SAS® on Display
An Introduction to the SAS Display Manager

David Beam
Systems Seminar Consultants
Madison, WI 53716 (608) 222-7081

Abstract
Many people new to the SAS System, or even those with extensive batch SAS experience, often reject the use of the interactive Display Manager as a testing and development platform. The common fears are that the Display Manager is cumbersome, hard to learn or terribly inefficient in using computer resources.

This paper will explain why and how to use the Display Manager for faster SAS program development. The discussion is primarily aimed at the mainframe MVS operating system, but the majority of the ideas will also pertain to other systems.

Introduction
There are three major methods of invoking and using the SAS System. The most common in the mainframe world is traditional batch processing, using a text editor and operating system commands to interface to the computer (for MVS, Job Control Language or JCL). SAS can also be invoked in line prompt mode, where a single SAS statement is typed and submitted at a time. This method is rarely used.

The best development and testing environment for SAS is the Display Manager mode, which marries the fast response of the line-prompt mode with the editing and "full-screen" capabilities of the batch system.

What is the Display Manager?
Using SAS under the Display Manager (referred to as DM in this paper) allows the user to work in screens or windows to accomplish program editing, submit code for processing, and reviewing the reports generated. The movement between screens, actions of the 'PF' keys on the keyboard, sizing and placement of screens is all controlled by the user using SAS DM commands.

If you are familiar with using a personal computer windowing software package, the Display Manager on the MVS mainframe is not as robust as you may expect. Many of the ideas of PC windowing, however, are available in the Display Manager.

Why Should you use the Display Manager?
Speed. That's what we all want more of. Using the interactive DM allows much faster development of programs, testing of new ideas and debugging of code.

Experimentation. The quick response encourages "what-if" analysis, by allowing small changes to a program to be tested and reviewed quickly. Trying new options or statements or procedures is a snap in DM.

Prototyping. Suppose your boss wants a major new report developed, but hasn't settled on a report format yet. In Display Manager, you can quickly create a small sample data set, prepare several sample report styles using a variety of options and procedures, and even let the boss see them online to assist in deciding which approach to take.

Help. Yes Virginia, there is help when using SAS. The online help screens are not meant to replace the written SAS documentation, but can be a big assist in checking syntax, looking up new options or remembering the name of that obscure procedure that you need NOW.

To use .. or not to use .. the Display Manager
There are some practical as well as real limits to using the Display Manager, particularly on the MVS mainframe. You should only work with data stored on disk, not magnetic tape. In fact, most sites turn off the ability to interactively mount tapes from the DM. The size of the files being processed should be relatively small (or only read in a small number of records from large files).

The overhead of any interactive software on the mainframe is greater than batch processing. If your site charges you for computer times used, the rates for DM use versus batch are probably higher. The increased development time and faster turn around will usually justify the extra cost.

Most of our development is a mix of Display Manager and batch. We often develop and test systems in DM, then take the tested code and run it in batch against
the large mainframe files. It is the best of both worlds.

What this paper does NOT cover...
The interactive development software tools from SAS such as SAS/AF®, SAS/ACCESS® and SAS/EIS® are not discussed here. SAS for Windows® or OS/2® is not described, although the points discussed here will also pertain to those environments.

Features of the Display Manager
The Display Manager is an interactive environment for developing and executing SAS programs, featuring:

- An excellent ISPF-like full screen editor
- Commands to copy and create programs
- Windows for viewing SAS log and output
- Commands to print windows
- User definable function keys and screen layout
- Windows for managing global SAS items
- Built in debugger (Release 6.11)

Quick Overview of the Main DM Windows
There are three primary windows used in the Display Manager:

Program Editor window
A text editor that operates similar to the IBM editor many mainframe users are familiar with. This screen is used to type SAS programs, submit them for processing, and storing and retrieving programs from an external storage file, or a SAS library.

Log window
This screen displays the SAS log, just like in batch processing. Your submitted source code, error messages, notes and warnings from the SAS software are displayed here.

