



CHAPTER

1

Using the SAS System under CMS

<i>Invoking SAS</i>	3
<i>Using SAS in Noninteractive Mode</i>	4
<i>Using SAS in Batch Mode</i>	5
<i>Interrupting a SAS Session</i>	5
<i>Interrupting a Windowing Environment Session</i>	5
<i>Interrupting an Interactive Line Mode Session</i>	6
<i>Connecting to a SAS Session</i>	6
<i>Specifying CMS File Identifiers</i>	6
<i>Examples of CMS File Identifiers</i>	7
<i>SAS Filenames</i>	8
<i>SAS Filename Restrictions</i>	8
<i>CMS Filetypes Used By SAS</i>	9
<i>Issuing CMS and CP Commands during a SAS Session</i>	9
<i>Accessing System Variables</i>	10
<i>Using the SYSGET DATA Step Function</i>	10
<i>Using the %SYSGET Macro Function</i>	11

Invoking SAS

In order to invoke SAS, you will need to ask your system manager what the *SAS command* (the command that invokes SAS) is at your site. The SAS command executes a CMS EXEC that was written in the Restructured Extended Executor (REXX) language by SAS Institute. It is not documented in any IBM manual.

At many sites, the SAS command is simply **SAS**, but a different command may have been defined during the SAS installation process at your site.

Note: The examples in this book all use **SAS** as the SAS command. △

The SAS command starts a windowing environment session, an interactive line mode session, or a noninteractive session, depending on the default mode of operation that has been specified for your site. To override the default mode of running SAS, you use commands similar to those shown in Table 1.1 on page 4. (Again, the exact commands that you use may be site-specific.)

See “Specifying System Options in the SAS Command” on page 14 for additional information about the SAS command.

Table 1.1 Commands for Invoking SAS

Mode	How to Invoke	How to Terminate	Description
Windowing Environment	<code>sas (dms</code>	<code>bye</code> or <code>endsas</code>	enables you to write and execute SAS programs and to view the SAS log and SAS procedure output in an interactive windowing environment. If it is the default at your site, then you can invoke it by entering <code>sas</code> with no options. For more information about the windowing environment, see help for base SAS.
interactive line mode	<code>sas (nodms</code>	<code>/*</code> or <code>endsas ;</code> statement	prompts you to enter SAS statements at your terminal, one line at a time. This mode is convenient for many applications on a line-mode terminal. For more information about interactive line mode, see help for base SAS.
noninteractive mode	<code>sas myprog</code>	n/a	executes under interactive CMS, but it is called noninteractive because the program runs with no intervention from the terminal. Noninteractive mode is useful for running programs repeatedly or for running programs that have already been debugged. See help for base SAS for additional information.
batch	n/a	n/a	is typically used for SAS programs that take a long time to run, particularly if those programs are executed on a regular basis. See "Using SAS in Batch Mode" on page 5 for details.

Using SAS in Noninteractive Mode

When you use SAS in noninteractive mode, the SAS program (which has been stored in a CMS file) usually runs without your intervention. CMS retains exclusive use of the terminal as the noninteractive SAS job executes. However, if your SAS program invokes windowing procedures, you can interact with the program during program execution.

To invoke SAS in noninteractive mode, enter the SAS command followed by the name of the SAS program file that you want to execute, plus any SAS system options that you want to specify. For example, suppose you have stored SAS statements and data lines in a program file named MYPROG SAS. At the CMS ready prompt, enter the SAS command, the name of the file, and any system options as follows:

```
sas myprog (nodate
```

The program file must have a filetype of SAS. However, you do not need to include the filetype in the filename because the SAS command uses the SAS filetype by default. The program file can reside in any accessed disk or SFS directory; SAS uses the standard CMS search order to locate the file. The file can contain either fixed-length lines or varying-length lines.

A SAS program file can be constructed by any of the following:

- the FILE command
- a CMS editor
- another program.

When you execute a SAS program noninteractively, the SAS log and your procedure output file are each written to a CMS disk file. Normally, these files are written to the disk on which the SAS program file was found. The SAS log is stored in a file of filetype SASLOG. The procedure output (if any) is stored in a file of filetype LISTING.

