

# PRODUCING HARDCOPY TEXT OUTPUT WITH UNIX SAS 6.12

This paper covers how the SAS System utilizes UNIX utilities for printing. This is important in order to generate text output from SAS. This article discusses basic file printing under UNIX, interfacing the SAS System with UNIX print utilities through the use of the **PIPE** device type, and using the SAS Forms Subsystem.

With Release 6.12, the SAS System added the Host Printing facility to provide additional options for printing from within SAS. Host Printing allows end users to change printer settings without having to know the Printer Control Language (PCL) being used behind the scenes. **It is the assumption of this paper that you are not using Host Printing (i.e. Xprinter).** You can confirm that host printing is turned off by checking a toggle switch. Go to View→ Preferences, click on the Display Manager tab, and check the Host Printing radio button to make sure it is not selected. If you are interested in the features offered by the new Host Printing facility, they are documented in the Online Help as well as in TS543 (a technical paper available from our web site or by calling Technical Support and requesting a copy).

## Unix File Printing

In the Unix environment, file I/O is done directly by a device that is defined in the **/dev** directory. These devices can be terminals, floppy drives, tapes, and printers. When printing a file, you can send a file directly to a device in the **/dev** directory, but this would tie up the device, and restrict the sharing of the resource by multiple users.

In order to allow several users to share the print devices, the *lp* and *lpr* commands are used. These print commands take advantage of print queues or spool directories to allow many users to share the same resource. For the System V workstations, the print command is *lp*. On the BSD workstations, there is a similar print command that is called *lpr*. For the IBM AIX operating system, there is also another print command called *enq*. In the IBM environment, the *lp* and *lpr* command are treated as front-end utilities to the *enq* command.

### System V *lp* Command

When printing a file with *lp*, the *lp* command will make a copy of the file in the **/usr/spool/"destname"** directory. Next, the *lpsched* daemon will move the files from the spool directory to the appropriate device. A log of all processing performed by *lpsched* is written to the **/usr/spool/lp/log** file. Here is a list of some of the many other *lp* commands an administrator will use:

1. *lpadmin* - used to setup printer configurations.
2. *lpstat* - used to query printer status
3. *accept* - flags a printer as available for usage.
4. *lpmove* - used to move files between print devices.
5. *reject* - drain a printer, and will not accept files for queueing.
6. *disable* - drain a printer, but still accepts files in queue.

### BSD *lpr* Command

With BSD machines, access to printers is controlled by the *lpd* daemon and the *lpr* print command. When you specify a destination for an *lpr* supported device, the *lpr* command reads the **/etc/printcap** file for printer information and the name of the **/usr/spool/destname** directory, in order to maintain a copy of the file, and to build a control file. Once the file is processed by the *lpr* command, the *lpd* daemon is notified, and it will determine whether the print device is remote. If it is, the *lpd* daemon will direct the

file to another *lpd* daemon process running on the remote workstation, and then clean up any shadows of the file on the local workstation. The heart and soul of the BSD print system is the **/etc/printcap** file. Here is a brief list of some of the many other *lpr* commands an administrator will use:

1. *lpq* - query printer status
2. *lpc* - administrative changes for printers
3. *lprm* - removing queued jobs

## AIX enq Command

---

With IBM's AIX environment, an additional print command and print spooling facility was added. The IBM print facility is started with the *enq* print command, and then controlled by the *qdaemon*. The *qdaemon* acquires printer information from the **/etc/qconfig** file, then directs the print file to the appropriate queue in the **/usr/lpd/qdir** directory, and maintains status information about the file in the **/usr/lpd/stat** directory. Next, the *qdaemon* process uses the appropriate **backend** program listed for the specified queue, and directs the file to the proper device. To setup new printer information and print queues, you can use AIX's *smit* interface (system management interface tool). The *enq* command is used to administer and maintain the various print files, queues and devices.

