



CHAPTER 18

Statements

Statements in the CMS Environment 219

ABORT 219
ATTRIB 221
CMS 221
FILE 222
FILENAME 227
FOOTNOTE 237
%INCLUDE 238
INFILE 238
LENGTH 242
LIBNAME 243
OPTIONS 248
TITLE 248

Statements in the CMS Environment

Portable statements are documented in *SAS Language Reference: Dictionary*. Only the statements that are CMS-specific or that have CMS-specific behavior are documented in this section.

If the SAS statement is also described in *SAS Language Reference: Dictionary*, that information is not repeated here. Instead, each statement description includes "CMS specifics" information that explains how the statement works under CMS. Then you are referred to *SAS Language Reference: Dictionary*. For statements that are entirely CMS-specific, a full description is given.

The "Valid" information that follows the short description of each statement indicates whether the statement is valid only in a DATA step, or anywhere in a SAS program.

ABORT

Stops execution of the current DATA step, SAS job, or SAS session

Valid: in a DATA step

CMS specifics: operating environment behavior

Syntax

ABORT <ABEND | RETURN> <n>;

Details

The ABORT statement causes SAS to stop processing the current DATA step. In all methods of operation, SAS stops processing the current DATA step and prints an error message in the SAS log. However, the way SAS handles the remaining DATA and PROC steps and the observations from the aborted DATA step depends on the mode of operation. Also, in windowing environments or interactive modes of operation the ABORT and STOP statements are analogous; however, in noninteractive mode, the two statements have different effects. See *SAS Language Reference: Dictionary* for information about using the ABORT statement in the various methods of running SAS.

The following options are used primarily in noninteractive processing; however, they can be used in any mode. These options have host-specific characteristics that enable you to terminate the operating environment job step that you are using to execute your SAS program and send return codes that can conditionally execute subsequent job steps.

ABEND

terminates the current SAS job; the return code is 5 or n if n is present.

RETURN

immediately terminates the SAS job or session; the return code indicates an error. This return code can then be used by CMS to conditionally execute later steps in an EXEC. The ABORT RETURN statement results in the return code 4 unless n is specified.

n

allows you to specify a condition code that SAS returns to its calling program or to CMS when SAS stops executing. The value of n must be an integer. Normally a return code of 0 is used to indicate that the program ran with no errors, and return codes greater than 0 are used to indicate progressively more serious error conditions. Return codes 0–6 and those greater than 997 are used by SAS.

You can use the return code from SAS to control execution from within an EXEC.

Before you perform extensive analysis on your data, you can use the ABORT statement to halt execution if the data contain errors. In the following statements, for example, the automatic variable `_ERROR_` is set to 1 if errors occur in the data lines. If any errors are found, SAS stops processing observations. When SAS reads the second data line, it sets `_ERROR_` to 1 and executes the ABORT statement. Data set CHECK is created with one observation. If data set CHECK already exists, it is not replaced if an error is found.

```
data check;
  input ssn 1-9 paycode 11-13;
  if _ERROR_ then abort;
  cards;
111222333 100
aaabbbccc 200
444555666 300
;
run;
```

The return code for the data step above is 3.

ATTRIB

Associates a format, informat, label, or length or all four with one or more variables

Valid: in a DATA step

CMS specifics: LENGTH= specification in *attribute-list*

Syntax

ATTRIB *variable-list-1 attribute-list-1* <... *variable-list-n attribute-list-n*>;

attribute-list

specifies one or more attributes to assign to *variable-list*. One of the attributes that you can specify is LENGTH=<\$>*length*, which assigns the length of variables in the variable list. Under CMS, numeric variables can range from 2 to 8 bytes in length, and character variables can range from 1 to 32,767 bytes in length.

variable-list

specifies one or more variables that will be assigned the attributes in *attribute-list*.

See Also

- *SAS Language Reference: Dictionary.*

CMS

Issues a CMS command, a CP command, or a REXX exec during a SAS session

Alias: X

Valid: anywhere

CMS specifics: all

Syntax

CMS <*command*>;

command

specifies the command issued to the operating environment. You can issue most CMS commands, CP commands, and REXX execs with the CMS statement. CP commands can be preceded by CP, but this is not required.

Details

The CMS statement is similar to the CMS (or X) command, the CMS (or SYSTEM) function, and the %CMS (or %SYSEXEC) macro statement.

Under CMS, CMS is an alias for the X statement. Under other operating environments, the CMS statement has no effect, whereas the X statement is always processed.

SAS executes the CMS statement immediately. A CMS statement can appear within a DATA or PROC step, or between steps. CMS statements within a step are scanned with that step and are executed before the step. CMS statements that occur between steps (that is, after a RUN statement but before the next DATA or PROC statement) are scanned and executed before the next step.

You can use the CMS statement to issue any CMS and CP command, as long as the command does not destroy your working environment. For example, you cannot issue a FORMAT command for the minidisk that contains your current SAS WORK library or an IPL command. You can use the CMS EXEC command to invoke an EXEC, as long as the EXEC does not include commands that will destroy your working environment.

Entering CMS Subset Mode You can also use the CMS statement to go into CMS subset mode during a SAS session. To enter CMS subset mode, place the CMS statement in your program without specifying a system command. (In a windowing environment session, enter CMS subset mode by issuing CMS as a command-line command. See “CMS” on page 320.)

After the statement is executed, you can enter CMS and CP commands as if you were in the subset mode of the CMS editor. In CMS subset mode, you are not prompted with a line number as you would expect during a SAS session. (The CMS subset prompt differs from installation to installation.)

Any commands that you enter in CMS subset mode are processed by CMS or CP; they are not processed as SAS statements. They can be any length.

If you enter CMS subset mode in the middle of a DATA or PROC step, any CMS or CP commands that you enter from your terminal are executed before the rest of the step is scanned.

To return to the SAS session, issue the CMS subset command RETURN. When you return to your SAS session from CMS subset mode, the numbering of SAS statements resumes where it left off.

Note: Because the CMS subset mode expects input from a terminal, it should not be used in the batch environment. Δ

Issuing CP Commands in Subset Mode You can issue any CP command that does not destroy your working environment in CMS subset mode. For example, the DEFINE STORAGE, FORMAT, and IPL commands cannot be used; but the LINK, SPOOL, and TAG commands can be used.

Issuing CMS Commands in Subset Mode The CMS commands that can be used in subset mode are limited to the commands that do not use the user area of memory. If you enter a command that is not allowed in subset mode, a message indicates that it is an invalid subset command. See the *VM/ESA CMS User's Guide* for a list of commands that cannot be issued from subset mode.

