CHAPTER 17

Statements

Statements in the OS/390 Environment

Portable statements are documented in *SAS Language Reference: Dictionary*. This chapter documents statements that are specific to OS/390 or that have aspects that are specific to OS/390.

ABORT

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

Valid: in a DATA step

OS/390 specifics: action of ABEND and RETURN, maximum value of *n*

Syntax

ABORT <ABEND | RETURN> <*n*>;

The following options are used primarily in batch processing, although they can be used with any method of running SAS. These options have host-specific characteristics.

ABEND

causes abnormal termination of the current SAS job or session. This also terminates the step in your OS/390 job stream that was used to execute your SAS job. Because the step terminates abnormally, the physical files that were allocated to the step are handled as specified by the abnormal-termination condition of the DISP parameter for those data sets. (The DISP parameter is specified in the JCL DD statement or the TSO ALLOCATE command, in the SAS FILENAME or LIBNAME statement, or in the SAS FILENAME or LIBNAME function.)

RETURN

causes an immediate normal termination of the SAS job or session. The step return code (condition code) should be used to indicate the error. To pass the specific return code back to the operating environment, use the *n* option. You can then use this return code in your JCL to conditionally execute later steps in your OS/390 job stream.

n

enables you to specify an ABEND code or a condition code that SAS returns to the operating environment when it stops executing. The value of *n* must be an integer. Under OS/390, the range of acceptable values is from 1 to 4095. If you do not specify a value for *n*, an ABORT ABEND statement returns a user abend 999; an ABORT RETURN statement returns condition code 20.

Details

You can use the ABORT statement to control the conditional execution of OS/390 job steps. For example, depending on the result of the OS/390 job step that executes your SAS program, you may need to either bypass or execute later steps. To do this you can establish a variable in your SAS DATA step program that is set to a particular value whenever an error occurs; in the following example, we use a variable named ERRCODE that is set to 16 if an error occurs in the DATA step. You can choose any variable name and value that are required by your program. Then, use the following ABORT statement, coded in the THEN clause of an IF statement, to cause the OS/390 job step to ABEND if ERRCODE=16:

```
if errcode=16 then abort abend;
```

When the OS/390 job step that is used to execute your SAS job ends (either normally or abnormally), the next OS/390 job step is processed. You could then use the following EXEC statement to conditionally execute that job step if an ABEND occurs. If ERRCODE is not set to 16, then the ABORT statement is not executed, and because an ABEND did not occur the job step is bypassed.

```
//stepname EXEC PGM=your-program,COND=ONLY
```

See Also

- □ *SAS Language Reference: Dictionary*
- □ *OS/390 JCL Reference* from IBM

ATTRIB

Associates a format, informat, label, length, or any combination of these attributes, with one or more variables

Valid: in a DATA step

OS/390 specifics: LENGTH= specification in *attribute-list*

Syntax

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

Details

LENGTH=<$> *length* is one of the attributes that may be specified in the *attribute-list*. The LENGTH= attribute specifies the length of variables in the *variable-list*. Under OS/390, numeric variables can range from 2 to 8 bytes in length and character variables can range from 1 to 32,767 bytes in length.

See Also

□ *SAS Language Reference: Dictionary*

DSNEXST

Checks to see whether the specified physical file exists and is available

Valid: anywhere

OS/390 specifics: all

Syntax

DSNEXST *'physical-filename'*;

'physical-filename'
is the name of a physical file. Quotation marks around the name are optional; however, the data set name must always be fully qualified. In this case, *physical-filename* cannot specify a UNIX System Services file.

Details

DSNEXST is a global statement. The first time the statement is issued, it creates the macro variable &SYSDEXST and assigns a value of 1 to it if the data set exists and is available for allocation or a value of 0 if the data set does not exist.

The following example allocates a data set differently depending on whether the data set already exists or not.

```
%macro mydsn;
   dsnexst 'my.data.set';
   filename myds 'my.data.set'
%if &sysdexst %then %do;
   disp=old;
   %end;

%else %do;
   disp=(new,catlg) space=(cyl,(1,1)) blksize=6160
      dsorg=ps recfm=fb lrecl=80 unit=disk
      volser='MYVOL';
   %end;

%mend mydsn;

%mydsn
```

The next example shows how you can submit some SAS statements if a data set already exists and bypass them if it does not.

```
%macro copylib;
   dsnexst 'my.data.library';
%if &sysdexst %then %do;
   libname mylib 'my.data.library' disp=shr;
   proc copy in=mylib out=work;
   run;
   %end;

%mend;

%copylib
```

In situations where there could be more than one user of the data set, the following example shows how you can use the &SYS99ERR automatic macro variable to distinguish between “data set does not exist” and “data set exists but is not available.”

```
%macro dsexist(loc);
  dsnexst &loc;
  %if &sysdexst=0 and &sys99err=1708
    %then %do;
    %put &loc does not exist;
    %end;
  %else %do;
    %put &loc exists;
    %end;
  %mend;

%dsexist(my.data.set)
```

See Also

□ *SAS Macro Language: Reference*

FILE

Specifies the current output file for PUT statements

Valid: in a DATA step

OS/390 specifics: *file-specification, type, host-options*

Syntax

FILE *file-specification* | LOG | PRINT <*type*> <*options*> <*host-options*>;

file-specification
identifies a file:

fileref
specifies a fileref or the allocated DDname of the file. A fileref must conform to the rules for DDnames. That is, it can consist of one to eight letters, numbers, or the national characters $, @, and #. The first character must be either a letter or a national character.

fileref(*member*)
specifies a member of a partitioned data set, where the PDS or PDSE is specified by the assigned fileref or allocated DDname.

The value of the FILEEXT= system option can affect the way SAS interprets PDS and PDSE member names. See "FILEEXT=" on page 344 for details.

'*physical-filename*'
specifies a physical file, which can be a sequential file, a member of partitioned data set (PDS) or an extended partitioned data set (PDSE), or a UNIX System Services file, using the following syntax:

□ a UNIX System Services file:

```
'/u/userid/raw'
```

or

```
'HFS:raw'
```

□ a fully qualified data set name. For example:

```
'myid.raw.datax'
```

□ a fully qualified data set name with a member in parentheses. For example:

```
'sas.raw.data(mem1)'
```

□ a partially qualified data set name with a period preceding. For example:

```
'.raw.data'
```

□ a partially qualified data set name with a period preceding and a member name in parentheses. For example:

```
'.raw.data(mem1)'
```

□ a temporary data set name. For example:

```
'&mytemp'
```

The value of the FILEEXT= system option can affect the way SAS interprets file specifications for PDS and PDSE files. See "FILEEXT=" on page 344 for details.

See "Specifying Physical Files" on page 13 for more information about partially qualified data set names.

LOG
directs output to the SAS log file.

PRINT
directs output to the SAS procedure output file.

type
specifies the type of file. When you omit *type*, the default is a standard external file. Nonstandard (host-specific) file types that you can specify for OS/390 are

DLI
for IMS-DL/I databases. For information about IMS-DL/I options for the FILE statement, see *SAS/ACCESS Interface to IMS-DL/I Software*.

HFS
for UNIX System Services files, see “Accessing UNIX System Services Files” on page 93.

PIPE
for pipelines in UNIX System Services, see “Piping Data from SAS to an UNIX System Services Command” on page 96.