**Special Note:** *When using the lp and lpr commands/options in AIX, they are just front-ends to the enq print command.*

## SAS Usage of Print Commands with the PIPE Device Type

Before attempting to setup a print facility for SAS, the SAS user or administrator should be familiar with his/her local print subsystem. The actual print command(s) that will be invoked by SAS will be one of the print commands that was discussed in the previous sections. Therefore, before trying to issue system print commands from within SAS, it is highly recommended that you test the exact same print command outside of the SAS system at a shell prompt.

### Creating a File Reference with the PIPE Device Type

---

The easiest method to setup printing for SAS under UNIX is to use the SAS *PIPE* device type. You can use this device type to create a SAS file reference. For example, the following is a file reference called *ascout*:

```
filename ascout pipe 'lp -dmyljet 2>${HOME}/lp.errors' ;
```

When the file reference *ascout* is used in a SAS window or program, a designated file (or contents of the window you are in ) will be piped to the specified *lp* command. In this example, the *lp* command is printing a file on the device called *myljet* and redirecting any print errors to a file called *lp.errors* in the user's HOME directory. This file reference is typically placed in an *autoexec.sas* file, or at the beginning of the user's SAS program.

**Special Note:** *The "2>\${HOME}/lp.errors" redirection of standard error is only allowed in Bourne and Korn shell.*

### Using the File Reference

---

The file reference you setup is actually a *pipe* to a UNIX print command. There are several ways for you to direct your output to this pipe device. The first way, and the easiest to code in your SAS job, is to use the *proc printto* statement to redirect your output or log information directly to the file reference. For example:

To redirect SAS Output to the printer:

```
proc printto print=ascout;

    Insert your sas code here

proc printto; run;
```

To redirect the SAS Log to the printer:

```
proc printto log=ascout;  
  
    Insert your sas code here  
  
proc printto;run;
```

**Special Note:** In the above examples the second *proc printto* is needed following your SAS code to actually release your output from the storage buffer and send it to the printer. For further documentation on *proc printto*, refer to the *SAS Procedures Guide*.

The second method to print to the file reference is using the *print* command from one of the SAS Display Manager window's command line, which will print the contents of that particular window using the unix print command you specified on the *filename* statement. For example, from the Output window command line, you can enter:

```
print file=ascout
```

Finally, the third method is similar to the above *print file=fileref* statement, but is performed using the X-windows interface to SAS. From the *File* pull-down menu, you can follow this path

```
File→  
  Print Utilities→  
    Set Print File  
      Enter Print File:  
        →at this point, enter the above "ascout" file reference,  
          which will print to the previously defined pipe.
```

With your Print File now set to "ascout", anything you print using the *File*→*Print* pull down selection or the *print* Display Manager command will use your predefined pipe.

**Special Note:** For the *Print File* name, you can also enter a file pathname( in single quotes), and this panel will save the print file as a file on disk. This file will be useful for printing or editing at a later time.

## Releasing the Print File

---

One fact to remember when setting a Print File using the *File*→*Print Utilities*→*Set Print File* menu selection or the *PRTFILE* command is that all future output will be sent to that Print File until it is cleared (or assigned to a different pipe or file).. You can clear it by issuing the following command:

```
prtfile clear
```

And, of course, you can exit SAS and the print file will be cleared automatically.

## Querying the Current File Resources

---

Two ways to see a list of your file references are:

From a SAS job, enter:

```
filename _all_ list;
```

and a list will be printed in your SAS log.

From the SAS X-window interface:

```
Globals→  
  Access→  
    Display File Names
```

and a window showing the filename references will be displayed.

Or, if you are using a command line in SAS Display Manager System (DMS), enter:

```
filename
```

and a window showing the filename references will be displayed.

## Using SAS Options to Modify Your Output

The default physical line size (number of columns per line) and page size (number of lines per page) of the files for the SAS log and procedure output is dependent upon their destinations. If these files will appear on your display monitor, the default line size and default page size will be derived from the size of your SAS Display Manager windows. (Note: In the Xwindows environment, the size of your window can be controlled by several Xresources and your SASUSER.PROFILE catalog. See the *SAS Companion for UNIX Environments: User Interfaces* for further information). If the files will be written to disk, then the default line size is 132, and default page size is 40. These two attributes can be overridden with the LINESIZE and PAGESIZE SAS system options.

