00.

**PROGRAM STATEMENTS TO CHANGE THE PAGE ORIENTATION****RELEASE 8.00 – IT'S EASY!**

In SAS 8.00, you can specify the desired page orientation on an OPTIONS statement. The syntax for this option is:

```
ORIENTATION='orient'
```

Where...

*orient* is the desired orientation, either PORTRAIT or LANDSCAPE

**RELEASE 6.12 – ALMOST AS EASY!**

The SAS Display Manager for Windows provides the DLGPRTSETUP command to open the Print Setup

dialog. The syntax for this command is:

```
DM 'DLGPRTSETUP ORIENT=orient NODISPLAY';
```

Where...

*orient* is the desired orientation, either PORTRAIT or LANDSCAPE

'NODISPLAY' coded on this command prevents the Print Setup Dialog window from being displayed.

## CHECKING FOR THE CURRENT PAGE ORIENTATION

In some cases, I don't want to change the page orientation; I just want to know what it is.

### RELEASE 8.00 – IT'S EASY!

In SAS 8.00, you can determine the current page orientation using the GETOPTION function. For example, the statement:

```
ORIENT = GETOPTION('ORIENTATION');
```

...returns the current page orientation to the variable named ORIENT.

### RELEASE 6.12 – NOT SO EASY!

SAS 6.12 doesn't provide a command that will return the value of the current page orientation. But I discovered a "feature" that would allow me to figure it out. Through much experimentation I discovered that in 6.12, the LINESIZE and PAGESIZE are NOT recalculated unless the page orientation changes. This is true even if you change the font and/or font size. Here's how I take advantage of this to determine the current page orientation:

- Use an OPTIONS statement to set the LINESIZE and PAGESIZE to their maximum possible values.
- Use a DM command to set the Page Orientation to PORTRAIT.
- Query the system to determine the current LINESIZE and PAGESIZE.
- If the current LINESIZE and PAGESIZE are still equal to their maximum values, the beginning Page Orientation is PORTRAIT; otherwise, the beginning Page Orientation is LANDSCAPE.

Sample code to illustrate the method follows:

```
%macro morient;
  /* Set LINE & PAGE SIZE to Maximums */
  options linesize=MAX pagesize=MAX;
  /* Set Orientation to PORTRAIT */
  dm 'dlgpertsetup orient=PORTRAIT nodisplay';
  /* Get Current LINESIZE & PAGESIZE */
  %let linesize =
    %sysfunc(getoption(linesize));
  %let pagesize =
    %sysfunc(getoption(pagesize));
  /* Are they still at the MAX? */
  %if &linesize = MAX AND
    &pagesize = MAX %then %do;
    /* If so, Orientation is PORTRAIT */
    %let orient = PORTRAIT;
  %end;
  /* If not, orientation is LANDSCAPE */
  %else %do;
    %let orient = LANDSCAPE;
  %end;
  %put Beginning Orientation was: &orient.;
%mend morient;
%orient
```

## CONTROLLING THE MARGINS

SAS 8.00 provides the options LEFTMARGIN, RIGHTMARGIN, TOPMARGIN, and BOTTOMMARGIN to control the indicated margins. Just specify the desired margins in decimal inches. For example:

```
OPTIONS LEFTMARGIN=0.5 RIGHTMARGIN=0.5;
```

You can't change margins with program statements in SAS 6.12. You must activate the Print Setup dialog and then open the Page Setup dialog. I've not found this to be a big problem, since most of the time I leave the margins all set to their defaults. However, there are times when it would be nice to be able to control the margins from within a program.

## THE "SETPRINT" MACRO

I've compiled what I know about using SAS program statements to control the print setup into an easy-to-use macro. The macro is named "SETPRINT.SAS" and accepts up to 3 positional parameters:

- "orient" is the first parameter, and it is used to specify the page orientation. Possible values are PORTRAIT or LANDSCAPE. If not specified, the macro will determine the current page orientation.

For Version 8.00 this parameter defines the setting used by the ORIENTATION option.

For Version 6.12 this parameter is the same as the ORIENT= parameter used by the DLGPRTSETUP command.