See Also

- “Issuing CMS and CP Commands during a SAS Session” on page 9

FILE

Specifies the current output file for PUT statements

Valid: in a DATA step

CMS specifics: *file-specification, host-options*

Syntax

FILE *file-specification* <options>;

FILE *file-specification* | LOG | PRINT;

FILE '*pipe-specification*' PIPE;

file-specification

identifies the external file. See “Identifying an External File” on page 60 for details on the following possible forms of the *file-specification* argument:

'*filename filetype <filemode / *>*'

specifies the filename, filetype, and filemode, respectively.

fileref

specifies the logical name associated with an external file, using 1–8 characters; the first character is either a letter (A-Z) or an underscore (_).

fileref(filename <filetype>)

specifies a member of the aggregate external file identified by the fileref. The aggregate must be an SFS directory or a CMS minidisk; it cannot be a MACLIB. If the filetype is not specified, the filename is used as the CMS filename and the filetype is assumed to be DATA.

LOG

specifies that lines produced by PUT statements are to be written in the SAS log. LOG is the default destination for output lines. Options cannot be specified with FILE LOG statements.

'*pipe-specification*' PIPE

specifies a standard CMS pipeline specification, which you can use to route output to any pipeline device driver. You cannot specify any options if you specify PIPE. For details on using pipeline specifications in your SAS programs, see “Using CMS Pipelines” on page 64 .

PRINT

specifies that lines produced by PUT statements are to be written to the same file as the output produced by SAS procedures. When the PRINT argument is the fileref, SAS uses carriage-control characters and writes the line with the characteristics of a print file. Options cannot be specified with FILE PRINT statements.

options

The *options* specified in a FILE statement control how SAS writes data into a file. In the CMS environment, the FILE statement accepts the following host options. When specifying more than one option, use a blank space to separate each option. Options

are not accepted with FILE LOG, FILE PRINT, or FILE PIPE statements. Valid keywords are:

```
BLKSIZE=
DENSITY=
DISP=
FILEVAR=
LABEL=
LEAVE=
LRECL=
PRINT|NOPRINT
RECFM=
SYSPARM=
TRACK=
VOLID=
```

See “Host Option Values” on page 224 for the values that can be assigned to the preceding keywords.

For information on portable options, see *SAS Language Reference: Dictionary*.

Details

This section provides examples of FILE statement usage and describes the values available for FILE host-options.

Using the FILE Statement Examples of using the FILE statement to write to CMS files follow.

```
data _null_;
  file 'mydata data *';
  put 'Test: writing to an external file';
run;
```

In the preceding example, the FILE statement creates the file MYDATA DATA on the first R/W disk with these default characteristics: LRECL=80, RECFM=FB, and BLKSIZE=960.

```
filename mydata 'old mydata a';
data _null_;
  file mydata;
  put 'Test: writing to an external file';
run;
```

In the preceding example, a fileref is assigned to create the file OLD MYDATA A with these default characteristics: LRECL=80, RECFM=FB, AND BLKSIZE=960. The fileref MYDATA is assigned by a FILENAME statement.

Host Option Values The following file characteristics can be specified in the *options* argument in a FILE statement. Multiple options are separated by blank spaces.

See *SAS Language Reference: Dictionary* for information on portable options.

Note that options cannot be specified if you use FILE LOG, FILE PRINT, or FILE PIPE.

BLKSIZE=value

specifies the buffer size allocated to contain records. Valid values are 1 through 65535.

Records do not have to be blocked; however, because disk or tape activity is reduced when records are blocked, it is more efficient to use blocks. Blocking

records is different when using CMS Native I/O Services than when using OS/MVS Simulation Services.

For CMS Native I/O Services, the following apply:

- No blocking of records is performed in variable-length files. The BLKSIZE= option is ignored, and RECFM=VB is treated the same as RECFM=V.
- For fixed-length files, BLKSIZE= specifies the maximum buffer size to contain records. The default block size is the value of LRECL= times the number of logical-record-length records that fit into 1,024 bytes. For example, a fixed-length file with LRECL=80 by default has BLKSIZE=960. Twelve 80-byte records can fit into 1,024 bytes, taking up 960 bytes.
- CMS does not support blocking for a device-type of TERMINAL.

For MVS Simulated I/O Services, the following apply:

- Blocking is not supported for a device-type of READER.
- Blocking sizes specified for device-type of TAPE, and MVS-shared DASD files must match the block size of the existing file.
- When RECFM=FB, the value of the LRECL= option is the length of the longest record, and BLKSIZE is an integer multiple of LRECL. For example, if the longest record is 70 bytes, LRECL=70, and BLKSIZE=70 x *n*, where *n* is an integer.
- When RECFM=VB, the value of the LRECL= option is the length of the longest record plus 4 bytes, which holds record descriptor information. The value of the BLKSIZE= option can be any value up to 32,760, as long as it is at least 4 bytes longer than the value of the LRECL= option. (These 4 bytes are the block descriptor information.)
- Any time the value of RECFM= option includes A, which specifies a PRINT file, add 1 to the value of the LRECL= option to allow for the column containing carriage-control characters.

DENSITY=*value*

specifies tape density in bits per inch. Valid values include 200, 556, 800, 1600, 6250, 38K.

DISP=*value*

specifies the status (disposition) of the file. Acceptable values are

MOD	specifies that output lines are to be written after any existing lines in the file.
OLD	specifies that any output lines are to be written at the beginning of the file. This is the default. For CMS, this option is identical to the NEW option.
NEW	specifies that any output lines are to be written at the beginning of the file. For CMS, this option is identical to the OLD option.

FILEVAR=*variable*

allows you to dynamically change input and output files in the middle of a DATA step.

LABEL=*value*

indicates the type of label processing for a tape file. LABOFF is the default. Valid values are

- BLP
- LABOFF
- NL
- SL

If LABEL=SL, NL, or BLP is specified, an additional label value *n* can be specified after the SL, NL, or BLP. The value of *n* indicates the file position in a multiframe volume. The default value is 1.

LEAVE=YES

indicates that a multiframe tape is not repositioned at open for LABOFF or BLP processing. For SL tapes, LEAVE=YES does not reposition before label processing. Omitting LEAVE or specifying LEAVE=NO causes a tape to be rewound and repositioned each time a file is opened. (See “Working with SAS Files on Tape” on page 36 for details on tape processing.)

LRECL=*value*

specifies the logical record length in bytes. Valid values are 1 through 65,535.

PRINT | NOPRINT

PRINT writes records to the file with ANSI printer control characters, making the output file a “print” file. Page headings are also produced. NOPRINT writes records without page headings or ANSI printer control characters.

RECFM=*format*

specifies the format of records in the file.

