Current and Future Enhancements of Version 6 of the SAS® System under CMS
John W. De Boskey, SAS Institute Inc., Cary, NC

ABSTRACT

Version 6 of the SAS System has a fresh, new look. Increased interactivity and a "point-and-click" environment will help novice users as they create their first SAS applications. Expert programmers can also realize an increase in productivity due to the addition of pop-up windows, added language syntax, and increased procedural functionality. In addition to the features currently included in Release 6.06 of the SAS System, there are many interesting projects scheduled for future releases of the SAS System under CMS. This paper discusses some of the current enhancements and future projects scheduled within Version 6 of the SAS System under CMS.

INTRODUCTION

This paper discusses the enhancements that have been made to the SAS System in the language, sequential, and performance areas. In the language arena, enhancements to the LIBNAME & FILENAME statements, transparent access to different type data sets via dataset library engines, CMS & X command resolution, the SORT procedure, and graphic windows are discussed. Sequential enhancements include standard label tapes, multi-volume support, and other associated options. With respect to performance, the SAS System now has full VM/XA®-support, vector processor support, user-created formats and informats in catalogs, and new, easier to maintain VM/XA®-non-VM/XA®-support. A list of future compatibility enhancements and project ideas is also provided for reference.

RELEASE 6.06 LANGUAGE ENHANCEMENTS

LIBNAME & FILENAME Statements

Version 6 of the SAS System provides two new language constructs to enhance I/O performance and portability; the LIBNAME and FILENAME statements. The LIBNAME statement is used to access specific SAS libraries. Similarly, the FILENAME statement is used to access specific CMS external files.

The LIBNAME statement enables the user to write more portable SAS applications by providing a common syntax for redirecting access to specific SAS libraries. This syntax is basically the same for all hosts running Version 6 of the SAS System. The LIBNAME statement also provides an interface for creating and accessing sequential format SAS libraries. The format of the LIBNAME statement is:

LIBNAME libref <engine> <physical-name> <options>;

You can specify the physical location of the library through the physical-name parameter. For example, by specifying the fileref member of an accessed disk, the user can force the SAS System to create a new SAS library or read an existing SAS file that resides on a specified disk. If the physical name parameter is specified as "filetype filemode", the SAS library is aliased to use a CMS filetype other than the default. When using engines that access flat files or data files created by other vendor products, the "filetype filemode" format of the physical name should be used. XPORT, BMDP® and OSIRIS® are examples of such engines. Finally, specifying a physical name of "TAPn" will cause the SAS System to read and write a SAS library in sequential format on the specified virtual tape device.

The engine parameter of the LIBNAME statement gives the user the ability to specify the I/O engine that should be associated with a SAS library. Every SAS library must be associated with an engine that is responsible for reading and writing the members of the library. If an engine is not specified in the LIBNAME statement, the SAS System attempts to determine the format of the SAS library. If the library doesn't exist, an appropriate default or BASE engine is chosen.

Performance gains can be realized if the LIBNAME statement specifies the location and engine associated with the SAS library. Specifying the fileref of the library allows the SAS System to avoid searching all accessed disks to find a specific library. Specifying an engine name avoids the overhead of determining the format of a given SAS library. Librefs defined by the LIBNAME statement are also found before ddnames defined by the CMS FILEDEF command. In all cases, the more information provided, the greater the performance benefit.

The FILENAME statement provides for external files what the LIBNAME statement does for SAS files. The FILENAME statement is used to redirect the input and output of external files from within the SAS System. The format of the FILENAME statement is:

FILENAME fileref <device-type> <physical-name> <options>;

The syntax of the FILENAME statement is very similar to that of the CMS FILEDEF command. Aside from the above syntax, the fileref parameter can alternatively be specified as fileref(member) when referencing a the fileref to indicate that the user wants to access a member of a CMS MACLIB, a CMS TXTLIB, or a partitioned data set (PDS) residing on shared OS DASD. A fileref can be associated with numerous I/O devices, such as DISK, TERMINAL, PUNCH, and PRINTER, through the device-type parameter. By specifying the physicalname parameter, the user can identify the fileref, filetype, and fileref of an external file. The user can also indicate that the file is on tape by specifying 'TAPln' as the physicalname parameter. Options such as BLKSIZE, LRECL, RECFM, and LABEL control the characteristics of external files created and read by the SAS System. The FILENAME statement also enables the user to reassign a ddname previously assigned with the CMS FILEDEF command (provided that ddname was not specified with the PERM option).