After the last step of your program executes, the CMS READY prompt shows that CMS is again ready to process commands.

Using SAS in Batch Mode

The details of how to submit programs to CMS batch machines depend on whether you are using the native CMS batch facility, or a batch facility that was developed by a third-party vendor. Ask your system manager or other co-workers for details about batch processing at your site.

In CMS batch mode, you can execute either a noninteractive SAS session or an interactive SAS session:

- A *noninteractive SAS session* is identical to invoking and executing a noninteractive SAS session in CMS interactive mode, except there is no terminal attached. Not having an attached terminal affects only windowing procedures. Some windowing procedures provide different functionality when no terminal is available. For information about particular windowing procedures, see *SAS Procedures Guide, SAS Language and Procedures: Usage*, or specific product documentation. You can route SAS system messages to the console log by specifying TERMINAL as the value of the LOG= system option when you invoke SAS.

Note: This is the preferred method of operation in a CMS batch machine. Δ

- An *interactive SAS session* requires that all SAS statements be queued on the console stack before you invoke SAS. You can queue SAS statements by using the QUEUE instruction in a REXX EXEC. This method is useful if you want to route all SAS system messages to the console log without specifying LOG=TERMINAL.

SAS programs that are run in a CMS batch machine or in any disconnected virtual machine are the same as SAS programs that are run in other modes; no special SAS statements or rules are needed. Also, CMS and CP commands can be used in a program that is submitted to a batch machine.

Interrupting a SAS Session

Under CMS, if you run SAS in either the windowing environment or interactive line mode, you can interrupt SAS processing. The method that you use depends on what type of terminal you use and on certain CP terminal options.

Interrupting a Windowing Environment Session

You can use the BREAK key (PA1) to interrupt your SAS session. (On an SNA terminal, you can also use the ATTN key.) Usually, you must first press the RESET key to unlock the keyboard before pressing the PA1 key.

The first time you press PA1, your terminal is merely placed in CP READ status. Press PA1 again to activate BREAK key processing, or enter a BEGIN command to return to your SAS session instead. When SAS processing is interrupted, a requestor

window asks you to specify an action. In most cases, you can choose either to continue or to terminate the current process.

Interrupting an Interactive Line Mode Session

To interrupt an interactive line mode session, use the BREAK command. BREAK is a CMS *immediate command*—that is, it is processed as soon as it is entered.

Connecting to a SAS Session

Under CMS, you can access or connect to a SAS session in any of the following ways:

3270 terminals

You can use devices that support extended data streams as well as those that do not. See “Terminal Support in the CMS Environment” on page 304 for more information about terminal support.

terminal emulators

Terminal emulators that you can use to access SAS on CMS include Emulus (available from SAS Institute) and EXTRA, Rumba, Communications Manager, and so forth (available from other vendors).*

SAS/CONNECT software

SAS/CONNECT supports cooperative and distributed processing between CMS and OS/390, OpenVMS, UNIX, OS/2, Windows, and VSE. It supports advanced communications protocols (APPC or TCP/IP), enabling local clients that are running SAS to communicate with one or more SAS applications or programs that are running in remote environments. For more information, see *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*.

SAS/SHARE software

SAS/SHARE enables local and remote clients in a heterogeneous network to update SAS data concurrently. It also provides a low-overhead method for multiple remote clients to read local SAS data. For more information, see *SAS/SHARE Software: Usage and Reference*.

Specifying CMS File Identifiers

Under CMS, a file identifier (or *fileid*) consists of three parts: a filename, a filetype, and either a filemode or a directory identifier. Filemodes pertain to files that reside either in a CMS minidisk or in an accessed Shared File System (SFS) directory to which a filemode letter has been assigned. Directory identifiers pertain only to SFS directories. The concept of an *aggregate external file* is new to SAS beginning in Version 7. An aggregate external file is a directory that SAS treats as a single file with aggregate members. Any minidisk, MACLIB, or SFS directory can be specified as an aggregate file.

