

# Getting the Most Out of the CMS Operating System with SAS® Software

Matt Aaronson, SAS Institute Inc., Cary, NC

## ABSTRACT

Data access is improved in Release 6.07 of the SAS System for the IBM® CMS operating system with SAS/SHARE® and SAS/CONNECT™ software, as well as access to VSAM and VSE files. Data management is improved with the provision of the TAPECOPY and TAPELABEL procedures. Data presentation and data analysis are extended with further enhancements to support IBM 3270 devices. These features, along with the new interface to the SAA REXX interpreter, performance improvements, and tips for tuning will help your productivity using the SAS System.

## INTRODUCTION

In every kind of organization, the need to capture and effectively analyze data continues to become more crucial to success. Release 6.07 of the SAS System on CMS provides extensions to functionality available with Release 6.06 to even better meet those needs.

For data capture applications that have multi-user contention problems (as many of them do), SAS/SHARE software provides relief. Advances in the area of data access expand the horizon of what data you can get to with the SAS System by providing connectivity to more platforms and more forms of physical data. The SAS System's power to analyze data continues to grow with usability and flexibility enhancements like the REXX macro interface, better device support, and improved performance.

This paper should help you better understand what's new and improved in Release 6.07 of the SAS System on CMS, and how you can better take advantage of it.

## DATA ACCESS ADVANCES

### Access to Version 5 SAS Data Sets on MVS and VSE Shared DASD

From CMS you can read Version 5 SAS data sets that are on MVS or VSE shared DASD. Once you have done the appropriate CP LINK, and CMS ACCESS, all you have to do in your SAS program is specify the DSN= option on the LIBNAME statement. For example if you have a shared VSE disk accessed as filemode F, you would issue:

```
LIBNAME myvselib V5 'f' DSN='sas.518.userdbc.private.lib'
```

Then you can treat the members of that library just like a native CMS SAS data library. Note that only read access is supported.

### Access to VSAM Files

You can read CMS and VSE VSAM data files with the SAS System on CMS. To do this you need to link and access the appropriate disk, and identify the VSAM catalog and data set to the SAS System by issuing the following CMS commands:

```
DLBL IJSYSCT filemode DSN catalog-dataset (PERM  
DLBL fileref filemode DSN VSAM-file-dataset (PERM
```

You do not have to be in the CMS/DOS environment (using SET DOS ON) when you issue the DLBL commands. Nor do you have to use the ASSGN command and the SYSxxx operand with the DLBL command. In fact, you probably don't want to do any of these

things since you can't enter the CMS/DOS environment from within the SAS System. Note that only read access is supported for now. Write access availability is targeted for the first maintenance release.

### Physical File Concatenation Support

There are three styles of physical file concatenation that are now supported in Release 6.07.

### SAS Source Input

You can now invoke several SAS programs back to back without having to issue:

```
SAS saspgm1  
SAS saspgm2  
SAS saspgm3
```

In SAS Release 5.18, you could accomplish this by issuing:

```
SAS saspgm1 saspgm2 saspgm3
```

With SAS Release 6.07 you can do this by issuing:

```
SAS (SYSIN=('saspgm1 SAS' 'saspgm2 SAS' 'saspgm3 SAS'))
```

One thing that this method allows you to do is dynamically include instream data files using the CARDS statement. For example if the file 'SASPGM2 SAS' ends with a DATA step with the CARDS statement, you could have a file 'SASDATA3 DATA' that contains data to be read in. This allows you more flexibility in writing your SAS programs because you can make them more modular. Another good use of SAS source input concatenation is to improve performance. The benefit of this is that you reduce the number of times the SAS System has to go through initialization processing.

### External Data Files

You can now associate a list of files with a fileref for the purpose of reading them sequentially. You can do this by issuing:

```
FILENAME myfileref ('firstfn firstft' 'secndfn secndft' etc)
```

The file characteristics of the files in the list do not have to be the same as long as you don't use any file characteristic options on the FILENAME statement. This feature keeps you from having to merge data files before running the SAS System, or having to do multiple DATA steps to do the merging. This applies to input only. For output, the first file in the list is used.

