THE SAS DISPLAY MANAGER SYSTEM FOR THE OS/MVS ENVIRONMENT

R. Michael Gatsavall
SAS Institute Inc.

The SAS Display Manager System (DNS) is a full-screen interactive facility available as an integral part of the base SAS software product. The display manager allows users to enter and edit programming statements, submit them to the SAS System, and to view the program output. DNS provides the user with quick and easy access to all parts of a SAS job in a full-screen environment.

Visually speaking, the display manager is composed of two primary screens. The first screen, displayed after initialization, is a split screen containing two logical windows. A horizontal dashed line indicates the division between the SAS log and the program editor window. For the sake of this discussion "screen" is used to refer to the entire terminal face and "window" is considered a logical subset of a screen.

This is the initial screen showing the log and program editor windows.

Users can dynamically enlarge or shrink the size of the program editor and log screen by using the SPLIT command. To alter the split, position the cursor on the row where the division is to occur and press the SPLIT function key. The entire screen can be taken up by the program editor, completely obscuring the log. However, the converse is not true, and at least one editor line remains on the screen.

The second screen is the procedure output window which can be accessed by either pressing the OUTPUT function key or typing OUTPUT on a command-line and pressing enter.
This is the procedure output screen prior to a procedure having been executed. At this point it is empty except for a command line.

Procedure output can be displayed automatically while the procedure is running by setting a DMIS option. If this option is set, the display manager will swap to the output window and the procedure output will be presented in page browse mode. Both the log and output windows are always positioned such that the newest information is displayed.

Interaction with the display manager is accomplished through the use of function keys and command-line commands. Any valid DMIS command can be assigned to a function key. Multiple commands, either typed on a command-line or assigned to a function key, are valid, but they must be separated by semicolons.

A trivial and not very useful example would be:

Commands assigned to function keys are concatenated in front of any existing text on the command-line. It is recommended that function key commands that do not take operands be followed by a semicolon. This helps to reduce conflicts due to the interaction between commands assigned to functions keys and command-line commands.

Functionally, the display manager can be broken down into four major components. The main DMIS module (DMIS proper), the stream or backing storage manager, the line editor and the command-line processor.

The main DMIS module is concerned with memory allocation, setting up initial conditions and options, interfacing with the full-screen I/O
routines for terminal communication and
initializing the remaining DBS support routines.
Another of this routine's many tasks is to
handle the reception of procedure and log
output. As information is received by the main
dBS module it is routed to the stream manager.
Optionally, the stream manager will journal the
data to a utility data set or maintain the
information in virtual storage as in-core arrays
for quick retrieval. Currently all three
windows or "streams" are controlled by the
stream manager. Although information contained
in the program editor is considered a stream, it
is actually maintained internally as a doubly
linked-list. This type of data management makes
life a lot simpler for a full-screen editor.
The program editor is a full-screen interactive
editor that is useful for preparing program
statements and data or for general editing. The
program editor has a large repertoire of line
commands that are typically one or two character
mnemonics most of which support repetition
factors.

For example, here is an almost complete data
step followed by a proc chart.

```
DATA TEST:
DO i = 1 TO 100;
OUTPUT;
RUN;
PROC CHART; VBAR i; HBAR i;
RUN;
```

To issue a line command overtype the line-number
field and press the enter key. Here we are
going to insert a new line after line 3:

```
DATA TEST:
DO i = 1 TO 100;
OUTPUT;
END;
PROC CHART; VBAR i; HBAR i;
RUN;
```

326
The DBS editor supports the usual assortment of single character commands like C (copy), D (delete), I (insert), M (move), R (repeat) and the block commands like CC, UD, HM and RR. Also available are TF (text flow), TS (text split), and data and column shifting commands. Four-way scrolling (left, right, backward and forward) is available by pressing the function keys or issuing scrolling commands from the program editor command-line. The "::" (colon) command is a special command-line command that allows the user to issue line commands from the command-line.

For example,

```
type :TS
```

on the program editor command-line, position the cursor in the edit window where the text split is to occur and press the enter key.

The DBS editor was designed to be an easy-to-use, familiar tool for the SAS programmer. Used in conjunction with command-line commands for copying and saving data to OS data sets, and most important, submitting program statements to the SAS System, the program editor rapidly becomes an indispensable feature.

The command processor, sometimes referred to as the command dispatcher, is responsible for verifying that "text" entered on a command line is a valid command verb or alias. The command processor is also responsible for dispatching the command to the proper entry point and handling any error situations that may arise while processing the command. In addition to the "::" (colon) and SPLIT command, the display manager provides the user with many other commands to control the SAS session and to manipulate the DMS environment. DHS supports FIND, CHANGE, CAPS ON or OFF, the LINESIZE command (in all three windows), and NULLS ON or OFF as well as cursor positioning commands. For color terminals DHS supplies commands to alter almost everything on the screen. For example, users can modify the color and attributes of the message lines, SAS source statements, error and warning messages in the log, and protected and unprotected fields. For monochrome terminals with extended data stream capability, users can modify the HUEB (highlighting, underlining, reverse video and blinking) attributes. There are also many miscellaneous commands available to the user. The major commands are described in some detail below.