This parameter is optional.

- "fontsize" is the second parameter, and represents the base point size to use for printing. This must be an integer from 1 to 7200, inclusive.

This parameter is the same as the "pointsize" parameter used by the SYSPRINTFONT option. However, the default behavior for the SETPRINT Macro is different. As used by SETPRINT, this parameter has no default value, and can be specified without specifying a "font". In that case SETPRINT uses "SAS Monospace" as the default font.

This parameter is optional, however, it becomes **required** if a font is specified using the "font" parameter (described below).

- "font" is the third parameter, and is the name of the font to use for printing. This must be a valid, case-sensitive font face name (for example, 'SAS Monospace' or 'Courier').

This parameter is the same as the "fontname" parameter used by the SYSPRINTFONT option. Again, the default behavior for SETPRINT is different than the SAS option. As used by SETPRINT, this parameter has no default value, unless a "fontsize" has been specified. In that case, "font" will take on a default value of "SAS Monospace".

This parameter is optional.

Additionally, SETPRINT will accept up to 4 optional keyword parameters. Use the following parameters to set the respective margins in decimal inches when running under SAS 8.00.

- "left=" sets the LEFTMARGIN
- "right=" sets the RIGHTMARGIN
- "top=" sets the TOPMARGIN
- "bottom=" sets the BOTTOMMARGIN

### SETPRINT DESIGN CONSIDERATIONS

In designing the macro, I changed the default behavior of the "fontsize" and "font" parameters to suit my purposes. I found that for most reports, I wanted to control the page orientation and the font size. I also reasoned that "SAS Monospace" should be available on any Windows system running SAS, and is almost always an acceptable choice for a report. For those reasons, I chose to make it the default.

When invoked, SETPRINT will make the specified Print Setup changes, and then write a message box to the SAS Log to display the current print setup specifications. See Figure 1.

In addition to being able to change print setup specifications, I also wanted the macro to be able to report on the current print setup – without changing any values. For that reason, none of the parameters have default values. If no parameters are specified, the macro will simply write the Print Setup message to the SAS Log. Since there is no way to determine the currently selected font or font size, those items are listed as "Unknown". See Figure 2.

### CONCLUSION: SETPRINT USAGE

On my system, I've put the SETPRINT.SAS source code into the :\SAS\CORE\SASMACRO folder. This is a system folder that is defined as the default macro autocall library. By placing SETPRINT there, it is always available within my SAS session.

The SETPRINT macro is easy to use. During the process of developing a report, I determine the print setup that is appropriate. If the report is best formatted using a Portrait orientation with a 10 Point SAS Monospace font, I just include the following line in my program, somewhere before the report is created:

```
%setprint (Portrait, 10)
```

Specifying an orientation and a font size is how I usually use the macro. But if I wanted to change the font and the font size

without changing the orientation, I could include the following macro call:

```
%setprint (font=Courier, fontsize=8)
```

Figure 3 contains the entire source code for the SETPRINT macro. I've tried to include sufficient comments to clearly describe the operation of the macro.

### REFERENCES

*SAS Language: Reference*, Version 6, First Edition, Cary, NC: SAS Institute Inc.

*The SAS System for Windows*, release 6.12, *SAS System Help*, Cary, NC: SAS Institute Inc.

*The SAS System for Windows*, release 8.00, *SAS System Help*, Cary, NC: SAS Institute Inc.

*The SAS System for Windows*, release 6.12, *SAS Companion for Microsoft Windows*, Cary, NC: SAS Institute Inc.

The SAS Institute Web Site,

<http://www.sas.com/service/doc/v8/whatsnew/html/zenid-30.htm>

Cary, NC: SAS Institute Inc.