### Maclibs

You can now associate a list of maclibs (or PDSs) with a fileref for the purpose of searching for a member. You can do this by issuing:

```
FILENAME myfileref ('first maclib' 'second maclib' etc)
```

This is the underlying mechanism that allows concatenation of auto-call libraries with the SASAUTOS= system option.

### Dynamic Input and Output File Changing Capability

You can now change the name of input and output files dynamically in the DATA step using the FILEVAR= option of the INFILE and FILE statements. An example of using this feature with the INFILE statement is to do an X LISTFILE with the EXEC option to create a list of input files. Then you can drive off the resulting list to read each of those files in a nested loop. An example of using this feature

with the FILE statement is to create files whose names are based on the values of a variable. These techniques keep you from having to know and hard code the names of all possible input or output filenames that your application might have to use.

#### CMS Shared File System Work Unit Support

CMS Shared File System compatibility is more complete in Release 6.07 because it exploits work units. In the future, SFS support will extend beyond the filemode access implementation and will accommodate pathnames.

#### Concurrent Update Access with SAS/SHARE Software

##### What is SAS/SHARE Software?

SAS/SHARE software manages multi-user concurrent update access to data. The classic example of the kind of problem that SAS/SHARE software handles is an application where two or more users need to make changes to data at the same time. On CMS there are several ways this can be handled. Some of them are

##### CMS minidisk implementation

If you keep the data on a minidisk, your application has to give exclusive update access to the entire minidisk to one user at a time.

##### CMS Shared File System implementation

If you keep the data in the CMS Shared File System, your application has to give exclusive update access to the entire file to one user at a time. While this alleviates the contention problems with other files that are inherent with the CMS minidisk implementation, it still restricts availability to the file. Even without Shared File System, you could write and maintain your own server to do this kind of file management, but unless you write a very sophisticated and robust server, you still have the restriction of exclusive update access at the file level.

##### Transaction building

You can write your application to keep transaction records for each user and apply those transactions to the master file every night. While there are no contention problems with this, your data become more and more out of date during the course of a day.

##### DBMS packages

You can buy a DBMS package to manage concurrent updates, but they're relatively expensive.

##### SAS/SHARE software

While the SAS System is postured to work with any of these implementations, you may not want your applications to have to bear the corresponding restrictions. That's where SAS/SHARE software fits into this arena. If you allow your data to be managed by SAS/SHARE software, then several users can make changes to different records within a file without having to wait for anyone else to finish. This is because SAS/SHARE software manages concurrent updates by giving exclusive access at the record level.

##### How Does SAS/SHARE Software Work?

Instead of giving any one user exclusive update access to a CMS minidisk, SAS/SHARE software delegates that function to a virtual machine that updates the data sets on behalf of users. A virtual machine performing this service is known as a SAS server.

The way the server works is that a SAS procedure, PROC SERVER, is executed. This procedure runs perpetually. Its purpose is to take requests from users and manage contention by keeping track of who is working on which records, in which data sets. Data integrity is maintained by locking out others from updating records already in use. Figure 1 illustrates this concept.

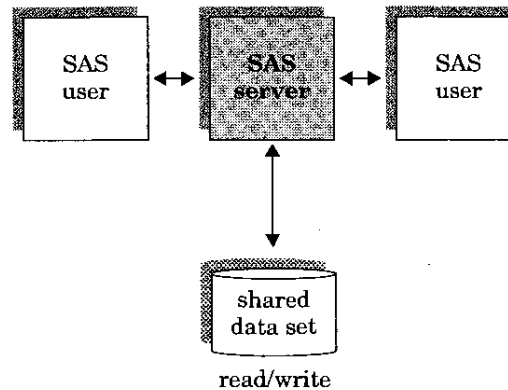


Figure 1 The SAS/SHARE Software Communication Model

While this figure shows only one data set being shared, note that a SAS server can be used to control many data sets. Also, while this figure shows only one SAS server, note that you can have many SAS servers, as long as each data set is controlled by only one server.