VSAM
for VSAM files (see “Accessing VSAM Data Sets” on page 92).

options
are portable options that can be specified in the FILE statement. See *SAS Language Reference: Dictionary* for information about these options.

host-options
are host-specific options that can be specified in the FILE statement. You can specify portable options and *host-options* in any order. When specifying more than one option, use a blank space to separate each option. The *host-options* that you can specify depend on what type of file you are accessing. See the following sections for details:

- □ “Standard Host Options for the FILE Statement under OS/390” on page 290
- □ “Host Options for Retrieving Information about Data Sets” on page 292
- □ “Options That Specify SMS Keywords” on page 303
- □ “VSAM Options for the FILE and INFILE Statements under OS/390” on page 292.
- □ “Host-Specific Options for UNIX System Services Files” on page 96

Standard Host Options for the FILE Statement under OS/390

The following options can be used with all external files under OS/390.

BLKSIZE=*value* | BLK=*value*
specifies the block size of the file. Block size is discussed in more detail in “DCB Option Descriptions” on page 300 and in “Overview of DCB Attributes” on page 302.

CLOSE=*keyword*
indicates how a tape volume is to be positioned at the end of the DATA step. Values for *keyword* can be

REREAD	positions the tape at the logical beginning of the data set.
LEAVE	positions the tape at the logical end of the data set.
REWIND	rewinds the tape to the physical beginning of the volume.
FREE	dynamically deallocates the tape volume.
DISP	is implied by the control language.

CSRC
specifies that you want to use the CSRCESRV services (available with OS/390) to compress data on output. For example:

```
data _null_;
   file myfile csrc;
   put ... ;
run;
```

You cannot use this option with an external file that has a fixed-length record format.

DCB=*fileref*
gives the fileref of an external file that was referenced in an earlier FILE or INFILE statement in the same DATA step. SAS uses that file's RECFM=, LRECL=, and BLKSIZE= information for the current file.

LINESIZE=*width*
works with LRECL to specify the maximum number of characters per line or record in print files, nonprint files, and the SAS log. Under OS/390, the range of acceptable values of LINESIZE= is 64 to 256. The default value of the LINESIZE= system option under OS/390 is 132. This default applies only to print files (with carriage returns) or to the SAS log. For nonprint files (without carriage returns) , the value of LRECL= is used in place of the default value for LINESIZE=.

LRECL=*value*
specifies the logical record length of the file. The value specified depends on the access method and the device type. For more information, see the discussion of LRECL= in "DCB Option Descriptions" on page 300 and *OS/390 JCL Reference*.

MOD
writes the output lines following any existing lines in the file. This option overrides a disposition that was specified in JCL or under TSO. It is not valid if the specified file is a member of a partitioned data set (PDS).

NOPROMPT
specifies that if the file that you reference in the FILE statement is unavailable, a requestor window is not displayed and an error is written to the SAS log.

OLD
writes the output lines at the beginning of the file, causing any existing data in the file to be overwritten. This option overrides a disposition that was specified in JCL or under TSO, and it is the default if no disposition is specified. Using OLD is necessary only if you used MOD for the file in an earlier FILE statement and you want to overwrite the file.

PRINT|NOPRINT
specifies whether carriage-control characters are placed in output files. Under OS/390, PRINT adds carriage control characters to the beginning of all lines of output directed to print files and to the SAS log.

RECFM=*record-format*
specifies the record format of the file. Valid values are

F	fixed length records, unblocked.
V	variable length records, unblocked.
FB	fixed length records, blocked.
VB	variable length records, blocked.

U undefined length records, unblocked.

The following values can be appended to the values above:

A the first byte of each record is an ANSI printer-control character.

S if appended to V, the file contains spanned records; if appended to F, the file contains standard blocks.

The following value stands alone; no other values can be appended:

N a format specific to SAS; this format indicates that the file can be processed as a stream of bytes with no record boundaries.

Host Options for Retrieving Information about Data Sets

The following options retrieve information about a data set from the operating environment control blocks. SAS assigns values to the variables that are defined by these options when it opens the data set. It updates the values every time it opens a new data set in a concatenation. These options can be used with all standard external files under OS/390.

DEVTYPE=*variable*

defines a character variable (minimum length 24) that SAS sets to the device type. SAS obtains the device type by using the OS/390 operating environment DEVTYPE macro. For more information, see the IBM documentation for your system.

defines a character variable (minimum length 96) that SAS sets to the Data Set Control Block (DSCB) information from a non-VSAM data set. For more information, see the IBM documentation for your system.

JFCB=*variable*

defines a character variable (minimum length 176) that SAS sets to the Job File Control Block (JFCB). For more information, see the IBM documentation for your system.

UCBNAME=*variable*

defines a character variable (minimum length 3) that SAS sets to the unit name (device address), which is derived from information in the unit control block (UCB).

VOLUME=*variable* | VOLUMES=*variable*

defines a character variable (minimum length 6) that SAS sets to the tape VOLSER or the disk volume serial number, which is taken from the JFCB. The JFCB lists only the first five volume serial numbers, so in the case of a multivolume file, the VOLUME= variable contains the concatenated volume serial numbers (up to five volumes).

VSAM Options for the FILE and INFILE Statements under OS/390

The following options can be used for VSAM files in the FILE statement, the INFILE statement, or both. (Unless otherwise indicated, the option can be used in both.)

BACKWARD | BKWD

tells SAS to read the VSAM data set backwards. (INFILE only)

BUFND=*value*

indicates how many data buffers to use for the VSAM data set.

BUFNI=*value*

indicates how many index buffers to use for the VSAM data set.

CONTROLINTERVAL | CTLINTV | CNV
indicates that you want to read physical VSAM control interval records rather than logical records. This is typically used for diagnostic purposes. (INFILE only)

ERASE=*variable*
defines a numeric SAS variable that you must set to 1 when you want to erase a VSAM record. (INFILE only)

FEEDBACK=*variable* | FDBK=*variable*
defines a numeric variable that SAS will set to the VSAM logical error code. This is similar to the _FDBK_ automatic variable. When SAS sets the FEEDBACK variable, you must set it to 0 to continue.

GENKEY
causes SAS to use the KEY= variable as the leading portion of a record's key. VSAM retrieves the first record whose key matches the generic key. (INFILE only)

KEY=*variable* | KEY=*(variable1 variable2 . . .)*
indicates that direct keyed access is being used to read records either from a KSDS or from an ESDS via an alternate index. Also, the variable contains the key value to be used in the retrieval of a record (input) or the writing of a record (output). (INFILE ONLY)

KEYGE
is used in conjunction with the KEY= option. KEYGE indicates that when KEY= is used in a retrieval request, SAS should retrieve any record whose key is equal to or greater than the specified key. This is useful when the exact key is not known. (INFILE only)

KEYLEN=*variable*
specifies a numeric SAS variable that when used with GENKEY, specifies the length of the key that is to be compared to the keys in the file.

KEYPOS=*variable*
indicates the numeric variable that SAS sets to the position of the VSAM key field. This option enables you to read keys without knowing the key position in advance. This variable is set to the column number (starting from 1).

PASSWD=*value*
gives the appropriate password for a VSAM data set that has password protection.