### CONTACT INFORMATION

If you have any questions or comments about SETPRINT.SAS, please contact the author at:

Delayne Stokke  
 GuideOne Insurance  
 1111 Ashworth Road  
 West Des Moines, IA 50265-3538  
 Work Phone: (515) 267-2212  
 Email: dstokke@guidemail.com or  
 dstokke@aol.com

```
232 %setprint(Portrait,10)
NOTE: *-----*
NOTE: *   P r i n t   S e t u p   S p e c i f i c a t i o n s   *
NOTE: *-----*
NOTE:      Page Orientation:                PORTRAIT
NOTE:      Line size for printed output:    95
NOTE:      Number of lines printed per page of output: 54
NOTE:      Font selected for printed output: SAS Monospace
NOTE:      Font size selected for printed output: 10
NOTE: *-----*
```

Figure 1. SETPRINT.SAS Print Setup Specifications message from macro invocation that includes a page orientation and a font size specification.

```
234 %setprint()
NOTE: *-----*
NOTE: *   P r i n t   S e t u p   S p e c i f i c a t i o n s   *
NOTE: *-----*
NOTE:      Page Orientation:                LANDSCAPE
NOTE:      Line size for printed output:    127
NOTE:      Number of lines printed per page of output: 41
NOTE:      Font selected for printed output: Unknown
NOTE:      Font size selected for printed output: Unknown
NOTE: *-----*
```

Figure 2. SETPRINT.SAS Print Setup Specifications message from macro invocation that does not include any parameters.

```

/*-----*/
* Macro Name: SETPRINT.SAS
* Purpose: Sets Printer Orientation, Fontsize, & Font.
* Input Parameters: (7)
* Parameter required/
* Name optional Parameter Description
* 1) orient, optional Paper Orientation.
* PORTAIT, or
* LANDSCAPE,
* 2) fontsize, optional Font Size in points.
* Can be any value that is valid
* for the font specified on the
* font= parameter.
* REQUIRED if a font is specified with the
* "font" parameter.
* -----
* The following parameter can usually be left
* to the default value.
* -----
* 3) font, optional Name of Font. SAS Monospace
* will be used if a fontsize is
* specified, but a font is not.
* -----
* The following KEYWORD parameters will only
* work under SAS Version 8.00 or higher.
* -----
* 4) left=, optional Left Margin in inches
* 5) right=, optional Right Margin in inches
* 6) top=, optional Top Margin in inches
* 7) bottom= optional Bottom Margin in inches
* Example Usage: %SETPRINT(PORTAIT,10)
* Date: July 14, 1999
* Modified: October 28, 1999
* Enhanced to work with Version 8.00
* Programmer: Delayne Stokke
* -----*/

%macro setprint(orient,
               fontsize,
               font,
               left=,
               right=,
               top=,
               bottom=);
/*-----*/
* Explicitly define macro variables to be LOCAL. This is done *
* to ensure that if there are GLOBAL macro variables of the *
* same name, the GLOBAL variables will not be changed. *
* (Note that Macro parameters are ALWAYS LOCAL.) *
*-----*/
%local linesize pagesize;

/* Fontsize specified: set font and fontsize */
%if %datatype(&fontsize) = NUMERIC %then %do;
%if %length(&font.) = 0 %then %let font = SAS Monospace;
options sysprintfont = "&font." &fontsize.;
%end;

/* Fontsize NOT specified */
%else %do;
%put ERROR: *--- ERROR --- ERROR --- ERROR --- ERROR ---*;
%put ERROR;;
%put ERROR: To specify a Font, you must first specify a font size!;
%put ERROR;;
%put ERROR: Example Usage: %nrstr(%SETPRINT)(&orient,&font.);
%put ERROR;;
%put ERROR: *--- ERROR --- ERROR --- ERROR --- ERROR ---*;
%end;

%let font = Unknown;
%let fontsize = Unknown;
%end;

%if &sysver >= 8.00 %then %do;
/* Orientation specified : set orientation */
%if (%upcase(&orient) = PORTRAIT OR
%upcase(&orient) = LANDSCAPE) %then %do;
options orientation=%upcase(&orient);
%end;
/* Orientation NOT specified : Get orientation */
%else %let orient = %sysfunc(getoption(orientation));
%end;