##### Using SAS/SHARE Software

There are two basic steps to using SAS/SHARE software:

1. Set up a SAS server. Most of this is one time only work, done by someone known as the server administrator. These steps are well documented in SAS Technical Report P-213, *SAS/SHARE Software for the CMS Environment*. They're being included here only for completeness.

##### VM directory

The VM directory entry needs a couple of special records like IUCV ALLOW, and OPTION MAXCON 1024.

##### SAS program to start the server

You will want a SAS program available on the SAS server that can be invoked to activate the server. It should contain LIBNAME statements and a PROC SERVER statement.

##### Server automation

You will probably want to have the server automatically logged on when your VM system is IPLed. You will need to arrange that with your VM system administrator. You will then want to have the PROFILE EXEC automatically invoke your SAS program.

2. Have your SAS application get to data through the server.

##### SERVER= option on LIBNAME statement

To have your SAS application access data through the server, use the SERVER= option of the LIBNAME statement. For example:

```
LIBNAME inventory SERVER=invserver;
```

##### So, What's New?

While SAS/SHARE software hasn't changed that much, the rest of the SAS System has changed a lot since the last time SAS/SHARE software was available in production. The combination of SAS/SHARE software's ability to manage concurrent updates and the power of the rest of Version 6 of the SAS System could be a productivity boon for many of you.

There are some specific Version 6 features that make a powerful combination with SAS/SHARE software:

#### PROC SQL

The SQL procedure that comes with base SAS software provides a relational language for analyzing data. The relational capabilities of PROC SQL, and SAS/SHARE software's ability to manage concurrent updates means that the SAS System provides a lot of the functionality that you might look for in expensive DBMS software packages.

#### SAS/CONNECT software

With the SAS/CONNECT and SAS/SHARE products together, it is possible for a user on a remote system to access and even update a shared library on CMS. SAS/CONNECT software will be covered in more detail later in this paper but the power of the two products combined is considerable.

#### Screen Control Language (SCL)

Using SCL, which is part of the SAS/FSP\* and SAS/AF\* products, you can write applications that use a single screen to update more than one file at a time with related data. With SCL you can also do more sophisticated validation at data entry time. What this means in combination with SAS/SHARE software is that it's relatively easy to set up robust interactive applications that allow several users each to be updating several shared data sets at the same time.

#### PROC FSVIEW

The SAS/FSP product's FSVIEW procedure is a full-screen application that shows multiple observations at a time. PROC FSVIEW in edit mode now does record level locking based on what observation you are on.

#### WINDOW and DISPLAY statements

The WINDOW and DISPLAY statements in base SAS software give you basic full-screen access to data sets even if you don't have SAS/FSP software. This puts full-screen file access power at everybody's fingertips.

#### WHERE statement

The WHERE statement provides rapid subsetting and searching, especially when used with indexed SAS data sets.

Despite the previous implications, there are some important new things in the SAS/SHARE product for Release 6.07:

#### Security

Servers can now be password protected, and password protection for SAS data sets and views are respected.

#### Catalog support

SAS catalogs can now be updated concurrently. This capability is based on entry type, meaning that some catalog entry types can be shared, some can be shared but with restrictions, and some can't be shared. This is covered in detail in *SAS/SHARE Software, Usage and Reference, Version 6, First Edition*. One example of the benefit of catalog support is that a SAS programmer can update screens and SAS/AF entries without having to kick everyone off the system. Another example is that you can now have concurrent access to different letters in an FSLETTER catalog.