For example, if you would like your SAS output file to have 50 lines per page (50 rows), and also have a width of 100 characters (100 columns), you can use the following methods to set PAGESIZE and LINESIZE:

On the SAS command line:

```
sas -pagesize 50 -linesize 100
```

In the SAS program or in the autoexec.sas file:

```
options pagesize=50 linesize=100;
```

Under the pull-down menu of the X-windows interface:

```
Globals→  
  Options→  
    Global Options  
      update the PAGESIZE and LINESIZE fields
```

Or, even in the config.sas file:

```
-pagesize 50  
-linesize 100
```

You can override these options by the settings you use in the FORMS subsystem. You can create a customized print form in SAS, and if you use the *forms=* option, or use the *File→Print Utilities→Set Form Name* panel and designate a form before printing the file, the values in the form for linesize and pagesize may take precedence over the SAS options. If for example, your output window shows 25 lines per page, but the form is set for 20 lines per page, your output will have 20 lines on the first page and 5 lines on the second, 20 on the third and 5 on the fourth, and so on. This can obviously cause a great deal of confusion.

***As a result, with version 6.11, we added a feature which allows you to enter -99 as the value of your “Characters per line, Lines on first page, and Lines on following pages” settings in your print form, causing the linesize and pagesize values you have set in your Display Manager session to be automatically honored when printing with the Forms Subsystem. We highly recommend that you take advantage of this feature if you are using Forms. See below for further information.***

## Using the Forms Subsystem in SAS

This section discusses some advantages of the Forms subsystem, how to create a form, and how to use your form. The Forms Subsystem is useful when setting up several print characteristics on the same print device. It allows you to send output altering

Printer Control Language (PCL) to your print device before it receives the print job. This capability will allow you to:

- choose a special font
- print portrait or landscape
- choose a paper drawer (ex. Letterhead from lower drawer)
- and whatever else you can find from the printer's control language

## How to Create a Form

---

There are two ways to create a form:

From a SAS command line, enter:

```
fsforms libref_name.catalog_name.form_name
example: fsforms sasuser.profile.hpljletr
```

Or, from the X-window pull-down menus:

```
File→
  Print Utilities→
    Open a form
      at the new Open a Form window, enter:
        Libname: sasuser
        Catalog: prtforms
        Form: hpljletr
```

```
Is this the default? Click on the appropriate reply
Click on "OK" to enter information
```

Inside the forms subsystem, you can step through a series of 6 panels that will be used to specify the characteristics of your print form. The panel names are:

- Printer Selection (This panel appears only when CREATING a form. It does not appear when EDITING an existing form.)
- Text Body and Margin Information
- Carriage Control Information
- Print File parameters
- Font Control Information
- Printer Control Language

You can scroll through the panels in two fashions:

- From the command line, you can enter *nextscr* and *prevscr* to go forward and backward
- From the **Locals** pull-down menu, use the **Next Screen** and **Previous Screen** selections.

As a default, new forms you create will be placed in your **sasuser.profile** catalog, but you can override the default by specifying a different library reference and catalog name in the 'Open a Form' window described in the 'How to Create a Form' section above. For more detailed information on the Forms Subsystem, refer to pages 811-821, in the *SAS Language - Reference Version 6, First Edition* manual.

## Creating Sample Form *hpljletr*

---

The following section provides an example of creating a form called *hpljletr*. This form prints letterhead output on a Hewlett Packard Laserjet IVsi. The form is saved under the libref of *sasuser* and a catalog of print forms called *prtforms*.

The form prints one page of letterhead from the lower drawer, and then prints the remainder of the job on regular paper from the top drawer.

To begin, enter the command:

**fsforms sasuser.prtforms.hpljletr**

You should now see the following on the title of the form **hpljletr**

Frame 1 Printer Selection

Form description [ HPLJLETR.FORM ]

Select a printer ...place the cursor on the type of printer you will be using for output. This selection will be used to define your "Font Control Information Frame" control and escape sequences for your printer type. If you do not see your exact printer listed, choose a similar one. For example, if you do not see a printer listed specifically for a Hewlett Packard Laserjet IVsi, select the Hewlett Packard Laserjet (+).