Output window
Reports from PROC steps (such as PROC PRINT) or PUT statements from DATA steps will be displayed here.

There are several other windows available, some of which are discussed later. For our initial discussion, we will concentrate on the above three.

Starting SAS with Display Manager
How you invoke SAS will vary from site to site. In a generic IBM MVS TSO or ISPF site, SAS is started from the TSO "Ready" prompt, or from the ISPF command line (usually ISPF option 6). Refer to your local SAS support staff for instructions.

SAS can be invoked with some system option overrides, or even with a specified program to be run at the start of your session, refer to the corresponding SAS documentation for your operating system for instructions.

Example from TSO Ready prompt:
```c
=> SAS
```

Example with a System option change:
```c
=> SAS OPTIONS("STATS NOCENTER")
```

The starting Display Manager screens will look something like this:

<table>
<thead>
<tr>
<th>Figure 1</th>
</tr>
</thead>
</table>

Note - The screens displayed in this paper were generated via a mainframe emulator software package running on a personal computer. The terminal you use will affect how the screens appear.

Moving around in Display Manager
There are several commands available to jump from one window to another inside DM. Each window has a "Command" line at the top, this is where windows movement commands are entered.

You can jump to another window or open a new window by typing the name of the window on the command line. For example, the following command will move to the LOG window:

```
Command => LOG
```

In any window, browse commands allow scrolling within the window, such as RIGHT, LEFT, UP, DOWN, TOP, BOTTOM. To backup one window (returning to the previous window) use END on the command line.

One nice command, if you are not sure which windows are active, is the NEXT command. It cycles through all
the Display Manager windows currently open.

Using the PF Keys on the Keyboard
The PF, or function keys on the keyboard have some default settings for use in SAS. These settings are for SAS use only inside the Display Manager. At some sites, users entering the Display Manager are allocated a small permanent SAS library that can be used to store your PF key settings if you change them.

To modify or display the PK Key settings, enter KEYS on any command line. The SAS default key settings are displayed in the SAS companion guide for your operating system. These keys can be used to scroll windows, submit text, move between windows, mark or cut or paste text, etc. Most valid Display Manager commands can be assigned to a PF key with the KEYS window.

Using the Program Edit window
Creating, editing and running SAS programs:

1. Type SAS source statements in the Program Editor window. You can use line commands and window commands to manipulate the text. Line commands are used in the sequence numbers, window commands are typed on the top, or command line of the window.

2. When the code is ready to run, enter the SUBMIT command on the window command line to send the contents of the window to the SAS system.

3. The Program Editor window will clear. The SAS log will be added to the LOG window and any output will be added to the OUTPUT window. If output is generated (example: PROC PRINT), the OUTPUT window will automatically become active. Use the END command in the OUTPUT window to complete the current output and return to the Program Editor.

4. To retrieve the last text submitted, enter the command RECALL on the command line in the program edit window. Subsequent RECALL commands can be used to retrieve code submitted before the last SUBMIT.

Sample Screens from a Display Manager Session
Program ready to SUBMIT, before the ENTER key is pressed:

The Output window, after program has run:

After entering END in the output window, you are returned to the Program Edit window:
Using External Program Files
In order to use SAS programs written in the Display Manager in traditional batch processing, or to edit the programs outside DM, you can store the code in external files. For this example, suppose we have a mainframe program storage area called a partitioned data set (PDS) named: "MYID.SAS.SOURCE".

Other operating systems have similar files, for example in the PC/Windows environment you may have a subdirectory called "C:\MYOWNISOURCE". The following concept can be adapted to these naming conventions.

At the command line of the Program edit window the following two commands will store or retrieve the program called "TEST1" in the above mentioned file:

To store the program:

```plaintext
===>
FILE "myid.sas.source(TEST1)"
```

To retrieve the program into the Program Editor:

```plaintext
===>
INCLUDE "myid.sas.source(TEST1)"
```

Every time you want to save the code to this external file, you will need to issue the "FILE" command. (Once a full file name is given, you only need to enter the command "FILE" if you want to save it to the same member, until you want to save to a different file).