You can choose one of the following record formats:

- | | |
|----|---|
| F | specifies fixed-length records, unblocked. |
| FB | specifies fixed-length records, blocked. |
| V | specifies variable-length records, unblocked. |
| VB | specifies variable-length records, blocked. |
| U | specifies undefined-record format. For CMS disk files, this is the same as V. |

You can use the following values in any of the previously listed formats except U.

- | | |
|---|---|
| A | specifies that the first byte of each record is an ANSI printer control character and that the file is to be handled as a print file. |
| S | specifies that the file contains spanned records (V), or the file contains standard blocks (F). FS, FBS, VS, and VBS files must be assigned with a CMS FILEDEF command. |

SYS Parm=*value*

passes an option string to a tape management system for standard label tapes. Entering a question mark (?) causes SAS to prompt you for option settings.

TRACK=*value*

specifies the tape setting. Valid values are

- 7TRACK
- 9TRACK
- 18TRACK
- 3490B
- 3490C
- 3590B
- 3590C
- XF

VOLID=*value*

specifies the volume serial number to be verified in the tape. If the value contains any special characters, it must be enclosed in single quotes.

See the FILEDEF command entry in *VM/ESA CMS Command Reference*, for more information about these options.

For more information about other options available in the FILE statement, see *SAS Language Reference: Dictionary*.

SAS does not interfere with FILEDEF options that are already specified. Therefore, TAPE options that are specified in the FILE, FILENAME, or INFILE statements that conflict with an existing FILEDEF to TAPE are ignored. See the appendix “Accessing SAS Data Libraries and External Files on Tape” in *SAS Companion for the CMS Environment* for more information.

If a variable-length file is opened for update (possibly in the DATA step) and if the replacement line is not the same length as the existing line, the standard CMS file system action truncates the file at that point. No message is given. The next read to that file returns EOF.

See Also

- *SAS Language Reference: Dictionary*

FILENAME

Associates a SAS fileref with an external file

Valid: anywhere

CMS specifics: *fileref, device-type, file-specification, options*

Syntax

FILENAME *fileref device-type* <'file-specification'> *options*;

FILENAME *fileref* <*device-type*> 'file-specification' | ('file-specification-1'...
'file-specification-n') *options*;

FILENAME *fileref* "TAP*n*";

FILENAME *fileref* PIPE 'pipe-specification';

FILENAME *fileref* | _ALL_ CLEAR;

FILENAME *fileref* | _ALL_ LIST;

FILENAME *fileref* EMAIL '*address*' <*email-options*>;

fileref

specifies the logical name that is to be associated with an external file, using 1–8 characters, with the first character either a letter (A-Z) or an underscore (_). For information on the CMS file naming conventions used by SAS, see “SAS Filenames” on page 8.

device-type

specifies the type of output or input device for the file. Valid values are

DISK
DUMMY
EMAIL
PRINTER
PUNCH
READER
TAPE
TERMINAL

In addition of the device types listed above, SAS also supports the following access methods: CATALOG, EMAIL, FTP, SOCKET, and URL. Access methods are specified the same way as device types in the FILENAME statement. See “Device Types” on page 234 for further information.

'file-specification'

identifies the external file to be associated with the specified *fileref*. The *file-specification* can be in one of the following formats:

*'filename filetype <filemode | SFS-directory | *>'*

If you omit *filemode* or *SFS-directory*, or if you specify an asterisk (*) when referencing the file for input, accessed disks are searched in the standard CMS search order, and the first occurrence of a file with a matching filename and filetype is read. If you omit *filemode* or *SFS-directory* or if you specify an asterisk (*) when referencing the file for output, the file is written on the first R/W disk.

Note that you cannot use an asterisk (*) for either *filename* or *filetype*.

'filename MACLIB <filemode>'

Specifies a MACLIB as an aggregate external file.

*'filemode | SFS-directory | *'*

Specifies a CMS minidisk or SFS directory as an aggregate external file.

*('filename-1 filetype-1 <filemode-1 | * | SFS-directory-1>'... 'filename-n filetype-n <filemode-n | * | SFS-directory-n>')*

specifies a concatenation of files. Note that if file specification includes concatenated files, all files must have the same attributes. The concatenation can contain aggregate external files (SFS directories, MACLIBs, or minidisks).

options

In the CMS environment, the FILENAME statement accepts the following host options. When specifying more than one option, use a blank space to separate each option. All options use a *keyword=value* format. Valid keywords are:

BLKSIZE=
 DENSITY=
 DISP=
 LABEL=
 LRECL=
 RECFM=
 TRACK=
 VOLID=

See “Host Option Values” on page 230 for the values that can be assigned to the preceding options. See *SAS Language Reference: Dictionary* for information on portable options.

EMAIL

specifies the EMAIL device type, which indicates that the specified file is to be sent as electronic mail, using the specified *email-options*.

'address'**email-options**

when the EMAIL device type is specified, the FILENAME statement accepts the following *email-options*:

ATTACH=
 CC=
 EMAILID=
 EMAILSYS=
 SUBJECT=
 TO=
 TYPE=

See “Electronic Mail Option Values” on page 233 for the values that can be assigned to the preceding options.

'TAPn'

specifies the virtual address of the tape device for a file on tape. The value of *n* is an integer identifying the control unit or device. For example, CMS typically associates virtual address 181 with TAP1. If no external file is specified when the device type is TAPE, it defaults to TAP1. This is valid only when device is TAPE. See “Tape Device Values” on page 230 for a listing of valid values.

PIPE 'pipe-specification'

specifies the external file as a CMS pipeline. A pipeline specification is comprised of one or more stages to which you want to route output or from which you want to read input.

The CMS pipeline specification follows the standard syntax for a CMS pipeline; that is it consists of an optional parenthesized list of CMS pipeline options followed by a sequence of one or more stages. The entire specification must be enclosed within a pair of single or double quotes. Comments delimited by /* */ may be contained within the quoted string. These comments are ignored.

For example, this FILENAME statement contains a valid CMS pipeline specification:

```

filename adata pipe
  "(name adata)      /* name the pipeline */
  < archive data a   /* read in a file */
  | unpack          /* unpack if necessary */
";

```

For further information, see “Using CMS Pipelines” on page 64 .

ALL

can be specified to clear or list all currently allocated filerefs.

CLEAR

can be specified to deallocate the specified fileref, or to deallocate all currently allocated filerefs.

LIST

can be specified to list the fileref name and physical name, or to list information about all currently allocated filerefs.

Details

See *SAS Language Reference: Dictionary* for information about portable options and on clearing and listing filerefs.

Tape Device Values

The following table lists the symbolic names and corresponding virtual addresses for CMS tape output devices.