The FILENAME statement offers performance gains when the physical-name parameter specifies the location of the external file. This specification avoids the overhead of searching for the external file on all accessed disks. As is the case with the LIBNAME statement, filerefs assigned through the FILENAME statement are found before those created with the CMS FILEDEF command. Also note that the specification of a FILENAME statement when performing I/O on CMS disk files causes the SAS System to use native CMS disk I/O rather than the more costly OS simulation I/O.

The addition of the LIBNAME & FILENAME statements within the SAS System provides the user with an interface for writing more portable SAS applications. Programs that utilize these two constructs usually require only minor modifications to the physical-name parameter in order to run successfully on other systems running Version 6 of the SAS System.

832
Finally, both the LIBNAME & FILENAME statements offer pop-up windows within the Version 6 Display Manager System that the user can use to inspect all currently existing logical assignments. Noninteractive users can specify the LIST, SHOW, or QUERY commands in these two statements to report the details of existing logical assignments. In addition to the LIST, SHOW and QUERY commands, the CLEAR command offers a means of deleting a logical assignment when it is no longer needed.

Transparent Access to SAS Libraries

Transparent access is a term that is used to describe the ability of the SAS System to recognize a particular native library format and to allow the user to access the data sets within the library as if they were in the base library format. From a user's standpoint, no changes must be made in order to access a Version 5 format data set using Version 6 of the SAS System. Converting Version 5 format data sets to Version 6 format can take place at the user's convenience. This eases the transition from one release of the SAS System to the next.

To further explain the concept of transparent access, the following CMS example is offered. Suppose a user has several Version 5 formatted libraries on his or her A disk. The user invokes Version 6 of the SAS System and then invokes the DATASETS procedure on one of the Version 5 format libraries. Remembering that a library name translates into the filetype of the member, the SAS System interrogates all files on the A disk having the given filetype. As soon as the SAS System identifies the format of the native library, the searching is complete. The SAS System loads the Version 5 compatibility engine into memory, and the engine is called upon to process the requests of the DATASETS procedure. In addition to accessing Version 5 data sets on disk, the Version 5 compatibility engine is responsible for accessing libraries created under Version 82.3 and Version 79.6 of the SAS System. The compatibility engine is also used to process Version 5 sequential format data sets and Version 5 data sets located on OS DASD. Access to these libraries is done transparently for the user.

Transparent access ultimately enables users to access any native SAS data library without burdening them with the task of identifying the release of the SAS System that created the library.

CMS & X Command Resolution

IBM, with VM/SP Release 4 CMS, introduced functionality that enabled programs running under CMS to use the full functionality of the CMS search order. Previously, programs that wanted to use the full CMS search order had to be coded with all the search rules, and rely on "hidden" CMS control blocks. The SUBCOM function, introduced in CMS Release 4, provides a single interface for CMS command resolution under program control, relieving CMS programmers (and SAS developers) of the need for reinventing search rules for each CMS program.

Version 6 of the SAS System uses this interface. Now, the CMS/X command, CMS/X statement, and the CMS function behave as if you had typed your CMS command from the CMS ready prompt, with full CMS search rules in effect.

Additionally, ABEND handling has been wrapped around the CMS/X command interface. This allows the SAS System to recover from errors that occur during CMS/X command processing.