```

SETPRINT Macro: Source Code. Figure 3, continued on next page.

```

/* Check for Margin Settings */
%if %datatyp(&left) = NUMERIC %then options leftmargin=&left.;
%else %let left = %sysfunc(getoption(leftmargin));
%if &left = %then %let left = Unknown;

%if %datatyp(&right) = NUMERIC %then options rightmargin=&right.;
%else %let right = %sysfunc(getoption(rightmargin));
%if &right = %then %let right = Unknown;

%if %datatyp(&top) = NUMERIC %then options topmargin=&top.;
%else %let top = %sysfunc(getoption(topmargin));
%if &top = %then %let top = Unknown;

%if %datatyp(&bottom) = NUMERIC %then options bottommargin=&bottom.;
%else %let bottom = %sysfunc(getoption(bottommargin));
%if &bottom = %then %let bottom = Unknown;

/* Get LINESIZE and PAGESIZE */
%let linesize = %sysfunc(getoption(linesize));
%let pagesize = %sysfunc(getoption(pagesize));

%end;

%else %do; /* NOT Version 8.00 */

/* Orientation specified : set orientation */
%if (%upcase(&orient) = PORTRAIT OR
%upcase(&orient) = LANDSCAPE) %then %do;

%let orient = %upcase(&orient);

%end;

/* Orientation not specified: determine current orientation */
%else %do;

/* Set LINE & PAGE SIZE to Maximums */
options linesize=MAX pagesize=MAX;

/* Set Orientation to PORTRAIT */
dm 'dlgprtsetup orient=PORTRAIT nodisplay';

/* Get Current LINESIZE & PAGESIZE */
%let linesize = %sysfunc(getoption(linesize));
%let pagesize = %sysfunc(getoption(pagesize));

/* Are they still at the MAX? */
%if &linesize = MAX AND
&pagesize = MAX %then %do;

/* If so, Orientation is PORTRAIT */
%let orient = PORTRAIT;

%end;

/* If not, orientation is LANDSCAPE */
%else %do;
%let orient = LANDSCAPE;
%end;

%end;

/* Toggle orientation to force SAS to recalculate Line size and Page size */
%if &orient = PORTRAIT %then %do;
dm 'dlgprtsetup orient=LANDSCAPE nodisplay';
%else %do;
dm 'dlgprtsetup orient=PORTRAIT nodisplay';
%end;

/* Toggle back to correct orientation & recalculate Line size and Page size */
dm 'dlgprtsetup orient=&orient. nodisplay';

%let linesize = %sysfunc(getoption(linesize));
%let pagesize = %sysfunc(getoption(pagesize));

/* Check for Margin Settings */
%if %datatyp(&left) = NUMERIC OR
%datatyp(&right) = NUMERIC OR
%datatyp(&top) = NUMERIC OR
%datatyp(&bottom) = NUMERIC %then %do;
%put WARNING: *----- WARNING --- WARNING -----*;
%put WARNING: Margin settings are only valid for SAS Version 8.00 and;
%put WARNING: higher. Specified Margins will be ignored.;
%put WARNING: To set margins in Version &sysver, use the Page Setup Dialog.;
%put WARNING: *----- WARNING --- WARNING -----*;
%end;

%end;

```

SETPRINT Macro: Source Code. Figure 3, continued from previous page.

```
%put NOTE: *-----*;  
%put NOTE: * P r i n t   S e t u p   S p e c i f i c a t i o n s *;  
%put NOTE: *-----*;  
%put NOTE:          &orient.;  
%put NOTE:          &linesize.;  
%put NOTE:          Line size for printed output:  
%put NOTE:          Number of lines printed per page of output: &pagesize.;  
%put NOTE:          Font selected for printed output: &font.;  
%put NOTE:          Font size selected for printed output: &fontsize.;  
  
%if &sysver >= 8.00 %then %do;  
%put NOTE:          Left Margin:          &left.;  
%put NOTE:          Right Margin:         &right.;  
%put NOTE:          Top Margin:           &top.;  
%put NOTE:          Bottom Margin:        &bottom.;  
%end;  
  
%put NOTE: *-----*;  
%mend setprint;
```

SETPRINT Macro: Source Code. Figure 3, continued from previous page.