RBA=*variable*
specifies a numeric variable that you set to the relative byte address (RBA) of the data record that you want to read. The RBA= option indicates that addressed direct access is being used; it is appropriate for KSDS and ESDS. If the CONTROLINTERVAL option is specified, the RBA= option can be used to access control records in an RRDS. (INFILE only)

RC4STOP
stops the DATA step from executing if when the VSAM data set is opened, a return code greater than 4 is returned by the system.

RECORDS=*variable*
defines a numeric variable that SAS sets to the number of logical records in a VSAM data set that has been opened for input.

RECORG=*record-organization*
specifies the organization of records in a new VSAM data set. Use this option only if SMS is active. Valid values are

KS	VSAM key-sequenced data set

ES	VSAM entry-sequenced data set
RR	VSAM relative-record data set
LS	VSAM linear-space data set

RESET
indicates that the VSAM file is to be reset to empty (no records) when it is opened. This option applies only to loading a VSAM data set that has been marked REUSE. This option cannot be used if the data set contains an alternate index.

RRN=*variable*
defines a numeric variable that you set to the relative record number (RRN) of the record that you want to read or write. This option indicates that keyed direct access is being used; it is appropriate for RRDS only.

SEQUENTIAL
specifies sequential VSAM record retrieval when either the RBA= (for an ESDS) or the RRN= option (for an RRDS) is specified. (INFILE only)

SKIP
indicates skip-sequential processing of VSAM files. Skip-sequential processing finds the first record whose value is the same as the value specified by the KEY= option; records are read sequentially from that point on. (INFILE only)

UPDATE=*variable*
defines a numeric SAS variable that tells SAS that not every record that it reads is to be updated. This option is used when you are updating records in a VSAM data set (INFILE only.) When you have both an INFILE and a FILE statement referencing the same VSAM data set, records are retrieved for update by default.

In most cases where you retrieve a record for update, no user, including you, can access that particular record or any other records in the same control interval until you free the record by executing a PUT or an INPUT statement for the data set. The UPDATE= option is used to avoid user lockout when only a few of many records read need to be updated. When you set the UPDATE= variable to a value of 1 before the INPUT statement, the record is retrieved for update. This is the default if UPDATE= is not specified. If you set UPDATE=0 before the INPUT statement, the record is not retrieved for update.

Options for UNIX System Services Files and Pipes

Several options can be specified in the FILE statement for files and pipes that are in the Hierarchical File System of UNIX System Services. For information about these options, see “Host-Specific Options for UNIX System Services Files” on page 96, “Writing to External Files” on page 79, “Using the FILE Statement to Specify Data Set Attributes” on page 83 .

See Also

□ *SAS Language Reference: Dictionary*
□ *SAS Guide to VSAM Processing*

FILENAME

Associates a SAS fileref with an external file

Valid: anywhere

OS/390 specifics: *fileref, device-type, physical-filename, host-options*

Syntax

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

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

FILENAME *fileref* | _ALL_ CLEAR;

FILENAME *fileref* | _ALL_ LIST;

fileref
identifies an external file. The fileref must conform to the rules for DDnames. That is, it can consist of one to eight letters, numbers, or the national characters $, @, and #. The first character must be either a letter or one of the national characters.

device-type
specifies a device type for the output. It can be one of the following:

CART
send the output to a tape cartridge.

DISK
sends the output to a disk drive.

DUMMY
specifies a null output device. This value is especially useful in testing situations. Any output that would normally be sent to the external file is dicarded.

HFS
specifies a UNIX System Services file.

MVS
specifies an MVS data set.

PIPE
specifies that SAS open a UNIX System Services pipeline for execution of UNIX System Services commands issued within the statement.

PLOTTER
sends the output to the default system plotter.

PRINTER
sends the output to the default system printer.

TAPE
sends the output to a tape drive.

TEMP
allocates a temporary data set.

TERMINAL
sends the output to your terminal.

You can specify *device-type* in two ways: either specify it between the fileref and the file specification in the FILENAME statement or use the standard file option UNIT= to specify it. (See “Standard File Options for the FILENAME Statement” on page 297.) If you specify *device-type* in both ways, the UNIT= option takes priority. If you do not specify a device type or UNIT value for a new file, SAS uses the current value of the SAS system option FILEDEV=.

***file-specification* or (*'file-specification-1'... 'file-specification-n'*)**

specifies an external file or a concatenation of external files. In a concatenation, each data set name must be enclosed in quotes, and the entire group of concatenated file specifications must also be enclosed in parentheses.

'physical-filename'

specifies a physical file, which can be a sequential data set, a member of partitioned data set (PDS), a member of an extended partitioned data set (PDSE), or a file in UNIX System Services. *'physical-filename'* must be enclosed in quotes and must refer to one of the following:

□ a fully qualified data set name. For example: **'myid.raw.datax'**

□ a fully qualified data set name with a member in parentheses. For example:

```
'sas.raw.data(mem1)'
```

□ a partially qualified data set name with a period preceding. For example:

```
'.raw.data'
```

□ a partially qualified data set name with a period preceding and a member name in parentheses. For example:

```
'.raw.data(mem1)'
```

□ for PDS members, a fully or partially qualified data set name with a wildcard name in parentheses. For example:

```
'.raw.data(mem*)'
```

```
'.raw.data(*mem1)'
```

```
'.raw.data(*)'
```

□ a temporary data set name. For example:

```
'&mytemp'
```

□ a UNIX System Services file:

```
'/u/userid/raw'
```

or

```
'HFS:raw'
```

The value of the FILEEXT= system option can affect the way SAS interprets file specifications for PDS and PDSE files. See "FILEEXT=" on page 344 for details.

The value of the FILESYSTEM= system option can also affect the way SAS interprets filenames. See "FILESYSTEM=" on page 351 for details.

See "Specifying Physical Files" on page 13 for more information about partially qualified data set names.

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.

host-options

are host-specific options that may be specified in the FILENAME statement. These options can be categorized into several groups. For details, see the following sections:

□ “Standard File Options for the FILENAME Statement” on page 297

□ “DCB Attribute Options” on page 300

□ “SYSOUT Data Set Options for the FILENAME Statement” on page 304

□ “Subsystem Options for the FILENAME Statement” on page 306

□ “Options That Specify SMS Keywords” on page 303

□ “Host-Specific Options for UNIX System Services Files” on page 96 .

You can specify these options in any order following *’physical-filename ’*. When specifying more than one option, use a blank space to separate each option. Values for options may be specified with or without quotes. However, if a value contains one of the supported national characters ($, #, or @), the quotes are required.

Standard File Options for the FILENAME Statement

Standard file options provide information about a data set’s disposition and physical attributes. The following standard options can be used with all external files under OS/390 except for files that are in the Hierarchical File System of UNIX System Services. (See “Host-Specific Options for UNIX System Services Files” on page 96 .)

DISP=*status* | (*status*,<*normal-termination-disp*>,<*abnormal-termination-disp*>)

specifies the status of the physical file at the beginning and ending of a job, as well as what to do if the job step terminates abnormally. If you are specifying only *status*, you can omit the parentheses.

status

specifies the status of the data set at the beginning of a job. Valid values are

NEW	a new data set is to be created.
OLD	the data set exists and is not to be shared.
SHR	the data set exists and may be shared.
MOD	if the data set exists, new records are to be added to the end; if the data set does not exist, a new data set is to be created. MOD cannot be specified for a partitioned data set.

The default is SHR.