* Although Emulus software is available from SAS Institute, it operates independently of the SAS System and its applications. It offers enhancements to 3270 terminal operations as well as extended, interactive features for users of SAS graphics applications.

filename

consists of one to eight alphanumeric characters, including the letters A through Z, the digits 0 through 9, and the following special characters: \$ # @ + - (hyphen) : (colon) _ (underscore). The filename identifies the file for the user.

filetype

consists of one to eight alphanumeric characters, including the letters A through Z, the digits 0 through 9, and the following special characters: \$ # @ + - (hyphen) : (colon) _ (underscore). The filetype describes or qualifies the *filename* field. CMS reserves certain filetypes for particular purposes. For example, a filetype of EXEC indicates that the file can contain a program of CMS commands. See the *VM/ESA CMS User's Guide* for information about reserved filetypes.

It is not always necessary to include the filetype in a CMS fileid, because some CMS commands and SAS language elements (such as statements, system options, and the SAS command) assume certain filetypes by default.

filemode

is a two-character field that consists of the filemode letter (A through Z) followed by the filemode number (0 through 6). See the *VM/ESA CMS User's Guide* for definitions of particular filemodes. The filemode letter can refer either to a CMS minidisk or to an accessed SFS directory.

If you do not specify a filemode, then CMS searches the filemodes that you have assigned in alphabetical order.

directory identifier

may consist of a *file pool identifier*, the *root* directory (usually your userid), and (possibly) one or more *subdirectory names*. The file pool identifier must be followed by a colon. See the *VM/ESA CMS User's Guide* for complete information about directory identifiers.

aggregate external file

is a directory that, for minidisks and SFS directories, is identified to SAS with a filemode letter, SFS directory name, or an asterisk, and without a filename or filetype. A MACLIB can also be specified as an aggregate file, using the syntax '*filename* MACLIB <*filemode* | *>'. For further information on aggregate files, see "Identifying an External File" on page 57.

Examples of CMS File Identifiers

- FILENAME statement that assigns a fileref to a file that resides in a minidisk or SFS directory to which a filemode letter has been assigned:

```
filename mysas1 'myprog sas a1';
```

- FILENAME statement that assigns a fileref to a file that resides in an SFS directory:

```
filename mysas1 'myprog sas pool1:mydir.mysubdir'
```

- FILENAME statement that assigns a fileref to an aggregate file that consists of an SFS directory:

```
filename mysas1 'pool1:mydir.';
```

- FILENAME statement that assigns a fileref to an aggregate file that consists of a MACLIB:

```
filename mysas1 'myprog MACLIB *';
```

- FILENAME statement that assigns a fileref to an aggregate file that consists of a CMS minidisk:

```
filename mysas1 'a';
```

SAS Filenames

CMS and SAS use different syntax rules to identify a file:

- CMS identifies a SAS file as it does all files, with a three-part name consisting of the filename, a filetype, and either a filemode or an SFS directory identifier, for example, TASTEST MYLIB A1.
- SAS identifies a SAS file by a two-level name—for example, MYLIB.TASTEST. The first part of the name is the libref; it corresponds to the CMS filetype. The second part of the name is the same as the CMS filename. Assuming that the file is on your A disk, the CMS fileid would be TASTEST MYLIB A1.

SAS name	CMS name
MYLIB.TASTEST	TASTEST MYLIB
libref.filename	filename filetype

The CMS filemode is not included in the two-level SAS name that you specify. However, the filemode can be specified as part of the *physical-name* parameter in the LIBNAME statement. This is discussed in more detail in “LIBNAME” on page 223.

SAS Filename Restrictions

Under CMS, SAS filenames can contain no more than eight characters. In order for CMS file-naming requirements to accommodate SAS file-naming requirements, the following SAS rules are applied:

Table 1.2 SAS File-Naming Rules

Data Member Names		Non-Data Member Names	
Number of Characters	Type of Filename Used	Number of Characters	Type of Filename Used
<= 8	CMS	<=7	SAS name and a prefix character as CMS filename
>8	generated CMS filename	>7	generated CMS filename

If any generated CMS file names are needed, a special CMS file is created that is a directory for the Version 7 library. If this directory is accidentally erased, it will be re-created automatically. The directory file has a CMS filename of :SASDIR and has the

same filetype as the library that it is associated with. All generated filenames are prefixed with :SAS and have a four-digit suffix, as in :SAS0001.