As statements are submitted to the SAS System by way of the SUBMIT command or function key, recall elements are built to identify these submitted lines of data. These recall elements are maintained in a LIFO stack. By using the RECALL command or function key, the user can recall groups of data that have been submitted to the SAS System. When the SUBMIT command is issued, all data in the program editor window is submitted to the SAS System and the program editor window is cleared. The RECALL command will retrieve from the stream manager the most recent group of statements that has been submitted to the SAS System and redisplay them in the program editor window. At this point, the data is again available for editing or re-submission.

Type SUBMIT on the command-line to submit this simple SAS job.

```
Command ---->

```

As you can see, the SAS source statements appear in the log window and disappear from the program editor window.
again will restore the output window to page browse mode.

Here is an example of procedure output in page mode.

After pressing the COMMAND function key, a command-line is inserted at the top of the screen and the output is now presented in line browse mode. Notice that part of the second page of the proc chart becomes visible.

The output window has now been restored to page browse mode.

Type RECALL to recall source statements to the program editor.
The SAS source statements have been recalled to the program editor window.

If you continue to enter the RECALL command you will eventually get the message "No data to recall.

This AUTOEXEC initializes the log as the active window and sets the linesize to 132. The next command switches the active window from the log to the program editor window. All commands after the PROGRAM command now pertain to the program editor window at least until a LOG or an OUTPUT command is encountered. The COLOR command is issued to alter the color of the unprotected fields. Actually all fields in the program editor are unprotected. The program editor banner is defined as the part of the screen that contains the "Command =>... Program Editor" text and the line-numbers down the left hand side of the screen. Nulls is turned on so that the insert key works.
and a data set is allocated by issuing the TSO command. Then member "sample" is copied into the program editor window from the partitioned data set referenced by the DDname x. The CURSOR command is used to position the cursor on row one prior to issuing the SPLIT command. The visual result of this AUTOEXEC will be that the program editor window will completely obscure the SAS log and display the member "sample". At the completion of a successful AUTOEXEC you will get the message "AUTOEXEC processing finished." displayed on the message line.

Another major highlight of DBS is function keys that can be modified by the user and saved from one session to another. Gaining access to the function key definition panel is accomplished by issuing the KEYS command from any command-line. This will invoke a full-screen panel displaying the current function keys definition for the window from which the command was issued.

### Command ---->

```
Command ---->
```

### KEYS COMMAND

```
01 help
02 split;
03 submit
04 recall;
05 find;
06 change
07 backward
08 forward
09 output;
10 left
11 right
12 cursor;
13 help
14 split;
15 current atr 45;split;submit;
16 recall;
17 find
18 change
19 backward
20 forward
21 output;
22 left
23 right
24 cursor;
```

Each window has a unique set of function key definitions. A search of three data sets is made to find the first occurrence of the key definitions to be displayed. The data sets, in order of search are the WORK data sets, SASUSER and finally SASUTL. The default set of key definitions is stored in the SASUTL data set which is referred to as an "installation profile library." This dataset contains, in addition to other objects, the "global" set of DBS key definitions. The installation profile library is intended to be a universal read only data set and can not be modified by DBS. In Version 5 a user profile library is available for customizing function key definitions to meet individual needs and preferences. The user profile must be a SAS data library allocated to the DDname of SASUSER. Once a set of key definitions has been found and the panel has been displayed, the user is free to make any changes. To exit the function key definition panel type END on the command-line and press enter.

### Assume that the keys have been modified, the list of data sets is searched in reverse order, starting with the SASUSER data set to find a library in which to save the modified key definitions. If the SASUSER data set is available then the new key definitions will be saved here. If the SASUSER data set is not allocated, DBS will use the WORK data set to save the new key definitions. This enables users without a SASUSER data set to modify the keys and have them remain in effect for the duration of the session.

As you briefly saw in the AUTOEXEC example, the display manager provides access to OS data sets with the INCLUDE and SAVE commands. COPY is a valid alias for the INCLUDE verb. Both sequential and partitioned data sets are supported as are fixed and variable length records. SAS, source statements or any kind of data can be copied into the program editor by issuing the INCLUDE or COPY command. Newly copied statements will be appended to existing records already in the program editor unless the TOP operand is specified.
You can copy in the AUTOEXEC file for editing by typing:

```
Command •••
```

After the COPY command is completed the screen will look like this:

```
Program Editor
```

The SAVE command is just the opposite of COPY. The entire contents of the program editor is written to the specified Dname.