Note: You can also supply any of these values for status as a separate, individual keyword in the FILENAME statement rather than as a subparameter of the DISP= option. △

normal-termination-disp

specifies what to do with the data set if the job step that was using the data set terminates normally. Valid values are

DELETE	deletes the data set at the end of the step.
KEEP	keeps the data set.
CATLG	places the entry in the system catalog or user catalog.

UNCATLG deletes the entry from the system catalog or user catalog.

For a new data set, the default is CATLG. For an existing data set, the default is KEEP.

abnormal-termination-disp

specifies what to do if the job step terminates abnormally. The default is to take the action that is specified or implied by *normal-termination-disp*. Valid values are

DELETE	indicates that the data set is to be deleted at the end of a job step.
KEEP	indicates that the data set is to be kept.
CATLG	indicates that the system should place an entry in the system or user catalog.
UNCATLG	indicates that the system is to delete the entry in the system or user catalog.

Here are some examples of the DISP parameter:

```
DISP=SHR
DISP=(NEW,CATLG)
DISP=(OLD,UNCATLG,DELETE)
```

SPACE=(*unit*,(*primary*,*secondary*,*directory*),
RLSE,*type*,ROUND)

is the amount of disk space to be provided for a data set that is being created.

unit

may be any of the following:

TRK	specifies that the space is to be allocated in tracks.
CYL	specifies that the space is to be allocated in cylinders.
blklen	specifies that the space is to be allocated in blocks whose block length is *blklen* bytes. The system computes how many tracks are allocated.

primary

specifies how many tracks, cylinders, or blocks to allocate.

secondary

specifies how many additional tracks, cylinders, or blocks to allocate if more space is needed. The system does not allocate additional space until it is needed.

directory

specifies how many 256-byte directory blocks are needed for the directory of a partitioned data set.

RLSE

causes unused space that was allocated to an output data set to be released when the data set is closed. Unused space is released only if the data set is opened for output and if the last operation was a write operation.

type

can be any of the following:

CONTIG	means that the allocated space must be contiguous.
MXIG	means that the maximum contiguous space is required.
ALX	means that different areas of contiguous space are needed.

Note: You can also specify MXIG or ALX as a separate, individual keyword on the FILENAME statement rather than as a subparameter of the SPACE= option. △

ROUND
specifies that the allocated space must be equal to an integral number of cylinders when the *unit* specified was a block length. If *unit* was specified as TRK or CYL, the system ignores ROUND.

Here are some examples of the SPACE parameter:

```
SPACE=(CYL,10)
   or SPACE=(CYL,(10,,10),,CONTIG)
SPACE=(1024,(100,50,20),RLSE,MXIG,ROUND)
```

If SPACE is not specified, its values are taken from the SAS system options FILEUNIT=, FILESPPRI=, FILESPSEC=, and FILEDIRBLK=, in the following form:

```
SPACE=(FILEUNIT,(FILESPPRI,
  FILESPSEC,FILEDIRBLK))
```

The default specifications are as follows:

for partitioned data sets:
SPACE=(CYL,(1,1,6))

for sequential data sets:
SPACE=(CYL,(1,1))

See *OS/390 JCL Reference* by IBM for complete information about how to use the SPACE= option.

VOLSER=*value* | VOL=*value* | VOL=(*value-1*, ..., *value-n*)
specifies the disk or tape volume serial number or numbers. Up to 5 volume serial numbers can be specified.

If VOLSER= is not specified, its value is taken from the SAS system option FILEVOL=.

UNIT=*value*
can name one of several different devices. Some valid values follow. Ask your system administrator whether additional valid values have been defined at your site.

- CART
- DISK
- DUMMY
- PLOTTER
- PRINTER
- TAPE
- TERMINAL

The default for UNIT= is DISK.

LABEL=(*subparameter-list*)
enables you to specify the type and contents of the label of either a tape data set or a disk data set, as well as other information such as the retention period or expiration date for the data set. It is identical to the JCL LABEL= parameter. Here is a simple example:

```
label=(3,SL,,,EXPDT=1999/123)
```

This label specification indicates that the data set sequence number is 3, that it uses standard labels, and that it expires on the 123rd day of 1999. See *OS/390*

JCL Reference by IBM for complete information about how to use the LABEL= option, including which subparameters you can specify in *subparameter-list*.

NOMOUNT
: specifies that the mount message is not to be issued for a volume that is not already online. The default action is to issue the mount message.

NOPROMPT
: specifies that if the file that you reference in the FILENAME statement is unavailable, a requestor window is not displayed and an error message is written to the SAS log.

REUSE
: indicates that the file that you reference in the FILENAME statement is to be freed and reallocated if it is currently in use. By default, SAS does not free and reallocate a file that is currently in use.

WAIT=*n*
: controls how many minutes SAS waits if the file that you reference in the FILENAME statement is unavailable. SAS tries to reacquire the reserved data set every 15 seconds. The value *n* specifies a length of time in clock minutes.

DCB Attribute Options

The following section describes DCB options that can be used in the FILENAME statement. For additional information about DCB characteristics, see "Overview of DCB Attributes" on page 302.

DCB Option Descriptions The following DCB options can be used in the FILENAME statement for all types of external files under OS/390, except for files that are stored in the directory structure of UNIX System Services. (For information about options that are available for UNIX System Services files, see "Host-Specific Options for UNIX System Services Files" on page 96 .) These options correspond to the DCB parameters that you would specify in a JCL DD statement.

BLKSIZE=*value*
: specifies the number of bytes in a block of records. A block is a group of records that SAS and the operating environment move as a unit when reading or writing an external file. The term also refers to the space allowed for each group of records. You seldom need to calculate block size when you write an external file because SAS automatically selects the block size.

The values of the FILEBLKSIZE(*device-type*)= system option contain, for each model of disk that is currently available, the block size that your installation considers best for external, nonprint data sets on that type of disk. Some installations may provide different FILEBLKSIZE default values for batch processing than they do for interactive processing. Therefore, to see the values for the FILEBLKSIZE(*device-type*)= option, run the OPTIONS procedure both in a batch job and in a SAS session under TSO.

For print data sets, which by default have variable-length records, SAS uses a default block size of 264, with one record per block.

You can use the OPT value of the FILEBLKSIZE(*device-type*)= option to tell SAS to calculate the optimal block size for nonprint files. (See

"FILEBLKSIZE(*device-type*)=" on page 341.) Or you can calculate the block size yourself:

- □ To calculate a block size for fixed-length records, multiply the LRECL= value by the number of records you want to put into the block.
- □ For variable-length records, multiply the LRECL= value by the number of records per block and add 4 bytes.

In each case, if you are writing the data set to disk, compare the block size to the track size for the disk. A block cannot be longer than one track of the disk device on which it is stored, and the operating environment does not split a block between tracks. Make sure that the block size does not leave a large portion of the track unused. (If you are not sure, consult your computing center staff.) See "Optimizing I/O" on page 151 for information about determining the optimal block size for your data.

The maximum block size for a data set on tape is 32,760.

BUFNO=*value*

specifies how many memory buffers to allocate for reading and writing. If BUFNO= is not specified, the default is BUFNO=5. See "Optimizing I/O" on page 151 for information about determining the optimal BUFNO= value for your data.

DSORG=*organization*

The following types of organization can be specified:

- □ DA (direct access)
- □ PO (PDS, PDSE)
- □ PS (sequential).