SORT Procedure

The interface to the sort utility under Version 6 of the SAS System has been modified to take advantage of EQUALS processing provided by the various sort utilities instead of the SAS System serializing the records and sorting by this field. The EQUALS option allows equivalent by variable records to remain in their input order or, in other words, their input sequence with respect to each other is preserved. For sort utilities that do not support the EQUALS parameter, there is a CMS SAS System option, NOSORTSEQOP, that inhibits the interface from passing EQUALS to the utility.

There is a new PROC SORT parameter, NODUPKEY, that enables the user to omit duplicate key fields. In order for the sort utility to be invoked with this option, the SORTSUMP option has to be set indicating that the utility allows the "SUM FIELDS=" control card.

ABEND handling has been added around the sort utility interface to enable the SAS System to recover from problems that might occur within the external sort program.

There is a new procedure, TRANTAB, in Release 6.06 of the SAS System that creates translate tables that can be used by PROC SORT via the TRANTAB= <table name > procedure option. The TRANTAB procedure is documented in SAS Technical Report P-197.

Example: Creating a Translate Table

The following example uses PROC TRANTAB to create a translate table for converting EBCDIC strings to uppercase. The table that is created will be the current EBCDIC table with the lowercase characters replaced by uppercase characters.

```
proc trantab table=ebcdic;
    list one;
    replace 'a' 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    replace '5' 'STUVWXYZ';
    replace 'j' 'JKLMNOPQR';
    save table=upper;
```

In the previous code segment, the LIST statement displays the translate table in hexadecimal representation. You can display the outbound portion of the table using ONE, the inbound portion using TWO, or both by specifying BOTH.

Windows Environment

Version 5 of the SAS System made a first pass of providing a windowing system for 3270 devices. Version 6 provides extensive windowing support to enhance the user interface. The following is a short overview of current support and future enhancements.

Windows are available, not only for viewing SAS log and listing output, but for displaying graphics, data set information, information about files defined through the LIBNAME & FILENAME statements, and information provided by SAS global statements, such as TITLE and FOOTNOTE.

By specifying
```
options gwindow;
```
as part of your SAS application, or specifying GWINDOW at start-up time, graphics output can be sent to a special graphics window. This graphics window can then be resized and moved about on the display. GWINDOW is the default setting. For Version 5 compatibility, NOGWINDOW should be specified.

You can open multiple windows and display them concurrently. Two other command line options to change the window format are TILE and CASCADE which cause SAS to treat windows in a tiled or cascaded format. The UNTILE, UNCASCADE, or RESIZE commands allow the windows to be put back into their original format. Also provided are pull-down menus and the ability to icon-ize windows, which leaves the windows open, but takes up much less space on the display.
If you have an IBM 3179 or 3472 terminal, you are also able to attach a mouse and gain much more flexibility from DMS. You can use the mouse to select items from the pull-down menus and requester windows.

Currently, as a future enhancement, investigation is underway to ascertain the feasibility of using the X Window System from MIT. This will require having IBM’s Transmission Control Protocol/Internet Protocol (TCP/IP) for VM. The X Windows System is an application program interface that allows an application access to a bit-mapped display on a workstation or a dedicated X-Windows terminal connected to the CMS system via a Local Area Network.

RELEASE 6.06 ENHANCEMENTS - SEQUENTIAL

Tape Usage

Tape support under Release 6.06 of the SAS System is designed to take advantage of tape management support available in the BSAM access method under CMS. If an external tape management system uses the DMSTVI interface, multivolume and multivolume facilities can be exploited by your SAS application.

The FILENAME & LIBNAME statements offer facilities and options that can be employed to replace the CMS FILEDEF command. New options that apply to CMS tape usage are BLKSIZE =, DENSITY =, EOV =, LABEL =, LEAVE =, SYSPARM =, Track =, and VOLID =. These options are designed to work similar to the ones that are offered in the CMS FILEDEF command.