You can see that the AUTOEXEC file is about to be modified.

```
Command •••
```

To save this member back into the partition data set type:

```
Command •••
```

331
At completion you will receive the message "Member 'autoexec' replaced" because the member autoexec already existed in the partitioned data set.

If the command fails because of a TSO syntax error, you will be prompted by TSO in line mode for the correction. If the command is accepted by TSO and fails for some other reason, you will receive a message with a reason code on the display manager. If you type just TSO you will enter TSO sub-command mode. From this point you are communicating directly with TSO and will be prompted by the characters "TSO". To exit TSO sub-command mode, type END or RETURN to return to the display manager.

The display manager also provides the user with the ability to print the SAS log, procedure output, the program editor window and the recall stack. From any command-line just type PRINT and the entire contents of that "stream" will be printed on the default system printer. All accumulated procedure output will be printed.

The FIND and CHANGE commands are probably the most frequently used commands when editing large files. FIND is used to search for a specific character string. Remember to enclose the string in quotes if it contains embedded blanks or special characters. CHANGE is used to change one or more occurrences of string1 to string2.

For example,

```
CHANGE abc xyz
```

will change the first occurrence of 'abc' to 'xyz'. Repeat find and repeat change are also available. Both FIND and CHANGE support the following operands: NEXT, FIRST, LAST, PREV, ALL, WORD, SUFFIX and PREFIX. FIND works from all windows, but CHANGE is restricted to the program editor window only.

DMS help members have been added to the existing full-screen help facility. The HELP command is available from any command-line and can also be invoked by pressing the HELP function key. HELP without an operand will display the main help menu and offer major topics for selection.

Notice that the text remains in the edit window after a SAVE command. The SASEXEC data set is allocated by the clist that was used to invoke this SAS session, but if (for the sake of argument) you had forgotten to do this, you could issue a TSO or X command to allocate the data set. There are two methods for getting this command to the SAS System. You can either type this command in the program editor and issue the submit command or you can issue it directly from a command-line. The X command allows you to enter host commands regardless of SCP type. Specifying an operand on, for example, the TSO command, will issue that specific TSO command. If the command is successfully completed, the command will disappear from the command-line and the screen will be refreshed.

Here is an example of an allocate command issued from the command line.

```
allocate dajmisc.data
```

```
Program Editor
```

332
A specific topic can be selected by typing HELP topic and pressing enter. In this case, the AUTOEXEC command.

Once you are in the help facility, pressing the HELP function key, or typing HELP HELP, will display the current key definitions for the help facility and the valid keywords that can be specified.

In conclusion, I would like to say that this paper is by no means technically complete; instead the intent is to acquaint the new display manager user to some of the major functions and capabilities that are available. The display manager helps to demonstrate that "full-screen" and "interactive" are the new way to run the SAS System.

Here is the main help menu.

<table>
<thead>
<tr>
<th>Command</th>
<th>Help text</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELP</td>
<td>Enter END to return to SAS Help is currently available for the following keywords:</td>
</tr>
<tr>
<td>OPTIONS SAS data set name options</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>PROCEDURES output formats</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>INFORMAT input formats</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>MENU macro language</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>SHEET installation notes</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>OPTIONS SAS system options</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>PROC SAS procedures for which HELP is available</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>RENAME information on your computing installation</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>HELPTOP information for your computing installation</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>OPTIONS statements for which HELP is available</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>VARLIST abbreviations for variable lists</td>
<td>HELP HELP</td>
</tr>
</tbody>
</table>

For more information enter HELP keyword or HELP HELP.

Type HELP HMS to view the HMS commands that are documented in the help facility.

Here is the main help menu.

<table>
<thead>
<tr>
<th>Command</th>
<th>Help text</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELP</td>
<td>Enter END to return to SAS Help is currently available for the following keywords:</td>
</tr>
<tr>
<td>OPTIONS SAS data set name options</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>PROCEDURES output formats</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>INFORMAT input formats</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>MENU macro language</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>SHEET installation notes</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>OPTIONS SAS system options</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>PROC SAS procedures for which HELP is available</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>RENAME information on your computing installation</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>HELPTOP information for your computing installation</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>OPTIONS statements for which HELP is available</td>
<td>HELP HELP</td>
</tr>
<tr>
<td>VARLIST abbreviations for variable lists</td>
<td>HELP HELP</td>
</tr>
</tbody>
</table>

For more information enter HELP keyword or HELP HELP.

Once you are in the help facility, pressing the HELP function key, or typing HELP HELP, will display the current key definitions for the help facility and the valid keywords that can be specified.

In conclusion, I would like to say that this paper is by no means technically complete; instead the intent is to acquaint the new display manager user to some of the major functions and capabilities that are available. The display manager helps to demonstrate that "full-screen" and "interactive" are the new way to run the SAS System.