The following types of organization, used to refer to physical files that contain location-dependent information, are also valid: DAU, POU, PSU.

You do not need to include the DSORG= value when you create an external file of type PS or PO. This is because the operating environment identifies a partitioned data set by the presence of a directory allocation in the SPACE= parameter. When you use a FILE statement to write data, SAS identifies a PDS or PDSE by the presence of a member name in the FILE statement. If no member name is present, SAS assumes that the data set is sequential.

LRECL=*value*

specifies the logical record length (that is, the number of bytes in a record). SAS defaults to the size that is needed (for either print or nonprint files) when a file is opened.

Logical record length is affected by the record format (see RECFM=). When the record format is fixed (indicated by an F as part of the RECFM= value), all records have the same length, and that length is the value of the LRECL= value.

When the record format is variable (indicated by a V as part of the RECFM= value), records may have different lengths, and each record contains 4 bytes of length information in addition to its other data. Therefore, you must specify an LRECL= value that is 4 bytes longer than the longest record you expect to write. If you do not know the length of the longest record to be put into a variable-format data set, choose a maximum value and add 4 to it to create an LRECL= value.

OPTCD=*value*

specifies the optional services to be performed by the operating environment. For example, specifying W requests a validity check for write operations on direct-access devices. For more information, see the appropriate IBM OS/390 JCL manual for your system.

Valid values are R, J, T, Z, A, Q, F, H, O, C, E, B, U, and W. You can specify more than one code by listing them with no blanks or commas between them (as with RECFM). A maximum of four characters is allowed.

RECFM=*record-format*

specifies the record format of the file. Valid values are

F	fixed length records, unblocked.
V	variable length records, unblocked.
FB	fixed length records, blocked.
VB	variable length records, blocked.
U	undefined length records, unblocked.

The following values can be appended to the RECFM= values:

A
: specifies that the first byte of each record is an ANSI-printer control character.

M
: specifies that the file is a machine control character file. SAS does not interpret machine-code control characters nor does it create them in output files. See *OS/390 JCL Reference* by IBM for more information.

S
: specifies that the file contains spanned records (when appended to V), or that the file contains standard blocks (when appended to F).

The next format stands alone; no other values may be appended.

N
: indicates that the file is in binary format. The file is treated as a byte stream; that is, line boundaries are not recognized. This record format is specific to the SAS System.

Overview of DCB Attributes DCB attributes and options are relevant to INFILE and FILE statements as well as to the FILENAME statement. This section provides some background information about DCB characteristics.

DCB attributes are those data set characteristics that describe the organization and format of the data set records. If you do not specify these attributes, the SAS System uses default values for them. This section discusses how and under what circumstances these attributes are changed or default values are used.

The discussion focuses on the RECFM, LRECL, and BLKSIZE file attributes. For more information, see the appropriate data administration guide for your system.

Values for these attributes are kept in each of the following operating environment control blocks:

Data Set Control Block (DSCB)
: is the description found in the VTOC of the disk device that the physical file resides on. These are the permanent characteristics of the data set. For tape devices, the data set label in the header of SL tapes contains this information.

Job File Control Block (JFCB)
: maps a physical file on a device to a logical name (DDname). Contains information from a JCL DD statement, TSO ALLOCATE command, SAS FILENAME statement, or SAS FILENAME function. These attributes are either temporary (for the duration of the allocation) or new (to be made permanent).

Data Control Block (DCB)
: describes the current state of an open data set. OS/390 and its access methods (BSAM for the SAS System) use the DCB to control how data is read or written. These attributes are temporary for input, but they become permanent for output.

For existing data sets, DCB attributes are almost always used unchanged from the DSCB. These attributes may be overridden by a DD statement or TSO ALLOCATE command or by SAS FILENAME, FILE, or INFILE options. If an option is specified in both places, the FILENAME, FILE, or INFILE option takes precedence.

When a data set is opened, OS/390 merges information from the DSCB (or data set label) and the JFCB to obtain the current DCB characteristics before entering the DCB open exit. SAS then merges its own information (FILENAME/FILE/INFILE statement options, data set device type, requested data set type, requested line size from LS=) and inspects the resulting DCB attributes. If the result is invalid for some reason, SAS terminates the open operation and issues an appropriate message. Attributes may be considered invalid for any of the following reasons:

- For RECFM=V, VB, BLKSIZE is not at least 4 bytes greater than LRECL.
- For RECFM=F, LRECL equals neither 0 nor BLKSIZE.
- For RECFM=FB, BLKSIZE is not a multiple of LRECL.
- BLKSIZE or LRECL is greater than the OS/390 maximum (32,760).
- LRECL is greater than BLKSIZE (except RECFM=VBS).
- RECFM is not consistent with the requested data set type.
- The requested data length cannot be contained in LRECL.

For any unspecified attributes, SAS takes default values that seem to fit existing attributes. This may cause unexpected combinations of attributes to be set, so be wary of specifying an incomplete set of attributes for a data set.

If no permanent attributes are present (as is possible with a new data set), and if none are given by FILENAME/FILE/INFILE options, then SAS uses default values that are based on the device type and data set type. The following table summarizes these default values.

Table 17.1

Attribute	**DISK**	**TAPE**	**PRINT/ SYSOUT**	**TERMINAL**	**DUMMY**
RECFM	FB	FB	VBA	V	FB
LRECL	80	80	260	261	80
BLKSIZE	*	**	264	265	*

* The smaller of the SAS system option FILEBLKSIZE(*device-type*)= value and the output device maximum, rounded down to a multiple of the LRECL.

** The smaller of the SAS system option FILEBLKSIZE(*device-type*)= value and 32,760, rounded down to a multiple of the LRECL.

If a line size (LS=) parameter is given, SAS uses this value to compute the LRECL, and the BLKSIZE is computed accordingly.

If permanent attributes are overridden on input, these values are used only for the duration of the INFILE processing; the permanent attributes of the data set are not changed. However, if attributes are overridden on output, the specified attributes become permanent for the data set, even if no records are physically written.

Options That Specify SMS Keywords

Several options that specify SMS (Storage Management Subsystem) keywords can be specified in the FILENAME or FILE statement when you create an external file. All of these options are ignored for existing data sets; they apply only when you are creating a data set. If you do not specify any of these options when you create an SMS data set, the system defaults are used. The default values are site-dependent; see your system administrator for details. For more information about SMS data sets, see *OS/390 JCL Reference* by IBM.

DATACLAS=*data-class-name*

specifies the data class for an SMS-managed data set. The name can have up to 8 characters. This option applies only to new data sets; it is ignored for existing data sets. The data class is predefined and controls the DCB attributes for a data set.

The implementation of the DATACLAS= option is compatible with the SMS DATACLAS= JCL parameter. For complete information about this parameter, see *OS/390 JCL Reference*. Ask your system administrator for the data-class names that are used at your site.

DSNTYPE=LIBRARY | PDS

specifies the data set name type.

LIBRARY	indicates that the data set is a PDSE.
PDS	indicates that the data set is a PDS.

DSNTPE= is valid only for SMS data sets, and is valid only when DSORG=PO.

LIKE=*data-set-name*

enables you to allocate an external file that has the same attributes as an existing file. See *OS/390 JCL Reference* for more information.