Table 18.1 Symbolic Names for CMS Tape Output Devices

Symbolic Name	Virtual Address	Symbolic Name	Virtual Address
TAP0	180	TAP8	288
TAP1	181	TAP9	289
TAP2	182	TAPA	28A
TAP3	183	TAPB	28B
TAP4	184	TAPC	28C
TAP5	185	TAPD	28D
TAP6	186	TAPE	28E
TAP7	187	TAPF	28F

Host Option Values

The following file characteristics can be specified in the *options* argument in a FILENAME statement. Refer to *SAS Language Reference: Dictionary* for information on portable options.

All options use a *keyword=value* format, and multiple options are separated by blank spaces.

Note that if file specification includes concatenated files, all files must have the same attributes.

BLKSIZE=*value*

specifies the buffer size allocated to contain records. Valid values are 1 through 65535.

Records do not have to be blocked; however, because disk or tape activity is reduced when records are blocked, it is more efficient to use blocks. Blocking records is different when using CMS Native I/O Services than when using OS/MVS Simulation Services.

For CMS Native I/O Services, the following apply:

- No blocking of records is performed in variable-length files. The BLKSIZE= option is ignored, and RECFM=VB is treated the same as RECFM=V.
- For fixed-length files, BLKSIZE= specifies the maximum buffer size to contain records. The default block size is the value of LRECL= times the number of logical-record-length records that fit into 1,024 bytes. For example, a fixed-length file with LRECL=80 by default has BLKSIZE=960. Twelve 80-byte records can fit into 1,024 bytes, taking up 960 bytes.
- CMS does not support blocking for a device-type of TERMINAL.
- Any time the RECFM= option includes an A, specifying a PRINT file, add 1 to the value of the LRECL= option to allow for the column containing carriage-control characters.

For MVS Simulated I/O Services, the following apply:

- Blocking is not supported for a device-type of READER.
- Blocking sizes specified for device-type of TAPE, and MVS-shared DASD files must match the block size of the existing file.
- When RECFM=FB, the value of the LRECL= option is the length of the longest record, and BLKSIZE is an integer multiple of LRECL. For example, if the longest record is 70 bytes, LRECL=70, and BLKSIZE=70 x *n*, where *n* is an integer.
- When RECFM=VB, the value of the LRECL= option is the length of the longest record plus 4 bytes, which holds record descriptor information. The value of the BLKSIZE= option can be any value up to 32,760, as long as it is at least 4 bytes longer than the value of the LRECL= option. (These 4 bytes are the block descriptor information.)
- Any time the value of RECFM= option includes A, which specifies a PRINT file, add 1 to the value of the LRECL= option to allow for the column containing carriage-control characters.

DENSITY=*value*

specifies tape density in bits per inch. Valid values include 200, 556, 800, 1600, 6250, 38K.

DISP=*value*

specifies the status (disposition) of the file. Acceptable values are

MOD	specifies that output lines are to be written after any existing lines in the file.
OLD	specifies that any output lines are to be written at the beginning of the file. This is the default. For CMS, this option is identical to the NEW option.
NEW	specifies that any output lines are to be written at the beginning of the file. For CMS, this option is identical to the OLD option.

FILEVAR=*variable*

allows you to dynamically change input and output files in the middle of a DATA step.

LABEL=*value*

indicates the type of label processing for a tape file. LABOFF is the default. Valid values are

- BLP
- LABOFF
- NL
- SL

If LABEL=SL, NL, or BLP is specified, an additional label value *n* can be specified after the SL, NL, or BLP. The value of *n* indicates the file position in a multfile volume. The default value is 1.

LEAVE=YES

indicates that a multfile tape is not repositioned at open for LABOFF or BLP processing. For SL tapes, LEAVE=YES does not reposition before label processing. Omitting LEAVE or specifying LEAVE=NO causes a tape to be rewound and repositioned each time a file is opened. (See “Working with SAS Files on Tape” on page 36 for details on tape processing.)

LRECL=*value*

specifies the logical record length in bytes. Valid values are 1 through 65,535.

RECFM=*format*

specifies the format of records in the file.

You can choose one of the following record formats:

- | | |
|----|---|
| F | specifies fixed-length records, unblocked. |
| FB | specifies fixed-length records, blocked. |
| V | specifies variable-length records, unblocked. |
| VB | specifies variable-length records, blocked. |
| U | specifies undefined-record format. For CMS disk files, this is the same as V. |

You can use the following values in any of the previously listed formats except U.

- | | |
|---|---|
| A | specifies that the first byte of each record is an ANSI printer control character and that the file is to be handled as a print file. |
| S | specifies that the file contains spanned records (V), or the file contains standard blocks (F). FS, FBS, VS, and VBS files must be assigned with a CMS FILEDEF command. |

SYSPARM=*value*

passes an option string to a tape management system for standard label tapes. Entering a question mark (?) causes SAS to prompt you for option settings.

TRACK=*value*

specifies the tape setting. Valid values are

- 7TRACK
- 9TRACK
- 18TRACK
- 3490B
- 3490C
- 3590B
- 3590C
- XF

VOLID=*value*

specifies the volume serial number to be verified in the tape. If the value contains any special characters, it must be enclosed in single quotes.

See the FILEDEF command entry in *VM/ESA CMS Command Reference*, for more information about these options.

For more information about other options available in the FILENAME statement, see *SAS Language Reference: Dictionary*.

If a variable-length file is opened for update (possibly in the DATA step) and if the replacement line is not the same length as the existing line, the standard CMS file system action truncates the file at that point. No message is given. The next read to that file returns EOF.

Electronic Mail Option Values

The FILENAME statement supports the following values for *email-options*, which are used in conjunction with the EMAIL device type to send mail from within a SAS session:

ATTACH=*'file-specification'*

specifies the names of one or more files that will be attached to the outgoing message. The syntax rules for specifying a single file are as follows:

*'filename <filetype <filemode | SFS-dir | *->>'*

To attach multiple files, use: (*'file1' 'file2' ... 'filer'*), as shown in the following example:

```
attach=('june94 txt .reports.june' 'july94 txt .reports.july')
```

CC=*'secondary-address'*

specifies one or more names and addresses that will receive the message along with those listed in the TO= option. These names and addresses will appear on the CC line of the outgoing message. The following examples show the syntax used for single and multiple addresses:

```
cc='joe@hisaddr.edu'
cc=("mark@work.com" "jane@play.org")
```

To specify a name along with an address, enclose the address in angle brackets (< >):

```
cc='Joseph Smith <joe@hisaddr.edu>'
```

Recipients can also be specified as nicknames that will be resolved from the user's NAMES file.

EMAILID=*'from-user'*

specifies the e-mail address that will appear on the From line in the electronic mail message. The default is the USERID and host name of the user running SAS. For a server application of SAS, you may prefer to supply a specific user name.