Special notes:

- The description "HPLJLETR.FORM" is filled in for you, but you can change this to be more descriptive
- You can not reselect the printer type. The only way is to delete the form and create another form.
- The printer selection only sets up the escape sequences for your printer type, which is why there is no disadvantage in choosing a plain Hewlett Packard Laserjet even though you may have the Hewlett Packard Laserjet Vsi. You can still take advantage of the features provided with the Vsi via the Printer Control Language frame of your form (see Frame 6 below).
- There is no way exit to the next frame in the form without selecting a printer.

You are given a chance to modify the defaults for lines per page, and columns per line, and the left/top/bottom margins. These settings will override any settings that are defined in the SAS PAGESIZE and LINESIZE. Note, that for this form the number of lines and the top margin for the first page have been modified. These changes are to accommodate the company logo on letterhead paper. Also, the values for *Characters per line* and *Lines on following* have been changed to -99. **Beginning with release 6.11, if you specify -99 for any of those values, the form will automatically use the same values that you have specified in your SAS PAGESIZE and LINESIZE options. It is highly recommended that you use -99 in those fields whenever possible, as it greatly reduces the chances of your form producing unexpected results.**

*Special Note: You can use the TAB key to move easily from one field in the Form Frames to the next.*

Frame 2: Text Body and Margin Information

Text Body:

Characters per line: [-99 ]  
Lines on first page: [43 ]  
Lines of following : [-99 ]

Margins:

First Page Left: [0] Top: [7] Bottom: [0]  
Following Pages: Left: [0] Top: [0] Bottom: [0]

Frame 3: Carriage Control Information

Generate Carriage Control Information? YES

Signal Page Skips before:

To turn on these options, place cursor over selection and hit enter:

- Printer Initialization
- \* First Text Page
- Following Printer Control Statements
- \* Following Text Pages

This form uses the default settings. The page eject before the first page is necessary if you have any escape sequences entered on the Printer Control Language page of your form. This is because blank lines containing the PCL are printed at the top of the first page, which causes your page formatting to be miscalculated, and each page of your output will wrap across multiple pages.

Note that the forms subsystem of SAS uses the *lp* command by default. If you need to use a different print command, you can change the SAS System Option *printcmd* before printing, as in the following example:

```
options printcmd='lpr' ;
```

If you would like to change the default printer destination outside the forms subsystem, submit the following option before printing:

```
options sysprint='-dmyljet' ;
```

The following **UNIX Print File Parameters** screen is a menu that allows you to set *lp* command options (or whichever print command you have chosen to use (i.e. *lp*, *lpr*)).

Frame 4: UNIX Print File Parameters

Copies: [ 1 ]                    number of copies to be printed. This is used with the -n option of "lp".

Dest: [ myljet ]                print destination. You would see this on the -d option of "lp".

◇ Send mail after file is printed.

Click on box, if you want "lp" to use -m option to mail you notification of any print problems.

Other line printer options:

-ox -opcl -ascii

The following are some useful options you may want to consider for the **UNIX Print File Parameters** screen:

-t "this is my special report"	...give printout a title
-p7	...set a priority
-oh	...suppress printing the header page
-otext	...specify the print file contains only text, NO PCL statements
-opcl	...specify the print file contains special control characters
-ox	...specify to print in simplex mode

The attributes defined in the "Font Control Information" frame are based on the printer chosen in the first frame. The top part of the frame has spaces for you to define up to 8 special characters that represent printer control characters. These characters are then used in the bottom of the frame to define text attributes concerning formatting changes such as underline text, bold text, subscripts, etc. Information regarding control and escape sequences for your printer is device specific information that can be found in your printer documentation. Customization is usually not needed for this frame.

Frame 5: Font Control Information		
<u>Character</u>	<u>Number</u>	<u>Description</u>
~	27	Escape
^		Control

Finally, the last blank page of sequence numbers is a screen that will allow you to send sequences of "printer control language" statements with your print job stream. The printer control language allows you to control fonts, print orientation, and which drawer to pull the paper. For a list of the control sequences that are available for your print device, refer to your printer's reference card. (A full explanation of the sequences used below follows this illustration).