MGMTCLAS=*management-class-name*

specifies a management class for an SMS data set. The name can have up to 8 characters. This option applies only to new data sets; it is ignored for existing data sets. The management class is predefined and controls how your data set is managed, such as how often it is backed up and how it is migrated.

The implementation of the MGMTCLAS= option is compatible with the SMS MGMTCLAS= JCL parameter. For complete information about this parameter, see *OS/390 JCL Reference*. Ask your system administrator for the management class names that are used at your site.

RECORG=*record-organization*

specifies the organization of records in a new VSAM data set. Use this option only if SMS is active. Valid values are

KS	VSAM key-sequenced data set
ES	VSAM entry-sequenced data set
RR	VSAM relative-record data set
LS	VSAM linear-space data set

STORCLAS=*storage-class-name*

specifies a storage class for an SMS data set. The name can have up to 8 characters. This option applies only to new data sets; it is ignored for existing data sets. The storage class is predefined and controls which device your SMS data set is stored on, such as disk or tape.

The implementation of the STORCLAS= option is compatible with the SMS STORCLAS= JCL parameter. For full details on this parameter, refer to *OS/390 JCL Reference*. See your system administrator for storage class names at your site.

SYSOUT Data Set Options for the FILENAME Statement

The following options apply to data sets that are sent to a system output device (usually a printer). The default value is usually the value that was specified by your site as the installation default. See "Writing to Print Data Sets" on page 85 , as well as your IBM JCL reference, for more information about print data sets.

ALIGN
: asks the operator to check the alignment of the printer forms before printing the data set.

BURST
: tells the operator that the printed output is to go to a burster-trimmer-stacker machine, to be burst into separate sheets.

CHAR1=*value*
: specifies a one- to four-character name for character-arrangement table #1 (used in conjunction with the 3800 Printing Subsystem).

CHAR2=*value*
: specifies a one- to four-character name for character-arrangement table #2 (used in conjunction with the 3800 Printing Subsystem).

CHAR3=*value*
: specifies a one- to four-character name for character-arrangement table #3 (used in conjunction with the 3800 Printing Subsystem).

CHAR4=*value*
: specifies a one- to four-character name for character-arrangement table #4 (used in conjunction with the 3800 Printing Subsystem).

CLOSE
: tells the operating environment to deallocate the data set when the DCB is closed.

COPIES=*value*
: specifies how many copies of the SYSOUT data set to print. The default is COPIES=1.

DEST=*dest-name*
: specifies a destination for the SYSOUT data set. If DEST is not defined, its value is taken from the SAS system option FILEDEST=.

FCB=*fcb-image*
: specifies the forms control buffer image that JES should use to control the printing of the SYSOUT data set.

FLASH=*value*
: specifies which forms overlay frame to use when printing the data set on a 3800 Printing Subsystem.

FLASHC=*value*
: specifies the number of copies on which to print the forms overlay frame.

FOLD
: requests that the print chain or print train for the universal character set be loaded in fold mode.

FORMDEF=*member*
: identifies a member that contains statements that tell the Print Services Facility from IBM how to print the SYSOUT data set on a page-mode printer. This option has no effect on SAS forms.

FORMS=*form-num*
: specifies the IBM form number. If FORMS= is not defined, its value is taken from the FILEFORMS= system option. This option has no effect on SAS forms.

HOLD
: tells the system to hold the SYSOUT data set when it is deallocated until it is released by the system operator.

ID=*dest-userid*
: specifies the user ID for the SYSOUT destination.

MODIFY=*value*
: specifies a copy-modification module that tells JES how to print the SYSOUT data set on a 3800 Printing Subsystem.

MODIFYT=*value*
: specifies which of the CHAR*n* tables to use. For example, if *value* equals 1, then the character-arrangement table that is identified by the CHAR1= option is used.

OUTDES=*value*
: specifies the output descriptor.

OUTLIM=*value*
: specifies a limit for the number of logical records in the SYSOUT data set.

PAGEDEF=*member*
: identifies a member that contains statements that tell the Print Services Facility how to format the page on a page-mode printer.

PGM=*pgm-name*
: specifies the SYSOUT program name.

PRMODE=*value*
: specifies which process mode is required for printing the SYSOUT data set.

SYSOUT=*value*
: specifies the output class for the SYSOUT data set. If SYSOUT is not defined, its value is taken from the SAS system option FILESYSOUT=.

UCS=
: specifies the universal character set.

UCSVER
: tells the operator to visually verify that the character set image is for the correct print chain or print train. The character set image is displayed on the printer before the data set is printed.

VERIFY
: tells the operator to verify that the image displayed on the printer is for the correct FCB image.

Subsystem Options for the FILENAME Statement

The following subsystem data set options are also available. For more information about subsystem data sets, see the appropriate IBM OS/390 JCL manual for your site.

SUBSYS=*value*
: specifies the name of the subsystem (from 1 to 4 characters).

PARM1=*value*
: specifies a subsystem parameter (up to 67 characters).

PARM2=*value*
: specifies a subsystem parameter (up to 67 characters).

PARM3=*value*
: specifies a subsystem parameter (up to 67 characters).

PARM4=*value*
specifies a subsystem parameter (up to 67 characters).

PARM5=*value*
specifies a subsystem parameter (up to 67 characters).

Options for UNIX System Services Files and Pipes

Several options can be specified in the FILENAME statement for files and pipes that are in the Hierarchical File System of UNIX System Services. For information about these options, see “Accessing UNIX System Services Files” on page 93 .

See Also

- □ “FILENAME” on page 190
- □ *SAS Language Reference: Dictionary*
- □ *OS/390 JCL Reference* by IBM

FOOTNOTE

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

Valid: anywhere

OS/390 specifics: maximum length of footnote

Syntax

FOOTNOTE<*n*> <’*text* ’ | "*text* ">

Details

Under OS/390, the maximum footnote length is determined by the value of the LINESIZE= system option. The maximum value of LINESIZE= is 256. Footnotes longer than the value of LINESIZE= are truncated.

Note: No space is permitted between FOOTNOTE and the number *n*. △

See Also

- □ *SAS Language Reference: Dictionary*

%INCLUDE

Includes SAS statements and data lines

Valid: anywhere

OS/390 specifics: *file-specification*, JCLEXCL, options

Syntax

%INCLUDE *source-1* <. . . *source-n*>
</<SOURCE2> <S2=*length*> <JCLEXCL>>;

The following list explains some of the components of the %INCLUDE statement. See *SAS Language Reference: Dictionary* for the complete syntax information.

source
describes the location of the information that you want to access with the %INCLUDE statement. The three possible sources follow:

file-specification
Under OS/390, this can be a fileref or a physical file name enclosed in quotes.

internal-lines
You can access lines that were entered earlier in the same SAS job or session. In order to use this technique in a line mode session, the SAS system option SPOOL must be in effect.

keyboard-entry
You can enter the statements or data lines directly from the terminal. Use an asterisk (*) to indicate that the statements are to come from the terminal.

SOURCE2
causes the SAS log to show the source statements that are being included in your SAS program. In other words, this option has the same effect as the SAS system option SOURCE2, except that it applies only to the records that you are currently including. Specifying SOURCE2 in the %INCLUDE statement works even if the NOSOURCE2 system option is in effect.