To specify the sender's name and address, include the address in angle brackets (< >), as follows:

```
emailid='Joseph Smith <joe@hisaddr.edu>'
```

EMAILSYS=*'stage-specification'*

overrides the value of the EMAILSYS= system option to send the message using a different CMS pipeline stage. See "EMAILSYS=" on page 267 for details.

SUBJECT='message-subject'

specifies the information that will appear on the Subject line of the outgoing message. Use quotes if the Subject line includes special characters. If quotes are omitted, the Subject line is converted to uppercase text.

TO='to-address'

specifies one or more names and addresses that will receive the outgoing message. These names and addresses will appear on the To line of the outgoing message.

The following examples show the syntax for a single address and for multiple addresses:

```
to='joe@hisaddr.edu'
to=('joe@hisaddr.edu' 'jane@play.org')
```

To include a name with an address, enclose the address in angle brackets (< >), as shown in the following example:

```
to='Joseph Smith <joe@hisaddr.edu>'
```

Recipients can also be specified as nicknames that will be resolved from the user's NAMES file.

Device Types

The following devices are supported for reading and writing external files and can be specified in the *device-type* argument of the FILENAME statement. Note that when files are concatenated, all files in the list must be processed by the same external I/O device (for example, all disk or all reader files).

DISK

specifies that the file is to be read from or written to disk. This is the default, which does not have to be specified. This device type uses the native CMS interface.

DUMMY

specifies that output to the external file is to be discarded.

EMAIL

specifies that the file is to be sent out in electronic mail. For details on associated *email-options*, see “Electronic Mail Option Values” on page 233.

PIPE

specifies that the external file is a CMS pipeline, which enables you to receive input from any CP or CMS command or pipeline device driver, or to route output to any pipeline device driver. No options are valid with the PIPE device type. See “Using CMS Pipelines” on page 64 for further details.

PRINTER

associates the fileref with the virtual printer. This device is for output only. Specifying this device type causes the CMS SAS interface to issue a FILEDEF for the device. Use the CP SPOOL and TAG commands to control the destination or use the FSFORMS command to set up printing forms. See “Using the PRINT Command and the FORM Subsystem” on page 89, for information about using the FSFORMS command.

PUNCH

associates the fileref with the virtual PUNCH. This device is for output only. Specifying this device type causes the CMS SAS interface to issue a FILEDEF for the device. Use the CP SPOOL and TAG commands to control the destination.

READER

specifies that the file is to be a reader file. The `BLKSIZE=` option is ignored for this device because CMS does not support blocking. This device is for input only. Specifying this device type causes the CMS SAS interface to issue a `FILEDEF` for the device. If you have more than one file in your reader, the one that is ordered first is read. Use the `CP ORDER READER` and `CP QUERY READER` commands to determine the order of your reader files.

TAPE

specifies that the file is in sequential format on tape. Specifying this device type causes the CMS SAS interface to issue a `FILEDEF` for the device.

TERMINAL

specifies that the file is to be read from or written to the terminal. The `BLKSIZE=` option is ignored for this device because CMS does not support blocking. This device type uses the Native CMS Interface.

File Characteristics for Devices

When you specify a device type on an external I/O statement, you can also specify values for file characteristics in the *options* argument. See “Device Types” on page 234 for a list of the options. If you do not specify a value for a file characteristic, a default is used. See the following table, entitled “Default File Characteristics for Devices Other Than DISK” for information on default values. For DISK defaults, see the following table entitled “Default File Characteristics for Disk Files”.

Table 18.2 Default File Characteristics for Disk Files

Types of Files	LRECL= Option	RECFM= Option	BLK-SIZE= Option	Line Length for Data
<i>Nonprint files:</i>				
DATA step file I/O	80	FB	960	80
FILE from a window	136	V	N/A *	136
<i>Print files:</i>				
LISTING	257	VA	N/A	Determined by <code>LINESIZE=</code> system option; 256 maximum
SASLOG	257 maximum	VA	N/A	Determined by <code>LINESIZE=</code> system option; 256 maximum
PRINT (from a window)	Determined by data; 136 maximum	VA	N/A	LRECL- 1

* N/A means that the `BLKSIZE=` option is not applicable for this type of file.

Table 18.3 Default File Characteristics for Devices Other Than DISK

Device Type	LRECL= Option	RECFM= Option	BLKSIZE= Option
PUNCH	80	FB	960
READER	80	F	80
TERMINAL	255	V	255
PRINTER	133	VA	137
TAPE	32760	U	32760
TEMP	80	FB	960
DUMMY	80	FB	960
PIPE	n/a	n/a	n/a

Using the FILENAME Statement When a FILEDEF Command Has Been Issued

A FILENAME statement reassigns a logical name assignment made by a FILEDEF command, provided that the PERM option is not specified in the FILEDEF command. SAS does not interfere with FILEDEF options that are already specified. Therefore, TAPE options that are specified in the FILE, FILENAME, or INFILE statements that conflict with an existing FILEDEF to TAPE are ignored. See “Working with SAS Files on Tape” on page 36 for more information. If the device specified in the FILENAME statement does not result in SAS issuing a FILEDEF command, then it does not matter whether a CMS FILEDEF command with the PERM option has already assigned the fileref.

See “Device Types” on page 234 to determine whether SAS issues a FILEDEF command for a device. Here are some examples:

The FILEDEF command without the PERM option Assume the following FILEDEF command is in effect:

```
FILEDEF MYDATA PUNCH
```

Then you can issue the following FILENAME statement:

```
filename mydata punch;
```

The FILENAME statement associates the virtual PUNCH with the fileref MYDATA. This works because the devices are the same. Now issue the following FILENAME statement:

```
filename mydata printer;
```

This FILENAME statement references the virtual printer to the fileref MYDATA. The devices are different, but the PERM option is not in the FILEDEF command. Therefore the FILENAME statement that assigns MYDATA to the printer overrides the FILEDEF command. And, because PRINTER is another device that requires a filedef, a new FILEDEF command is issued.

The following FILENAME statement also reassigns the fileref, this time to the disk file TASTEST DATA *:

```
filename mydata 'tastest data';
```

The FILEDEF command with the PERM option Now, assume that the following FILEDEF command with the PERM option is in effect:

```
FILEDEF MYDATA PUNCH (PERM
```

Then you can issue the following FILENAME statement to reference the virtual PUNCH:

```
filename mydata punch;
```

Both the FILEDEF command and the FILENAME statement reference the virtual punch.

Now issue the following FILENAME statement:

```
filename mydata printer;
```

This statement results in the message that the fileref is not available. The FILEDEF command was issued with the PERM option, and the devices that were referenced by the logical name MYDATA do not match.