**Using the FILENAME Statement**
The SAS FILENAME statement allows you to name an external file with a simpler "reference" name. The above FILE and INCLUDE commands get a bit tedious when you use the same storage area over and over, so the following program statement can be submitted once per Display Manager session to name an external file:

```plaintext
FILENAME source "MYID.SAS.SOURCE";
```

**NOTE** - this is a STATEMENT, not a command, so it must be coded in the lines of the Program Edit window and SUBMIT must be issued to send it to SAS.

Technical note - the word 'source' in the above example is telling the operating system to set up a DDName called SOURCE that can be used in SAS instead of typing the full name of the file. Now the above FILE and INCLUDE can be:

```plaintext
===>
FILE source(test1)
Or
===>
INCLUDE source(test1)
```

**Naming SAS Libraries**
When running a batch program in the MVS system, users need to have Job Control Language (JCL) to name files needed by the program. In version 6 of the SAS System, these JCL file definitions can be coded directly in the SAS program code, eliminating many of the needs for the JCL "DD" statements.

Suppose you need to access a permanent SAS library with the mainframe name "Myid.sas.data". The SAS LIBNAME statement can be used in the program, with a short or reference name added:

```plaintext
LIBNAME sasin "MYID.SAS.DATA";
```

In the above example, the word SASIN can now be used to reference a permanent SAS data set, such as the following program snippet:

```plaintext
DATA sasin.sales;
    SET tempin;
    ...
    ...
RUN:

PROC PRINT DATA = sasin.sales;
RUN:
```

**Naming External Files**
As the LIBNAME statement above, the FILENAME statement can be used to reference external files. Many of the JCL parameters (such as disposition, block size, etc.) can be used as well, refer to the appropriate SAS documentation.

If you are reading data from a raw or non-SAS source, the following can be submitted with your program:

```plaintext
FILENAME rawin1 "ACCT.GLDATA.MONTH.WIS";
```

Your SAS program can point to this file on any INFILE or FILE statement, such as:

```plaintext
DATA sasin.sales;
    INFILE rawin1;
    INPUT @1 company $25.
    ...
    ...
RUN;
```

**Once per Session!!!**
Both the FILENAME and the LIBNAME statements need only be entered in the Display Manager ONCE per session, as they are remembered by SAS until you terminate your session. Which brings up a very important point....
How to Leave the Display Manager...
The omnipresent “QUIT” that seems to work in almost all software does NOT work in SAS...

To leave the Display Manager session, type one of the two following secret words on any command line and press enter:

**BYE** or **ENDSAS**

CAUTION - if you just spent 45 minutes typing the greatest program that your boss will love....and you haven’t saved the source code (FILE ...) before you type BYE, weep now. SAS won’t warn you.

Save your work before leaving Display Manager.

**Neat SAS Tip when using PDS files..**
I must admit. When preparing for this paper, I discovered a SAS Display Manager command for the MVS system that has been available for years, but I hadn’t learned to use it. On any command line, the following command will open up a new window showing all the members in the referenced PDS. From this window you can browse code, delete members, edit, rename and include into the program edit window any member:

```
==>
MEMLIST ‘full.PDS.file.name’

Or

==>
MEMLIST ddname
```

Refer to:
“SAS Companion for the MVS Environment” manual for complete description. (My experience in learning about the MEMLIST command proves it **pays** to read the manuals!!!!!!!!)