CMS Filetypes Used By SAS

Unless you specify otherwise, SAS assumes default filetypes for the CMS files it reads and writes. Here are the default CMS filetypes used by SAS:

DATA

is the default filetype used when no CMS filetype is supplied by a user or by SAS.

LISTING

is the default filetype for output from SAS procedures.

SAS

is the default filetype for a file containing SAS statements. SAS applies this filetype to configuration files, autoexec files, and files containing SAS programs.

SASLOG

Is the default filetype for the SAS log.

SASSAVE

is the default filetype for SAS autosave files.

SPK

is the default filetype for SAS rendered ZIP files.

SAS uses the following filetypes for special SAS data libraries:

- LIBRARY
- MESSAGE
- SASHELP
- SASTAPE
- SASUSER
- SASTAPE
- WORK

Issuing CMS and CP Commands during a SAS Session

The ability to issue one or more CMS or CP commands during a SAS session or program can be very useful. For example, suppose you are using SAS in interactive line mode and you want to access a file on another user's minidisk. In that case you would want to issue the necessary LINK and ACCESS commands without leaving your SAS session.

Table 1.3 on page 10 lists several ways of issuing CMS and CP commands from within a SAS session or program and tells you where to look for more information. The Notes column indicates whether you can use the method to issue only a single CMS or CP command, or whether you can also use the method to enter CMS subset mode. (From CMS subset mode, you can enter multiple CMS or CP commands.)

Table 1.3 Ways of Issuing CMS or CP Commands during a SAS Session

SAS Language Element	See...	Notes
X or CMS statement	CMS statement“CMS” on page 203	single command or subset mode
X or CMS command	CMS command“CMS” on page 297	single command or subset mode
SYSTEM or CMS function	CMS function“CMS” on page 126	Single command. Returns the return code that was set by the execution of the command.
%CMS or %SYSEXEC macro statement	“Macro Statements” on page 162	single command

When you issue commands in this manner, the responses from CMS or CP are routed to the console, not to the SAS log or procedure output file. In the windowing environment, the display is temporarily cleared, and the response appears on an otherwise blank screen. To return to the windowing environment, press either the CLEAR key or the PA2 key.

Accessing System Variables

Under CMS, SAS provides a mechanism for accessing system variables that have been created with the CMS GLOBALV command. You can access GLOBALV variables directly from SAS by using either the %SYSGET macro function or the SYSGET DATA step function.

The group name and variable name can be any that are valid for the CMS GLOBALV command (see the *VM/ESA CMS Command Reference*). If you omit the group name, the variable is extracted from the group SASV7 if it is found there, or else from the default group UNNAMED. If the variable has not been set via GLOBALV, SYSGET or %SYSGET returns an error.

SYSGET uses the EXECCOMM interface and therefore the length of the variable name is limited to 250 characters. GLOBALV imposes a maximum length of 255 characters on the variable's value. The full length is returned by SYSGET or %SYSGET if the SAS variable is assigned a sufficient length. Otherwise the value returned by SYSGET or %SYSGET is truncated to the SAS variable's length, which is 200 characters by default.

CAUTION:

Using SYSGET or %SYSGET can replace the value of an existing variable in your SAS EXEC. SYSGET and %SYSGET use the EXECCOMM interface, which assigns variables in the SAS EXEC. If the variable you assign with SYSGET or %SYSGET has the same name as an existing variable in the currently active EXEC, the existing variable will receive the value specified by SYSGET or %SYSGET. This change to a global variable could adversely affect other aspects of your SAS session. Δ

Using the SYSGET DATA Step Function

When you use the SYSGET DATA step function to access system variables, you must enclose the string that specifies the group name and the variable name in single quotes:

`SYSGET(group-name variable-name)`

For example, if the group name is MINE and the variable name is TIME, your DATA step might look like this:

```
data test;
  x=sysget('mine time');
  . . . more lines of data . . .
run;
```

Using the %SYSGET Macro Function

When you use the %SYSGET macro function to access system variables, do not enclose the string that specifies the group name and the variable name in single quotes:

`%SYSGET(group-name variable-name)`

For example, if the group name is MINE and the variable name is TIME, your code might look like this:

```
%let x=%sysget(mine time);
```