For the most part, any operation that can be done with disk files is also available from tape. One exception to this rule is that the LIBNAME statement does not allow the user to use the ‘TAPn’ form of the physical name when you specify the TRANSPORT engine. However, for Release 6.06 of the SAS System, the work-around that will allow you to create sequential transport format files is as follows:

```
FILENAME & TAPE 'TAP1' LABEL=SL 1;
LIBNAME & TAPE; 
```

The TAPECOPY, TAPECMP, and TAPEDLABEL procedures are not available at this time and will be implemented in the future.

The VMTAPE tape management system, a product of the Systems Center, has facilities for deferring tape mounts until the tape file is opened. The SAS System supports this ability through the SYSPARM operand on the LIBNAME and FILENAME statements. By use of the deferred mount capability, applications can reference tape files with an OS-style dataset name instead of coding volume serial numbers.

Example: Multivolume Tape Data Sets

When an external tape management facility is available, you can issue the following SAS statements to create a data set that can span multiple volumes:

```
x LABELDEF <ddname> FID ? VOLID ?; 
LIBNAME <ddname> TAPE 'TAPn' LABEL=SL 1; 
data <dbname, dataset> [FILEDEF=RAW];  
... data step statements ... 
run; 
```

The tape management system requests a FILEID and then the VOLID.

example FILEID is: userid.<ddname>
example VOLID is: SCRATCH (since we want to create one)

RELEASE 6.06 ENHANCEMENTS - PERFORMANCE

XA Exploitation

Version 6 of the SAS System is completely XA compatible. Version 6 is capable of using the much greater address space that an XA system supports, and it loads and runs above the 16-Meg line. It is now also practical to install larger portions of heavily used images into segments since a larger linear address space is available. Along these lines, SAS Institute has provided ALL606 LLIST, a segment loadlist control file that loads ALL of SAS into segments above the 16-Meg line. The linear layout of the ALL606 LLIST file ensures that all images can be used from segments simultaneously, unlike the overlay structure that is used when loading segments below the 16-Meg line. ALL606 can also be edited to remove infrequently used images to reduce space. The functionality of the ALL606 LLIST is possible due to the fact that Version 6 is composed entirely of reentrant code.

Vector Facility

IBM’s Vector Facility (VF) is an optional feature of all models of 3090 processors. The Vector Facility is an extension of the real processor and is designed to provide significantly increased levels of performance for computationally intensive applications. If a real VF exists, and if the VM user directory does not explicitly state otherwise, then a virtual VF is automatically defined to the virtual machine at logon time.

The option that controls whether or not the SAS System uses an existing VF is VECNOVECTOR, with the default being VECTOR. The (NO)VECTOR option is documented in the CMS Companion and is in the Help subsystem as a host option. It can be specified in a configuration file or as a SAS invocation option.

There are several SAS procedures that use the Vector Facility to varying degrees. Our testing has shown that use of the Vector Facility can decrease execution times by as much as 50%. The decrease in time your application will see depends on its complexity. A simple task with little data will receive little benefit, but a more math-intensive task will see greater benefits.

Some background information concerning the Vector Facility:

- There are scalar instructions and there are vector instructions.
- The Vector Facility is an optional feature of all models of the 3090 processor.
- The Vector Facility is an extension of the real processor and is designed to provide significantly increased levels of performance for a computationally intensive application.
- The Vector Facility is a synchronous instruction processing facility capable of high-speed manipulation of fixed-point and floating-point data.

scalar A simple data item. It can be a floating-point number, a binary integer, or logical data.

vector A linearly ordered collection of scalars. Each scalar is an element of the vector. All elements are of the same type.

- There are scalar instructions and there are vector instructions.
- The Vector Facility is an optional feature of all models of the 3090 processor.
- The Vector Facility is an extension of the real processor and is designed to provide significantly increased levels of performance for a computationally intensive application.
- The Vector Facility is a synchronous instruction processing facility capable of high-speed manipulation of fixed-point and floating-point data.
In Version 6 of the SAS System, formats and informats created by the FORMAT procedure are stored in SAS catalogs. The catalog name is decided by the user’s specification of the LIBRARY = state­ment in the FORMAT procedure. This ability to collect user-written formats and informats into a single SAS catalog gives the user a simple method of building format and informat libraries. Likewise, the user is able to manage these catalogs using normal SAS utilities such as the CATALOG procedure or the CATALOG window avail­able in Version 6 of the SAS System.