S2=*length*
specifies the length of the record to be used for input. Possible values are

S	sets S2 equal to the current setting of the SAS system option S=.
0	tells SAS to use the setting of the SAS system option SEQ= to determine whether the line contains a sequence field. If the line does contain a sequence field, SAS determines the line length by excluding the sequence field from the total length.
n	indicates which columns SAS should scan and which columns, if any, contain sequence numbers that should be ignored. *n* specifies the column in which to start scanning (for variable-length records) or stop scanning (for fixed-length records).

If the source lines in an external file that you are including contain sequence numbers, then either delete them before including the SAS program in your SAS session, or specify S2=0.

JCLEXCL
ignores any lines of JCL in the included source.

See Also

- *SAS Language Reference: Dictionary*

INFILE

Specifies an external file to read with an INPUT statement

Valid: in a DATA step

OS/390 specifics: *file-specification, type, host-options*

Syntax

INFILE *file-specification* <DATALINES | CARDS> <*type*> <*options*>;

file-specification
identifies a file:

fileref
specifies the assigned fileref or the allocated DDname of the file. A fileref must conform to the rules for DDnames. That is, it can consist of one to eight letters, numbers, or the national characters $, @, and #. The first character must be either a letter or a national character.

fileref(*member*)
specifies a member of a partitioned data set, where the PDS or PDSE is specified by the assigned fileref or allocated DDname.

The value of the FILEEXT= system option can affect the way SAS interprets PDS and PDSE member names. See "FILEEXT=" on page 344 for details.

'*physical-filename*'
specifies a physical file, which can be a member of a partitioned data set (PDS), an extended partitioned data set (PDSE), or a UNIX System Services file, using the following syntax:

- a fully qualified data set name. For example:

```
'myid.raw.datax'
```

- a fully qualified data set name with a member in parentheses. For example:

```
'sas.raw.data(mem1)'
```

- a partially qualified data set name with a period preceding. For example:

```
'.raw.data'
```

- a partially qualified data set name with a period preceding and a member name in parentheses. For example:

```
'.raw.data(mem1)'
```

- for PDS members, a fully or partially qualified data set name with a wildcard name in parentheses. For example:

```
'.raw.data(mem*)'
```

```
'.raw.data(*mem1)'
```

```
'.raw.data(*)'
```

- □ a UNIX System Services file:

  ```
  '/u/userid/raw'
  ```

  or

  ```
  'HFS:raw'
  ```

- □ a temporary data set name. For example:

  ```
  '&mytemp'
  ```

The value of the FILEEXT= system option can affect the way SAS interprets file specifications for PDS and PDSE files. See “FILEEXT=” on page 344 for details.

See “Specifying Physical Files” on page 13 for more information about partially qualified data set names.

DATALINES | CARDS

specifies that input data immediately follows a DATALINES or CARDS statement in your SAS program.

type

specifies the type of file. When you omit *type*, the default is a standard external file. Nonstandard (host-specific) file types that you can specify for OS/390 are as follows:

DLI

for IMS-DL/I databases. For information about IMS-DL/I options for the FILE statement, see *SAS/ACCESS Interface to IMS-DL/I Software*.

HFS

for files in UNIX System Services, see “Accessing UNIX System Services Files” on page 93 .

PIPE

for files in UNIX System Services, open a pipe to issue UNIX System Services commands from within the statement. See “Piping Data from SAS to an UNIX System Services Command” on page 96 .

IDMS

specifies that the file is a CA-IDMS file. For information about CA-IDMS options for the INFILE statement, see *SAS/ACCESS Interface to CA-IDMS Software: Reference*.

ISAM

specifies that the file is an ISAM file. See “Accessing ISAM Files” on page 92.

VSAM

specifies that the file is a VSAM file. See “Accessing Nonstandard Files” on page 91.

VTOC

specifies that the Volume Table of Contents (VTOC) is to be accessed.

options

are either portable or host-specific. See *SAS Language Reference: Dictionary* for information on portable options.

You can specify portable options in any order, using a blank space as a separator.

The *host-options* that you can specify depend on which type of external file is being accessed. See the following sections for details:

- □ "Standard Options for the INFILE Statement under OS/390" on page 311
- □ "Host Options for Retrieving Information about Data Sets" on page 312
- □ "VSAM Options for the FILE and INFILE Statements under OS/390" on page 292
- □ "VTOC Options for the INFILE Statement under OS/390" on page 312
- □ "Host-Specific Options for UNIX System Services Files" on page 96 .

When specifying more than one option, use a blank space to separate each option.

Standard Options for the INFILE Statement under OS/390

The following standard options can be used with all standard external files under OS/390.

BLKSIZE=*value* | BLK=*value*

specifies the block size of the file. Block size is discussed in more detail in "DCB Attribute Options" on page 300.

CCHHR=*variable*

specifies a character variable to which the physical address (cylinder head record) of a record is returned. This applies to files on CKD disks only.

CLOSE=*keyword*

indicates how a tape volume is to be positioned at the end of the DATA step. Values for *keyword* are

REREAD	positions the tape at the logical beginning of the data set.
LEAVE	positions the tape at the logical end of the data set.
REWIND	rewinds the tape to the physical beginning of the volume.
FREE	dynamically deallocates the tape volume.
DISP	is implied by the control language.

CSRC

specifies that you want to use the CSRCESRV services that are available with OS/390 to decompress data on input. The following example illustrates the use of this option:

```
data;
   infile myfile csrc;
   input;
run;
```

DCB=*fileref*

gives the fileref of an external file that was referenced in an earlier FILE or INFILE statement in the same DATA step. SAS uses that file's RECFM=, LRECL=, and BLKSIZE= information for the current file.

LINESIZE=*width*

works with LRECL to specify the maximum number of characters per line or record in print files, nonprint files, and the SAS log. Under OS/390, the range of acceptable values of LINESIZE= is 64 to 256. The default value of the LINESIZE=

system option under OS/390 is 132. This default applies only to print files (with carriage returns) or to the SAS log. For nonprint files (without carriage returns), the value of LRECL= is used in place of the default value for LINESIZE=.

LRECL=*value*

specifies the logical record length of the file. The value specified depends on the access method and the device type. For more information, see the discussion of LRECL= in “DCB Option Descriptions” on page 300 and in *OS/390 JCL Reference*.

RECFM=*record-format*

specifies the record format of the file. Valid values are

F	fixed length records, unblocked.
V	variable length records, unblocked.
FB	fixed length records, blocked.
VB	variable length records, blocked.
U	undefined length records, unblocked.

The following values can be appended to the RECFM= values:

A	the first byte of each record is an ANSI printer control character.
M	specifies that the file is a machine control character file. SAS does not interpret machine-code control characters nor does it create them in output files. See *OS/390 JCL Reference* by IBM for more information.
S	the file contains spanned records (V), or the file contains standard blocks (F).

The following value stands alone; no other values can be appended:

N	indicates that the file is in binary format. The file is treated as a byte stream; that is, line boundaries are not recognized. This record format is specific to the SAS System.

Host Options for Retrieving Information about Data Sets

For information on options that enable you to retrieve information about a data set from operating environment control blocks, see “Host Options for Retrieving Information about Data Sets” on page 292.

VTOC Options for the INFILE Statement under OS/390

The following options are used only in INFILE statements that involve VTOC (Volume Table of Contents) access:

CCHHR=*variable*

defines a SAS character variable of length 5 whose value is set to the CCHHR of the last VTOC record that was read by SAS. The returned value is in hexadecimal format; it can be printed by using the $HEX10. SAS format.