**Review of above ideas...**
Let’s look at a typical SAS Display Manager Program Edit window using the above statements. The following screen demonstrates:

1. Referencing a SAS library
2. Reading a non-SAS data file
3. Creating a permanent SAS dat set
4. Saving the program to an external file

By placing the cursor on a topic and pressing ENTER, the next level of help is reached. Let’s select the topic “SAS Windows” to see all the Display Manager windows available:
SAS Windows Help Screen:

Now we will see a more detailed help screen, by selecting the following help choices (from the Main Help menu):
1. SAS Language
2. SAS Functions
3. Function Categories
4. Character Functions

the resulting screen:

Other Windows worth mentioning
As seen on page 6, there are many other windows available within the Display Manager. Below are listed a few of the windows we use frequently, don’t be afraid to open any of the windows in DM to see what they can do:

- **MANAGER**
  Manage all reports currently generated
- **LIBRARY**
  see what SAS libraries allocated
- **DIR**
  look at last accessed SAS library files
- **FILENAME**
  see what non-SAS files allocated
- **OPTIONS**
  see current OPTIONS settings and change interactively
- **CALCULATOR** (or DCALC)
  Pop-up calculator

Pitfalls and Problems Using Display Manager
As great as the DM is for fast development, quick testing, there are a few things that give us grief on a semi-regular basis. Below are a few irritations that need to be mentioned, with some possible remedies.

Be aware that these problems relate to version 6.07 on the MVS system, and may not be the same problem on other platforms or under later releases.

Missing ending quotation marks
If you enter a program statement that surrounds text with quotes, and forget the ending quote or use the wrong one (single versus double), the SAS log reports a problem with a text string too long. Example:

```
TITLE "Hello World; (ending quote needed)
PROC PRINT;
RUN;
```

In batch processing, you merely correct the TITLE statement and resubmit. In DM, SAS is still looking for the ending quote. Sometimes submitting the following helps:

```
" ; RUN;
```

If not, occasionally we need to exit SAS (BYE) and restart the Display Manager. Save your work first!! (NOTE: in release 6.10 under PC/Windows, a nice message appears in the LOG saying how to fix the unmatched quotes.)
No automatic program save
SAS does not offer the ability to FILE your source code every few minutes. You must do that yourself, as well as before exiting DM.

Cancel a SUBMIT
Display Manager does allow you to interrupt a SUBMIT by pressing the "ATTN" key for your system, however, our experience with a variety of mainframe keyboards/emuators is that the "ATTN" key is not easy to find!

Printing from DM to Mainframe Printers
It is not that this can't be done, but often requires the SAS software installer to set up FORM definitions, and can be cumbersome to maintain for a large number of users. Our normal practice is to use batch processing when we need to print large reports, or to use the FILE command at the top of the OUTPUT window to save the report to a flat file.

The PMENU System in Display Manager
For those of you who prefer drop down menus instead of typing commands at the top of the windows, there is a menuing system in Display Manager called PMENU. It can be invoked as an installation default, or a user can issue the PMENU command on any screen (often set to the PF9 key). It can be turned on/off.

This menu system is very nice on an operating system that allows full use of a mouse, but under MVS some users feel it is slower than typing commands.

Below are a couple examples of screens in Display Manager showing the PMENU system. Each window in DM has a menu bar at the top. You select choices by placing the cursor on the choice and pressing ENTER:

PMENU in Program Edit, about to SUBMIT:

Coming Attractions
As mentioned earlier, release 6.11 under Windows has a DATA step debugger built into the Display Manager. The debugger consists of windows and a new group of commands. You will be able to trace the action of your DATA step, by stepping through the execution, or skipping statements as it runs. You will be able to check the values of variables while the step is processing.

At the time this paper was prepared it was unclear to the author if or when the debugger would be available for the MVS mainframe. Refer to the references below for more information.

Conclusion
I think you can see that the Display Manager is one of my favorite development tools in SAS. Testing ideas, learning new coding techniques, "tweaking" code with minor changes and quick response under Display Manager is worth the effort to learn the screens and commands.

Keep in mind that data sets and file sizes should be kept small in the interactive world of the mainframe. Saving code outside of SAS libraries allows easier use in batch processing. Each new release of the Display Manager adds new power and capabilities that any developer should be eager to embrace.
References


Author

The author will be glad to answer questions and accept suggestions at the following address:

David Beam
Systems Seminar Consultants
1220 Femrite Drive
Suite 206
Madison, WI 53716

voice: (608) 222-7081
Internet: train@sys-seminar.com