Now issue the following FILENAME statement:

```
filename mydata 'tastest data c';
```

This FILENAME statement assumes the default device DISK. Notice that device DISK uses the Native CMS Interface. This means that the FILENAME statement does not issue a FILEDEF command. Because a second FILEDEF command to the logical name MYDATA is not issued, the PERM option has no effect. When SAS searches for filerefs, it looks first at those assigned by the FILENAME statement, then at those assigned by the CMS FILEDEF command. So, in this case, instead of associating the fileref MYDATA with the virtual PUNCH, SAS associates it with the DISK file TASTEST DATA C.

See Also

- *SAS Language Reference: Dictionary*

FOOTNOTE

Prints up to ten lines at the bottom of the procedure output

Valid: anywhere

CMS specifics: maximum length of footnote

Syntax

```
FOOTNOTE<n> <'text' | "text">;
```

Details

Under CMS, the maximum footnote length is determined by the value of the LINESIZE= system option, which has a maximum value of 256 in the CMS environment. If the value specified for FOOTNOTE= is greater than the value of LINESIZE=, the footnote is truncated to the maximum line size.

See Also

- *SAS Language Reference: Dictionary*

%INCLUDE

Includes SAS statements and data lines

Valid: anywhere

CMS specifics: *file-specification*

Syntax

%INCLUDE *file-specification*</options>;

Details

The %INCLUDE statement includes and executes SAS statements and data lines in external files, from earlier lines in the same job or session, or from the terminal.

The *file-specification* must be valid for CMS.

This statement has no host-specific options in the CMS environment.

See Also

- “Using the %INCLUDE Statement” on page 83
- *SAS Language Reference: Dictionary*

INFILE

Specifies an external file to read with an INPUT statement

Valid: in a DATA step

CMS specifics: *file-specification, type, options*

Syntax

INFILE *file-specification* < options;>

INFILE LOG | PRINT;

INFILE '*pipe-specification*' PIPE;

file-specification

identifies the file. It can have the following forms:

*'filename filetype <filemode / SFS-directory / *>'*
 specifies the filename, filetype, and filemode, respectively. For details, see
 “Identifying an External File” on page 60.

fileref
 specifies the fileref. See “Identifying an External File” on page 60 for details.

fileref(filename <filetype>)
 gives the fileref of an aggregate external file. The filename given must be enclosed
 in parentheses. The aggregate can be a CMS MACLIB, SFS directory, or CMS
 minidisk. The default filetype for files in SFS directories or CMS minidisks is
 DATA. When referring to members of a MACLIB, do not specify a filetype.

If your CMS installation supports the access of OS/390 data sets from CMS
 (shared DASD), a member of an OS/390 partitioned data set can be read with this
 file specification, but the fileref must be assigned using the CMS FILEDEF
 command.

options

See “Host Option Values” on page 239 for the values that can be assigned to these
 host options:

BLKSIZE=
 DENSITY=
 DISP=
 FILEVAR=
 LABEL=
 LRECL=
 PRINT|NOPRINT
 RECFM=
 TRACK=
 VOLID=

For information about other options that are available in the INFILE statement,
 see *SAS Language Reference: Dictionary*.

'pipe-specification' PIPE

specifies a standard CMS pipeline specification, which you can use to receive input
 from any CP or CMS command or pipeline device driver. You cannot specify any
 options if you specify PIPE. For details on using pipeline specifications in your SAS
 programs, see “Using CMS Pipelines” on page 64 .

Details

The INFILE statement identifies the file to be read; therefore, it must execute before
 the INPUT statement. The INPUT statement reads from the file pointed to by the most
 recently executed INFILE statement.

When you read from more than one external file in a DATA step, you can partially
 process one file, go to a different file, and return to the original file. An INFILE
 statement must be executed each time you want to access a different file, even if you
 are returning to a file previously accessed.

Host Option Values

Note that if file specification includes concatenated files, all files must have the same
 attributes.

BLKSIZE=*value*

specifies the buffer size allocated to contain records. Valid values are 1 through 65535.

Records do not have to be blocked; however, because disk or tape activity is reduced when records are blocked, it is more efficient to use blocks. Blocking records is different when using CMS Native I/O Services than when using OS/MVS Simulation Services.

For CMS Native I/O Services, the following apply:

- No blocking of records is performed in variable-length files. The BLKSIZE= option is ignored, and RECFM=VB is treated the same as RECFM=V.
- For fixed-length files, BLKSIZE= specifies the maximum buffer size to contain records. The default block size is the value of LRECL= times the number of logical-record-length records that fit into 1,024 bytes. For example, a fixed-length file with LRECL=80 by default has BLKSIZE=960. Twelve 80-byte records can fit into 1,024 bytes, taking up 960 bytes.
- CMS does not support blocking for a device-type of TERMINAL.

For MVS Simulated I/O Services, the following apply:

- Blocking is not supported for a device-type of READER.
- Blocking sizes specified for device-type of TAPE, and MVS-shared DASD files must match the block size of the existing file.
- When RECFM=FB, the value of the LRECL= option is the length of the longest record, and BLKSIZE is an integer multiple of LRECL. For example, if the longest record is 70 bytes, LRECL=70, and BLKSIZE=70 x *n*, where *n* is an integer.
- When RECFM=VB, the value of the LRECL= option is the length of the longest record plus 4 bytes, which holds record descriptor information. The value of the BLKSIZE= option can be any value up to 32,760, as long as it is at least 4 bytes longer than the value of the LRECL= option. (These 4 bytes are the block descriptor information.)
- Any time the value of RECFM= option includes A, which specifies a PRINT file, add 1 to the value of the LRECL= option to allow for the column containing carriage-control characters.

DENSITY=*value*

specifies tape density in bits per inch. Valid values include 200, 556, 800, 1600, 6250, 38K.

DISP=*value*

specifies the status (disposition) of the file. The single acceptable value is OLD, which specifies that any output lines are to be read from an existing file.

FILEVAR=*variable*

allows you to dynamically change input and output files in the middle of a DATA step.

LABEL=*value*

indicates the type of label processing for a tape file. LABOFF is the default. Valid values are

- BLP
- LABOFF
- NL
- SL

If LABEL=SL, NL, or BLP is specified, an additional label value *n* can be specified after the SL, NL, or BLP. The value of *n* indicates the file position in a multiframe volume. The default value is 1.

LEAVE=YES

indicates that a multfile tape is not repositioned at open for LABOFF or BLP processing. For SL tapes, LEAVE=YES does not reposition before label processing. Omitting LEAVE or specifying LEAVE=NO causes a tape to be rewound and repositioned each time a file is opened. (See “Working with SAS Files on Tape” on page 36 for details on tape processing.)