Frame 6: Printer Control Language	
PRINT INIT	~&11H
PAGE 1	~&14H
PAGE 2	&11H
PRINT TERM	~E

**PRINT INIT** uses the following control sequences at the initialization of the print job. This example specifies that the top drawer be used for the cover sheet which is typically the header page of printer output.

**PAGE 1** loads the next control sequence before printing the first page of output and uses these settings until the next **PAGE** statement or **PRINT TERM**. This sequence tells the printer to take the first page of paper from the bottom paper drawer, which contains letterhead.

**PAGE 2** loads the next control sequence before printing the second page of output, and for the remainder of the print job. The example given tells the printer to use regular paper for pages 2-n.

**PRINT TERM** uses the last sequence on the termination of the print job. The sequence shown in the example is 'resetting' the printer back to the initial settings.

Now that you have completed all the panels have been completed, just use the save command or **File→End** from the pull-down menus to save the completed form. You may now use your new form.

## Printing with a Form

---

There are three ways to set a form while attempting to print from the SAS Display Manager System windows (program editor, log, output, etc):

1. Use the "options forms=" statement:  
`options forms=sasuser.myforms.hpljletr;`
2. From the **File→Print Utilities→Set form name** panel:  
Enter Form Name: `sasuser.myforms.hpljletr`
3. From the SAS Command Line:  
`formname sasuser.myforms.hpljletr`

Once you have set the form, the next output file that is printed will use this form for the print characteristics and destination. If no destination has been set, it will check the value of your SYSPRINT SAS option (as defined on page 6). If that has not been set, it will print to your default system printer (usually defined by the \$LPDEST environment variable).

If you want to use forms with PROC FSLETTER, there are three ways to set the form name for your letter:

1. On the command line of your letter, enter:  
`form sasuser.myform.formname`
2. Using the pull-downs, and currently editing a letter:  
Locals→  
Assign a Form  
Form name: `hpljletr`
3. Or, before printing the letter while using pull-downs:  
File→  
Print Utilities→  
Set Form Name  
Enter Form: `hpljletr`

## Obtaining a List of Forms

---

If you happen to forget the names of your forms, and want to see a current list of those available, you can use the catalog procedure or PROC CATALOG to display the contents of the catalog to the SAS log, as shown below:

```
Using PROC CATALOG in SAS job:  
  
proc catalog cat=sasuser.myforms;  
contents;  
run;
```

## Troubleshooting

This troubleshooting section is a culmination of common problems and solutions that are frequently asked from UNIX customers. Hopefully this list of problems will provide insight to some areas that might be of concern in your environment.

## Choosing a non-"lp" Print Command

---

As a default, the SAS forms subsystem uses *lp* as the default print command. To override this usage, you can set the *printcmd* SAS System option. In the following example, the default SAS print command is set to *nlp*:

```
options printcmd='nlp';
```

Alternatively, you can create an alias for *lp* using your system's print command. This way, when files are printed using forms, they will be implemented by the command that is referenced in the alias, and will be passed the *lp* print options listed in the **Print File Parameter** frame. For example, if your system uses a print command called *nlp*, then you can do the following:

#### From c shell:

In your *.cshrc* file, you will need to add the following:

```
alias lp 'nlp'
```

#### From bourne shell and korn shell:

```
ln -s /usr/local/bin/nlp $HOME/bin/lp
```

or

```
alias lp='nlp'
```

## Common Errors for Printing with PIPE

---

### Large Files not printing

When using the *lp* command on a file, a symbolic link is created from the file to the */usr/spool* directory. Alternatively, when you use a UNIX *pipe* to the *lp* command, a temporary copy of the file is created under the */usr/spool* directory. Consequently, you may have difficulty printing a large print job if the size of the file exceeds the available disk space in the */usr/spool* directory. For example, you may experience this problem using the *pipe* device type when attempting to print a SAS output window that contains a large amount of output.