#### Explicit locks

The LOCK statement allows you to acquire, list, or release locks on a SAS data library, SAS data set, SAS catalog, or SAS catalog entry. While you have a lock on

one of these data objects, no other user can read, update, or obtain a lock on it. Explicit locks provide several benefits:

- You can use locks to get and keep exclusive control of data throughout multi-step programs.
- Locks facilitate concurrent access of some catalog entry types.
- When a LOCK statement is issued for any data object below the library level, an implicit lock is created for all higher levels. Implicit locks are internal contrivances that prevent others from locking any of those higher levels while you've got something locked at a lower level. Implicit locks do not prevent others from locking or updating other branches in the structure.

#### Capacity

In previous releases of SAS/SHARE software, SAS servers handled about 20 users, but now we've seen good results with some applications of more than 80 users. Your mileage may vary, but you can now use SAS/SHARE software for bigger applications than before.

#### Performance issues

Any data important enough to share are probably used a lot, which means performance is critical.

#### SAS server performance implications

Be sensitive to the fact that performance improvements that affect a SAS server usually have a bigger total impact than performance improvements for individual users. This is because it's not only the user currently requesting services that is affected, but other users who are queued up waiting for services are affected also.

#### Internal improvements

In addition to the performance improvements that will be discussed in the last section of this paper, there have been internal performance improvements specific to the SAS/SHARE product. For example, a special buffering configuration is used by default on all SAS servers.

#### WHERE statement

Using the WHERE statement whenever possible will improve performance. This is because the WHERE statement reduces the number of observations being transmitted by the server. Note that the subsetting IF statement does not provide the same performance benefit, because it does the subsetting after the server has passed the data to you.

#### SAS data set indexes

SAS data set indexes can be of particular benefit in the SAS server environment in combination with WHERE processing. SAS data set indexes will be discussed in more detail later in this paper.

#### %SHRMACS

void issuing %SHRMACS more than once per session.

#### SAS server environment

Run the server in XA mode, if possible, so that you can run in a large virtual machine and still use segments. If you can't run XA, run in the biggest machine size possible that can still use segments.

#### Additional servers

If performance degrades on a server that's managing many data sets, you can improve performance by creating another server, and moving some of the data sets under the control of the new server.

#### Read-only versus update applications

You should use SAS/SHARE software only for libraries that require concurrent update access. Libraries that only require concurrent read access should be handled outside of the SAS/SHARE product using CP LINK RR and CMS ACCESS. This reduces unnecessary traffic on the SAS servers.

Do not extrapolate from these statements that a library controlled by SAS/SHARE software for concurrent update should be read using CP LINK RR and CMS ACCESS. Attempting to access data controlled by SAS/SHARE software without using SAS/SHARE software will do you more harm than good because you can run into problems with outdated minidisk directories.

#### Connectivity and future directions

The Inter-User Communications Vehicle (IUCV) is the primary access method used by SAS/SHARE software. IUCV allows concurrent update for users on one physical machine. With Release 6.07, the Advanced Program-to-Program Communication (APPC) access method is also available. With the APPC access method, it's possible for VM users to have concurrent update access to data across VM clusters. It also makes it possible for VM users to have concurrent update access to data on MVS and for MVS users to have the same accessibility to data on VM. For example, you can run a SAS application on CMS that accesses SAS files, or even DB2 views, that are controlled by a server on MVS. Another example is that a SAS application on MVS can get to SQL/DS views controlled by a server on CMS. The beauty of this is how easy it is for the application to get to the data. Because this is the first release with APPC support, SAS Institute would like to keep close contact with anyone who uses it. If you are interested, please contact the SAS Technical Support Division for documentation.

The underlying components that support the advances just described are the following:

- The multiple engine architecture of the SAS System.
- For communication across VM clusters, the APPC access method uses TSAF.
- For communication between VM and MVS and future communication with OS/2, the APPC access method uses standard LU6.2 protocol.

With future availability of the Transmission Control Protocol/Internet Protocol (TCP/IP) access method, SAS/SHARE software will provide transparent data access to any platform supported by the SAS System that has Ethernet. This means that the capabilities I've out-

lined as part of the APPC access method represent the beginning of the connectivity breakthroughs that will be forthcoming with the SAS System in the future.