Previous releases of the SAS System under CMS stored formats and informats as loose TEXT files on the user’s A disk. The user was responsible for each individual TEXT file or could optionally collect the TEXT files into a CMS TXTLIB file. This process was confus­ing and cumbersome for many users. Storing user-written formats and informats in SAS catalogs provides an easy interface for format and informat management.

For example, suppose the user creates two SAS catalogs containing formats and informats (JOB1.FORMATS.CATALOG and JOB2.FORMATS.CATALOG) and places these catalogs on the A minidisk.

\[
\text{PROC FORMAT LIBRARY=JOB1;}
\]

... format and informat definitions ...

\[
\text{PROC FORMAT LIBRARY=JOB2;}
\]

... format and informat definitions ...

In order to access the formats in JOB1.FORMATS.CATALOG, use the LIBNAME statement to assign the LIBRARY libref to the JOB1 library as follows:

\[
\text{LIBNAME LIBRARY 'JOB1 A';}
\]

At this point, all of the formats and informats contained in JOB1.FORMATS.CATALOG are made known to the SAS System. Since only one library can be assigned to the LIBRARY libref at a time, the user must change the definition of the LIBNAME statement in order to access the formats and informats contained within the JOB2 library as follows:

\[
\text{LIBNAME LIBRARY 'JOB2 A';}
\]

For reasons of compatibility, the SAS System will continue to look for formats and informats in loose TEXT and specified CMS TXTLIB files. However, this search begins only after a search for the SAS catalog pointed to by the LIBRARY libref has been exhausted.

Storing formats and informats in SAS catalogs has several key advantages. Because the format of the SAS catalog is known to the SAS System, utilities such as the CATALOG procedure and the new CATALOG window can be used to manage formats and informats present within the catalog. Users can also gain an increase in efficiency since the search for a user-written format or informat begins with the SAS catalog pointed to by the LIBRARY libref.

Segments

Segment support for Version 6 is upwardly compatible with Version 5 of the SAS System, providing at minimum the same functionality as before. A shift of the series indicator in segment names assures that Version 5 and Version 6 segments can co-exist peacefully. Other minor changes include:

- SASC5eG1 module updated and renamed to SASCSE66.
- In LOADLIST files updated & renamed to In LLiST.
- CMSSEGS SAS has been replaced by CMSSEGS EXEC, thus the installation no longer requires the use of SAS before SAS is installed into segments.
- The manual prompting of segment information has been replaced by a CMSSEGS control file; however, prompting is still available if a "?" is coded.
- The CMSSEGS program automatically generates a DMKSNT file for non-Bimodal systems or an EXEC with segment definitions for Bimodal systems.

Another area where segment support has been refined is the load/overlay levels for SUPERVISOR, PROC1, and PROC2. These levels can now be loaded noncontiguously by specifying the load address of each level on the LOADADDR card in the CMSSEGS control file. This was previously done with boundary cards, making it much more difficult to directly specify the starting address of each overlay level.

CONCLUSION

Release 6.06 of the SAS System on CMS presents many new and different capabilities to both the SAS System user and SAS System applications programmer. New levels of portability insure that an application can be migrated from the development platform to the production platform with a minimum of time and effort. Enhanced performance and greater functionality mean greater productivity and better throughput for your application. Version 6 of the SAS System is an exciting new development system and gives users the functionality they require in today’s world.

SAS is a registered trademark of the SAS System.

IBM is a registered trademark and VM/XA is a trademark of IBM Corporation.

BMDP is a registered trademark of BMDP Statistical Software, Inc.

OSIRIS is a registered trademark of Columbia Computing Services.

X Window System is a trademark of the Massachusetts Institute

835