To circumvent this problem, the easiest method is to save the print file to hard disk, and then print with the *lp* command. Two methods to save the file to your current working directory are:

From the output or log window command line, enter:

```
print file='my.file.output'
```

Now that you have the file on disk, you can exit SAS and print the file, or from the Program Editor, submit:

```
x 'lp -dmyljet my.file.output';run;
```

Also, you can create another type of file reference that can handle large print files. This file reference will save the output file to disk, print the saved file, and then remove the file. Below is an example of this file reference:

```
filename catout pipe 'cat >BIGFILE; lp -dmyljet BIGFILE; rm BIGFILE';
```

## Common Problems with Form Setup and Usage

---

### “unrecognized file type” as only line of print file

If the only line of the print file is a statement which reads:

```
“unrecognized file type”
```

then you are probably trying to send **pc** statements to the printer, but have not specified the correct printer option to signal the printer that your file contains a mixture of ascii and control sequences. Check the **UNIX Print File Parameters** and your specified **-o** options. (On some systems **-oraw** will work, but check the unix man page for your specific print command).

### Extra page breaks coming out between pages

If you have extra page breaks (blank pages) in the middle of your output, the most likely culprit is a disagreement between your SAS LINESIZE and PAGESIZE options and the 'Lines per Page' settings in your form. **We highly recommended that you set the second frame of your form (Text Body and Margin Information) to have the following settings:**

Characters per line:	[-99]
Lines on first page:	[-99]
Lines on following:	[-99]

These settings cause the form to automatically use the values you have set in your LINESIZE and PAGESIZE options, which greatly reduces your chances of getting extra blank pages or output broken across several pages.

Lastly, we recommend that you leave the Carriage Control Information frame of your form with the default values turned on. These options are to generate a page break before the first text page and following each text page, especially if you have any Printer Control Language customizations in your form.

### Page breaks in the middle of your output

This error is typically caused by a disagreement between your form settings and your PAGESIZE option (see above) or you are trying to print more lines than can physically fit on the page with the font chosen. To resolve the first issue, you need to either set your form to have -99 for the Text Body and Margin Information values as shown above. If you have a specific need that requires you set them to a specific value, the value chosen should always be less than the coordinating SAS option value. For example, *Characters per line* should always be less than LINESIZE, and *Lines on first page* and *Lines on following* should always be less than PAGESIZE.

### Only the title is printed - the text is not

If your printed output consists of a page with only the title line followed by several blank pages, then you should add the following escape sequence to the Printer Control Language frame of your form:

```
PRINT INIT
~&k2G
PRINT TERM
~E
```

Also ensure you have chosen to generate carriage control information in Frame 3 of your form:

```
Generate Carriage Control Information? YES
```

### Output printed in *duplex* mode

If your printer supports duplex printing (front and back of paper), but you want to print to only one side of the paper, check the **UNIX Print File Parameters** and add the specific print option to print in **simplex** mode. This option may be **-ox** for your site. Check with your UNIX system administrator for the necessary option.

### Output is not routed to the destination you specified in your SYSPRINT option

The SYSPRINT SAS option allows you to change the destination used by your print command. For example, to have SAS use the *nlp* print command and send it to the *mydeskjet* print queue with a customized title, you might specify the following options:

```
options printcmd='nlp';
options sysprint= "-dmydeskjet -t'my title'"
```

This SYSPRINT option will work properly as long as you have not made ANY customizations to the Print File Parameters frame of your form. Once you make a change to that frame, any values you set in your SYSPRINT option will be ignored.

If you are still unable to make your printout magically go to your printer, check to make sure you have not set a Print File (type PRTFILE on the SAS command line). If it returns a file location, clear the Print File with the following command:

```
prtfile clear
```

Next, try to print again with one of the previously mentioned methods.