#### Cooperative Processing with SAS/CONNECT Software

SAS/CONNECT software is a product that allows you to do cooperative processing across platforms. Specifically it lets you run processes on other platforms, and *upload* and *download* data across platforms. So SAS/CONNECT software provides a lot more than data access functionality. Release 6.07 gives you connectivity to other CMS systems, OS/2, MVS, VMS, UNIX, and MS-DOS.

From the data access standpoint, this means you can upload and download SAS data sets, catalogs, and external files with any of these platforms without having to convert them to transport files. In fact, you can copy multiple files from the same library across platforms in a single step.

From the remote processing standpoint, this means you can use each platform to run what it is best suited for in a given application.

One way you might use these capabilities is to run SAS programs on the platform with the most horsepower. The value of this is obvious, but there are many other uses for remote processing. What if you do most of your work on a CMS id, and you're more comfortable there, but you need to write a SAS application to run on MVS? You can write your SAS code on your CMS id, and use SAS/CONNECT software to make it run over on MVS to test it. Your output can come directly back to your CMS id, and when you're done, you can upload the code you've written to MVS to run it natively in production.

The description of access methods that I gave in the section on SAS/SHARE software also applies to SAS/CONNECT software. In addition, you may want to note that the connections with VMS and UNIX are done through TCP/IP and TELNET. And the connections with OS/2 and MS-DOS are done through ASYNC, and 3270/HLLAPI protocols.

Future advancements in cooperative processing will make the ability to choose the right tool for the job something you won't want to miss.

#### DATA MANAGEMENT IMPROVEMENTS

The SAS System provides full multivolume and standard label tape support, as long as your tape management system does too.

##### The TAPECOPY Procedure

You can now use SAS software to copy data from tape to tape with the TAPECOPY procedure. You can use this procedure to copy an entire volume, or specific files from one or more volumes, to a single volume. This works for both standard labeled and nonlabeled tapes, and for both 9 track reels and 18 track cartridges under CMS. Be warned that you cannot use PROC TAPECOPY to append to existing files on a tape. It always begins writing at the beginning of the output tape, so any existing files are destroyed, including the label.

##### The TAPELABEL Procedure

You can now use SAS software to list the label information on an IBM standard labeled tape with the TAPELABEL procedure. For each logical file on a tape, this will display information like the data set name, DCB information, and data set history. You can get information for each physical file on a tape by using the MAP option.

## DATA PRESENTATION IMPROVEMENTS

### Icons for Advanced IBM 3270 Devices

Our industry has seen productivity gains with icons. Icons are now available with the SAS System on CMS for 3179G, 3472G, and InfoWindows terminals. For these devices, you get icons instead of words in a box, and you can iconify SAS windows, like the KEYS and OUTPUT windows.

### X Window System Interface

Release 6.07 on CMS has an experimental interface to the X Window System. This lets X terminal users have the same graphical interface to the SAS System as they have with the rest of their environment. In the future, a production quality version will be available on both CMS and MVS. If you want to use it, please contact the SAS Technical Support Division.

## SAS SOFTWARE USAGE ENHANCEMENTS

### Friendlier Attention Handling

With the SAS System on CMS, you can now halt processing from either line mode or full-screen applications without being thrown into CP READ and having to re-IPL CMS. By hitting the break key, your process is interrupted, and you are prompted to see if you want to cancel or continue your current processing. If you choose to halt processing you are returned to control of the SAS System, not CP.

### Accessibility to Global Variables

Passing variables to SAS programs has been made easier with the provision of a mechanism to access global variables. This can be used to pass parameters to a SAS program from any exec that was used to invoke the SAS System (including nested execs), or from any exec that was run prior to invoking the SAS System. The SAS System can get to these variables in two different ways:

```
x = %SYSGET('group-name variable-name')
%LEBT x=%SYSGET(group-name variable-name)
```

### The SAA REXX Macro Interface

You can now exploit the capabilities of the Systems Application Architecture (SAA) REXX interpreter as a macro language in the same spirit that REXX is used by XEDIT macros. These macros, which have a filetype of SASMACRO become an extension of the SAS language, giving new possibilities to the SAS System in the CMS environment.