LRECL=value

specifies the logical record length in bytes. Valid values are 1 through 65,535.

RECFM=format

specifies the format of records in the file.

You can choose one of the following record formats:

- | | |
|----|---|
| F | specifies fixed-length records, unblocked. |
| FB | specifies fixed-length records, blocked. |
| V | specifies variable-length records, unblocked. |
| VB | specifies variable-length records, blocked. |
| U | specifies undefined-record format. For CMS disk files, this is the same as V. |

You can use the following values in any of the previously listed formats except U.

- | | |
|---|---|
| A | specifies that the first byte of each record is an ANSI printer control character and that the file is to be handled as a print file. |
| S | specifies that the file contains spanned records (V), or the file contains standard blocks (F). FS, FBS, VS, and VBS files must be assigned with a CMS FILEDEF command. |

SYSPARM=value

passes an option string to a tape management system for standard label tapes. Entering a question mark (?) causes SAS to prompt you for option settings.

TRACK=value

specifies the tape setting. Valid values are

- 7TRACK
- 9TRACK
- 18TRACK
- 3490B
- 3490C
- 3590B
- 3590C
- XF

VOLID=value

specifies the volume serial number to be verified in the tape. If the value contains any special characters, it must be enclosed in single quotes.

See the FILEDEF command entry in *VM/ESA CMS Command Reference*, for more information about these options.

For more information about other options available in the FILE statement, see *SAS Language Reference: Dictionary*.

SAS does not interfere with FILEDEF options that are already specified. Therefore, TAPE options that are specified in the FILE, FILENAME, or INFILE statements that

conflict with an existing FILEDEF to TAPE are ignored. See the appendix “Accessing SAS Data Libraries and External Files on Tape” in *SAS Companion for the CMS Environment* for more information.

If a variable-length file is opened for update (possibly in the DATA step) and if the replacement line is not the same length as the existing line, the standard CMS file system action is to fail the write attempt and return an error.

Examples

This example assigns a fileref to a disk file and then uses the fileref in the INFILE statement.

```
filename food 'my data a';
data calories;
  infile food;
  input x y;
run;
```

In this example, member ONE is read from a MACLIB identified by a fileref.

```
filename mymaclib 'use1 maclib';
data banquet;
  infile mymaclib(one);
  input line $char80.;
run;
```

This example reads data from a TAPE.

```
filename x tape 'tap1';
data budget;
  infile x;
  input x y;
run;
```

See Also

- “Using the INFILE Statement” on page 82
- *SAS Language Reference: Dictionary*

LENGTH

Specifies the number of bytes SAS uses to store a variable’s value

Valid: in a DATA step

CMS specifics: length of numeric variables

Syntax

LENGTH < *variable-specification-1* <... *variable-specification-n*>> < **DEFAULT**=*n*>;

variable-specification

has the following form:

variable-1 <... *variable-n*> <\$> *length*

where

variable

names the variable to be assigned to a length.

\$

indicates that the preceding variable or variables are character variables.

length

is a constant that specifies the number of bytes. The *length* argument can range from 2 to 8 bytes for numeric variables and from 1 to 32,767 bytes for character variables.

DEFAULT=*n*

changes the default number of bytes used for storing the values of newly created numeric variables from 8 to the value of *n*. Under CMS, *n* can range from 2 to 8 bytes.

See Also

- “Using the LENGTH Statement to Save Storage Space” on page 113
- *SAS Language Reference: Dictionary*

LIBNAME

Associates a libref with a SAS data library or lists file attributes for a SAS data library

Valid: anywhere

CMS specifics: *libref, engine, physical-name, engine/host-options*

Syntax

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

LIBNAME *libref* *engine* <'segment-name'> SEGMENT=YES <*options*>;

LIBNAME *libref* | _ALL_ CLEAR;

LIBNAME *libref* | _ALL_ LIST;

libref

is a first-level SAS name that identifies the library. The libref can be a maximum of eight characters. The first character must be a letter (A through Z) or an underscore (_). The remaining characters can be any of these characters or numerals (0 through 9). This libref is used to reference the library throughout SAS.

engine

is a SAS name that identifies the engine to be loaded to process the library. The default engine that is used for new libraries is the BASE engine. Valid values for *engine* include the following:

V8 (BASE)

specifies the engine for accessing SAS files in the latest format available. BASE is an alias for V8.

V8TAPE (TAPE) (V8SEQ)

specifies the engine for accessing SAS files in the latest sequential format, either on tape or disk. TAPE is an alias for V8TAPE.

V7

specifies the engine for accessing SAS files Version 7 format.

V7TAPE

specifies the engine for accessing SAS files in Version 7 sequential format, either on tape or disk.

V6

specifies the engine for accessing SAS data sets that were created under the Version 6 releases of SAS.

V6TAPE

specifies the engine for accessing SAS files in the Version 6 sequential format, either on tape or disk.

V5 (V5TAPE)

specifies the engine for accessing SAS data sets that were created under the Version 5 releases of SAS.

BMDP

specifies the engine for accessing BMDP files.

ORACLE

specifies the engine for accessing ORACLE data sets.

OSIRIS

specifies the engine for accessing OSIRIS files.

SPSS (SPSSX)

specifies the engine for accessing SPSS files.

SQLDS

specifies the engine for accessing SQL data sets.

SQLVIEW

specifies the engine for accessing SQL data views.

XPORT

specifies the engine for accessing files in transport format.
See “SAS Library Engines” on page 53 for more details on engines.

'physical-name'

is a quoted string that describes the physical location of the library. The *physical-name* can be specified in the following ways:

'filemode'

specifies the disk-mode letter or the disk-mode letter and optional filemode access number. Use this form for *physical-name* when you want to use a filemode other than the default. If you specify filemode as " or "*", SAS uses the standard CMS search order to locate an existing SAS library. If the library exists on more than one minidisk, then SAS stops searching as soon as it finds a member. Consequently, only the member on that minidisk is used. Otherwise, if the library does not exist, the assignment defaults to the first R/W accessed disk.

'filetype filemode'

specifies the filetype to be used for the library and the disk-mode letter. *filemode* can also include a filemode access number. Use this form for *physical-name* when you want to use the libref as an alias for the filetype.

'filetype sfs-dir'

specifies an SFS directory to be used as a SAS library. Use this form for *physical-name* when you want to use the libref as an alias for the filetype.

'sfs-dir'

specifies an SFS directory to be used as a SAS library.

'filename filetype filemode'

specifies the complete CMS fileid. Use this form for *physical-name* with the BMDP, OSIRIS, SPSS, and XPORT engines.