## Helpful Hints

### **More useful Printer Control Language Statements**

---

In addition to the example of producing letterhead output, there are other common usages for the Forms Subsystem and printer characteristics. A few of the more frequently asked questions and solutions are:

## Printing landscape with compressed fonts:

### Frame 6: Printer Control Language for Landscape and Compressed

```
PRINT INIT

~&l10      ← this is: tilde(~) ampersand(&) lowercase L(l) one(1) zero(0)

~&k2S      ← this is: tilde(~) ampersand(&) lowercase K(k) two(2) upper S(S)

PRINT TERM

~E        ← this is: tilde(~) uppercase E(E)
```

## Simulating Forms in the Batch Environment

---

Since the Forms subsystem is not available in “Batch” mode, if you need to add Printer Control Language customizations to your print file, it has to be done through your SAS code. Instead of using the Printer Control Language frame of your form, you can use the SAS Data Step to send the appropriate PCL as the beginning of your print stream. The following is a simple example of this. If you are interested in this method, please visit our web site for an example that makes use of the macro facility at: <http://www.sas.com/techsup/download/sample/unix/dstep>.

```
filename test pipe 'lp -oraw -dmylaserjet';
proc printto print=test; run; /* This sets the print file to your pipe */
data one;
  x=1;
run;
data _null_;
  file test;
  put '1b'x '&l10';          /* This puts printer in landscape mode */
  put '1b'x '&k2S';          /* This puts printer in compressed mode*/
run;
proc print data=one;run;    /* This will send the output directly to the
                             printer since you are using proc printto with a
                             pipe */
proc printto;run;         /* This is necessary to release the print file*/
```

---

## How do I know which form I am currently using?

---

You can determine which form you are currently using by issuing the following SAS Display Manager command:

```
formname
```

The form currently being used will be echoed to the message line of the active window.

You can also check the FORMS option in your PROC OPTIONS output or in the Globals→Options→Global Options screen.

If the name given with either of the above methods is `DEFAULT.FORM`, check the following:

Do you have a `DEFAULT.FORM` in your `SASUSER.PROFILE` catalog? If so, this is the form that SAS is using. Otherwise, SAS is using the `DEFAULT.FORM` in your `SASHELP.BASE` catalog (this form is shipped with the SAS System).

## How can I create a postscript file from within SAS?

---

The easiest way to create a postscript file from SAS is to use a UNIX utility like *pstext* as a filter to your print file. For example as previously mentioned, the SAS option *printcmd* controls the UNIX shell command executed when you print from within SAS. You can run the output through a postscript filter and create a postscript file....or run it through the postscript filter, and pipe that output to a print command that sends it to a postscript printer. Two ways to do this are:

### To create a postscript file:

```
options printcmd='pstext';
options sysprint='>/tmp/file.ps';
```

Then, when you issue a `print` command from within SAS, the file `/tmp/file.ps` will be created.

### To send output to a postscript printer:

```
options printcmd='pstext';
options sysprint='| lp -dmylaserjet';
```

Then, when you issue a `print` command from within SAS, a postscript version of the file is sent to the printer *mylaserjet*.

## How do I know which printer SAS will use?

---

When printing, SAS looks in the following locations to find which printer to use.

- The Destination option of the Print File Parameters screen in the current Form being used (if one is in use).
- The `SYSPRINT` option. Note that if ANY changes have been made in the Print File Parameters screen of your form, your `SYSPRINT` option will be totally ignored.
- The `$LPDEST` environment variable.

SAS searches the above locations in the order listed, and once it finds a printer destination, it does not continue to search. On other words, if you have a printer specified in all three locations, the one in the Print File Parameters of your form will be the one SAS uses.

The location you choose to set your printer destination will vary based on your needs. Certainly, if you change it in your Print File Parameters of all your forms, you will experience a lot of work if you obtain a new printer, since you would need to change ALL forms. However, if you are an applications developer trying to prevent your users from altering their environment, that may be just the place to specify a print destination. Choose the most suitable method for you.

## I'm using an Xemulator on a PC to display UNIX SAS. Can I use my local PC printer from within SAS?

To be able to use a printer from within SAS, you have to be able to print to that printer from the unix shell level (i.e. from the command line in an xterm). If you are able to open an xterm and display it on the same PC, and use your print command to get output to that printer, you should be able to print from within SAS. Whatever print command you use will become your *printcmd* option, and your destination will become your *sysprint* option (These options are discussed in detail in a previous section).

If, however, you are not able to print from an xterm to that printer, there is no way to access it from within SAS.