One function that the REXX macro interface provides is the ability to generate and control SAS code in the same way the SAS macro facility does.

Another potential use of the REXX macro interface is to blend your CMS and SAS environments. The difference between this and just using the X or CMS commands and statements is that with the SASMACRO method, when you are in the CMS environment, it knows about the SAS System. For example, you can set up your SAS session in such a way that you can issue things like the FILELIST command from within the SAS System as either a command or statement. Then from within FILELIST, you can issue SUBMIT next to a SAS program to run it. Several sample SASMACROs come with Release 6.07, and there are other valuable tips in SAS Technical Report P-219, *Changes and Enhancements to the SAS System, Release 6.07, for the CMS Environment*.

While the SAS System on CMS has this functionality, we haven't even scratched the surface of the kinds of things it can be used for. Use your imagination, and let others know what you find.

## PERFORMANCE ENHANCEMENTS AND PERFORMANCE IMPROVEMENT TIPS

Almost every aspect of performance is improved markedly in Release 6.07. For you to reap the benefits of some of the performance improvements you might need to make changes at your site. But most of the improvements are internal to the SAS System, so you don't have to do anything to enjoy them. I'll talk about some specifics, but because things can be different from site to site, the most important thing is for you to have Release 6.07 installed at your site, so you can see the difference for yourself.

### Sequential I/O

Since most I/O is processed sequentially, a lot of attention was given to it in Release 6.07. Sequential I/O performance improvements in Release 6.07 are dramatic. Here are some of the factors contributing to this result:

- The lean file format of Release 6.07 SAS data sets reduces observation length overhead from 4 bytes in Release 5.18, and 12 bytes in Release 6.06, to 1 bit. This means there's less total I/O, resulting in faster processing.
- Code paths for sequential I/O have been streamlined.
- Default buffering is tuned for sequential processing, with override options available for tuning special cases.

### File Indexes

You can improve I/O performance in some situations by indexing SAS data sets. This technique works best for SAS data sets that are frequently read for small subsets of data, or that you frequently read in sorted order. The SAS System will use the index to optimize WHERE processing and BY processing. To understand how to make good use of indexes, see "Effective Use of Indexes in the V6 SAS System" by Stephen Beatrous in the *Proceedings of the Sixteenth Annual SAS Users Group International Conference*.

### Compressed SAS Data Sets

SAS data sets with large observation lengths and lots of white space can be more efficiently processed if they are compressed. Improvements in decompression speed in Release 6.07 should make this feature more attractive.

### The WHERE Statement

Use the WHERE statement whenever possible, and as early in your SAS program as possible to reduce I/O.

### Sort

In Release 6.07, the SORT procedure doesn't resort files that are already in the proper sequence. The list of sort variables is maintained in order to check the sort order and prevent unnecessary sorts before they are done.

### **Execution Speed Performance Improvements**

Most procedures run faster than they did in Release 6.06, and at least as fast as they ran in Release 5.18. One reason for this is because of changes in the procedure/supervisor interface.

### **CONCLUSION**

I hope that this paper has opened your eyes to the possibilities of how the SAS System on CMS can help you and your organization be more productive. I hope that if you have or anticipate the need for concurrent update management, that you understand how the SAS/SHARE product can help you. I hope that if you have connectivity issues that you can see how the SAS System can address

them. And I hope that the performance improvements and usability enhancements that have been made in Release 6.07 of the SAS System give you the incentive to have it installed at your site if it isn't already.

SAS, SAS/AF, SAS/FSP, SAS/SHARE, and SAS/CONNECT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. IBM and OS/2 are registered trademarks or trademarks of International Business Machines Corporation.

Other brand and product names are registered trademarks or trademarks of their respective companies.