(*filetype-1* <*filemode-1* | *SFS-directory-1*> . . . *filetype-n* <*filemode-n* | *SFS-directory-n*>)

specifies a concatenation of more than one library that will be accessed by SAS in order of specification using a single fileref. See "Concatenating SAS Data Libraries" on page 33 for more information on concatenation of SAS data libraries.

'TAP*n*'

specifies the tape device for a sequential library. *n* is a hex character from 0 through F.

'segment-name' SEGMENT=YES

If the option SEGMENT=YES is specified, *segment-name* specifies the name of a segment that contains the library. If *segment-name* is not specified and SEGMENT=YES is specified, then SAS uses the libref as the segment name.

engine/host-options

is a list of host and engine options to be applied to this library. When specifying more than one option, use a blank space to separate each option. *engine/host-options* can be specified in the following format:

option-1=value option-2=value

The following list presents valid options for *engine/host-options*. Of these options, only the BLKSIZE= option is meaningful with disk libraries. All of the others are for sequential format libraries only.

BLKSIZE=*nnnnn*

specifies the logical block size in bytes of the file. *nnnnn* must be greater than 0 but less than or equal to 32,760. The default is 32,760.

DENSITY=

specifies the tape density. Valid values are

- 200
- 556
- 800
- 1600
- 6250
- 38K.

If the DENSITY= option is not specified, the tape density will be the density that the tape drive was last assigned.

DISP=MOD

for a sequential library, DISP=MOD specifies that members copied into the library with the COPY procedure will be appended to the end of the library without

checking for duplicate members. By default, when DISP=MOD is not specified, the COPY procedure replaces all existing members in the destination sequential library with the new members copied into the destination library from the source library.

The DISP=MOD option is valid only for sequential libraries using the V6TAPE or later engine. This option applies only to the COPY procedure.

Any duplicate members appended to the end of the sequential library will not be accessed by SAS, since SAS accesses the first member with the specified name. To ensure that all members in the destination library can be accessed by SAS, you can:

- Use the EXCLUDE statement in the COPY procedure.
- Remove unwanted duplicate members after the COPY procedure using the DELETE statement in the DATASETS procedure.
- Rename duplicate data sets using the RENAME statement in the DATASETS procedure.

For details on the DELETE and RENAME statements of the DATASETS procedure, see *SAS Procedures Guide*. For further information on sequential tape processing, see “Working with SAS Files on Tape” on page 36.

DSN=

specifies the OS/390 data set name (one- to eight-byte qualifiers separated by periods, up to 44 characters long, including periods) of a SAS data set residing on OS/390-shared DASD. Enclose the value in single quotes.

EOV=

specifies whether there is automatic limited end-of-volume processing when the end of the tape is sensed on output. Valid values are

- | | |
|-----|--|
| YES | specifies that there is end-of-volume processing. This is the default. |
| NO | specifies that there is not end-of-volume processing. |

LABEL=

specifies the type of tape label. Valid values are

- | | |
|---------|--|
| SL <n> | specifies the IBM standard label. <i>n</i> is the position of the file on a multfile volume. If <i>n</i> is not specified, 1 is assumed. |
| NL <n> | specifies that the tape has no IBM standard labels. <i>n</i> is the position of the file on a multfile volume. If <i>n</i> is not specified, 1 is assumed. |
| BLP <n> | specifies that label processing is to be bypassed, but that the tape is to be positioned before the library is processed. <i>n</i> is the position of the file on a multfile volume. If <i>n</i> is not specified, 1 is assumed. |
| LABOFF | specifies that there is to be no label processing. The tape is not positioned if this value is specified. This is the default. |

LEAVE=

specifies whether a standard label tape is to be moved before processing the label. This option is valid only when LABEL=SL is specified. Valid values are

- | | |
|-----|--|
| YES | specifies that the tape is not to be moved before processing the label. |
| NO | specifies that the tape is to be rewound before processing the label. This is the default. |

SEGMENT=

specifies that the library is (SEGMENT=YES) or is not (SEGMENT=NO) contained in a CMS saved segment.

SYS Parm=

passes an option string to a tape management system for standard label tapes. See “Working with SAS Files on Tape” on page 36 for further information.

TRACK=

specifies the tape setting. Valid values are

- 7TRACK
- 9TRACK
- 18TRACK.
- 3490B
- 3490C
- 3590B
- 3590C
- XF

VOLID=vvvvvv

specifies the one- to six-character serial number that identifies the volume. If the vol id is not a valid SAS name, you can use quotes around it to prevent SAS from attempting to parse the vol id as multiple tokens. This option is valid only when it follows LABEL=SL.

Details

The LIBNAME statement can also list or clear librefs. See *SAS Language Reference: Dictionary* for information about using the LIBNAME statement for these purposes.

You can use the LIBNAME statement to list attributes of SAS data libraries. The following output example shows the results of the following LIBNAME statement:

```
libname sashelp list;
```

Output 18.1 Data Library Attributes Listed by the LIBNAME Statement

```
NOTE: Copyright (c) 1989-1999 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Release 7.01 TSxxx
      Licensed to SAS INSTITUTE DATA CENTER, Site xxxxxxxxxx.
NOTE: Running on IBM Model 3090 Serial Number xxxxxx.

1  libname sashelp list;
NOTE: Libref=  SASHELP
      Engine=   V8
      Filefmt=  701
      Physical Name= SAS4HELP SG
NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
```

See Also

- “SAS Files and SAS Data Libraries” on page 28
- *SAS Language Reference: Dictionary*

OPTIONS

Changes the value of one or more SAS system options

Valid: anywhere

CMS specifics: *options*

Syntax

OPTIONS *options...* ;

Details

The OPTIONS statement changes the value of one or more SAS system options. System options are either portable or host-specific. “Summary Table of SAS System Options” on page 308 provides a description of all system options supported in the CMS environment, both portable and host-specific.

Some system options cannot be changed during a SAS session; they can be changed only when you invoke SAS. SAS system options that can be specified in an OPTIONS statement are identified in “Summary Table of SAS System Options” on page 308.

See Also

- “Introduction” on page 13
- *SAS Language Reference: Dictionary*.

TITLE

Specifies title lines for SAS output

Valid: anywhere

CMS specifics: maximum length of title

Syntax

TITLE *<n>* *<'text' | "text">*;

Details

Under CMS, the maximum title length is determined by the value of the LINESIZE= system option. The maximum value of the LINESIZE= system option is 256 in the CMS environment.

If the specified title is greater than the value of the LINESIZE= option, the title is truncated to the specified line size.

See Also

- *SAS Language Reference: Dictionary*

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS[®] Companion for the CMS Environment, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS[®] Companion for the CMS Environment, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-481-0

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute, Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS[®] and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

IBM[®] and DB2[®] are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration.

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

The Institute is a private company devoted to the support and further development of its software and related services.