CVAF

tells SAS to use the Common VTOC Access Facility (CVAF) of the IBM program product Data Facility/Device Support (DF/DS) for indexed VTOCs. If the VTOC is not indexed, or if your installation does not have CVAF, this option is ignored.

Note: When you use CVAF and CCHHR=, values that are returned for Format-5 DSCB records are not valid, because indexed VTOCs do not have Format-5 DSCB records. △

Host Options for UNIX System Services Files and Pipes

Several options can be specified in the INFILE statement for files and pipes that are in the Hierarchical File System of UNIX System Services. For information about these options, see “Host-Specific Options for UNIX System Services Files” on page 96 .

See Also

- □ “Reading from External Files” on page 86
- □ *SAS Language Reference: Dictionary*

LENGTH

Specifies how many bytes SAS uses to store a variable’s value

Valid: in a DATA step
OS/390 specifics: length of numeric variables

Syntax

LENGTH *variables* <$> *length* . . . <DEFAULT=*n* >;

Note: This is a simplified version of the LENGTH statement syntax; see *SAS Language Reference: Dictionary* for the complete syntax and its explanation. △

length
can range from 2 to 8 for numeric variables and from 1 to 32,767 for character variables.

n
changes from 8 to *n* the default number of bytes that SAS uses for storing the values of newly created numeric variables. Under OS/390, *n* can range from 2 to 8.

See Also

- □ “Using the LENGTH Statement to Save Storage Space” on page 145
- □ *SAS Language Reference: Dictionary*

LIBNAME

Assigns a SAS libref and an engine to a SAS data library

Valid: anywhere
OS/390 specifics: *libref, engine, physical-filename, engine/host-options*

Syntax

LIBNAME *libref* <*engine* > <’*physical-filename*’> < *engine/host-options*>;

LIBNAME *libref <engine>* <(*'physical-filename-1'*, ..., *'physical-filename-n'*)>

LIBNAME *libref* | _ALL_ CLEAR;

LIBNAME *libref* | _ALL_ LIST;

Details

The LIBNAME statement assigns a libref to a SAS data library. The LIBNAME function provides similar functionality. See “LIBNAME” on page 199.

Note: The LIBNAME statement is also used to list the attributes of a SAS data library and to clear a libref. For more information, see “Listing Your Current Librefs” on page 46and “Deallocating SAS Data Libraries” on page 45 . △

Assigning Librefs Assigning a libref associates a SAS data library with an engine and with engine/host options. Use the following form of the LIBNAME statement to assign a libref.

LIBNAME *libref* <*engine* > <*'physical-filename'*> <*engine/host-options*>;

You can use this form to assign a libref to either a permanent SAS data library or a temporary one. The following statement allocates a temporary SAS data library, assigns the libref ABC to it, and associates it with the V6 engine:

```
libname abc v6 '&mytemp' disp=new;
```

The physical file name for this temporary SAS data library is a specially formatted name chosen by the operating environment; it includes your user prefix and the temporary name that you specify (MYTEMP in this example).

You can use the following arguments with this form of the LIBNAME statement:

libref

is a one- to eight-character name that conforms to the rules for both SAS names and DDnames. That is, the first character must be a letter; subsequent characters can be either letters or numbers.

Note: Unlike filerefs, librefs cannot include the national characters $, @, and #. △

engine

specifies which engine to use to access the SAS data library. The following engines can be specified in the LIBNAME statement under OS/390:

V8
: specifies the engine for accessing Version 8 SAS files in disk format. BASE is an alias for the V8 engine.

V78TAPE
: specifies the engine for accessing Version 8 SAS files in sequential format, either on tape or on disk. TAPE and V8SEQ are aliases for V8TAPE.

V7
: specifies the engine for accessing Version 7 SAS files in disk format.

V7TAPE
: specifies the engine for accessing Version 7 SAS files in sequential format, either on tape or on disk.

V6
: specifies the engine for accessing Version 6 SAS files in disk format. V609, V608, V607, and are aliases for the V6 engine.

V6TAPE
: specifies the engine for accessing Version 6 SAS files in sequential format, either on tape or on disk. V6SEQ is an alias for V6TAPE.

V5 specifies the engine for read-only access to Version 5 SAS files on disk.

V5TAPE specifies the engine for read-only access to Version 5 SAS files in sequential format, either on tape or on disk.

XPORT specifies the engine for accessing SAS files in transport format.

BMDP specifies the engine for (read-only) access to BMDP files.

DB2 specifies the engine for accessing DB2 database tables.

ORACLE specifies the access engine to ORACLE database tables.

OSIRIS specifies the engine for (read-only) access to OSIRIS files.

REMOTE specifies the engine that SAS/CONNECT and SAS/SHARE use to access remote files.

SPSS | SPSSX specifies the engine for (read-only) access to SPSS files.

If you do not specify an engine, then SAS attempts to assign an engine according to the rules described in “How SAS Assigns an Engine When No Engine Is Specified” on page 40 . See the discussion of engines in *SAS Language Reference: Dictionary* for more information about the library engines.

'physical-filename'

specifies a physical file, which can be a standard OS/390 data set or an HFS directory in UNIX System Services, using the following syntax:

- □ a fully qualified data set name. For example:

```
'myid.raw.datax'
```

- □ an HFS directory in UNIX System Services:

```
'/u/userid/mylib'
```

or

```
'HFS:mylib'
```

The HFS prefix needed when the directory does not contain a “/” to indicate an HFS file. This information can also be provided by default by setting the FILESYSTEM= system option. For information on the FILESYSTEM= system option, see “FILESYSTEM=” on page 351.

See “Specifying Physical Files” on page 13 for more information about partially qualified data set names.

For some engines, the *physical-filename* may begin with a single or double ampersand (& or &&), followed by a one- to eight-character value. If the *physical-filename* begins with &, a temporary physical file is allocated. Such engines are those that can cause a physical file to be created by way of a LIBNAME statement and include the V8, V8TAPE, V7, V7TAPE, V6, and V6TAPE engines. The first character after the & must be alphabetic; others may be alphanumeric or the national characters $, #, or @.

The *physical-filename* argument is optional. If you specify it, it must follow the engine name. If you did not specify an engine, then it must follow the libref.

engine/host-options

are host-specific options that apply to the SAS data library. Each option is identified by a keyword, and most keywords assign a specific value to that option. You may specify one or more of these options using the following forms:

keyword=value | *keyword*

When you specify more than one option, use a blank space to separate each option. The specific options that are available depend on which engine you have specified. See “Engine/Host Options for the V8 Engine” on page 57 for details about specific *engine/host-options*, and about the values that you specify with the various engines.

The complete list of options is presented here:

DATACLAS=*data-class-name*

specifies the data class for an SMS-managed data set. The name can have up to 8 characters. This option applies only to new data sets; it is ignored for existing data sets. The data class is predefined and controls the DCB attributes for a data set.

The implementation of the DATACLAS= option is compatible with the SMS DATACLAS= JCL parameter. For complete information about this parameter, see *OS/390 JCL Reference* by IBM. Ask your system administrator which data-class names are used for SAS data libraries at your site.

DISP= *status* | (< *status* >,< *normal-termination-disp*>,< *abnormal-termination-disp*>)

specifies the status of the data set at the beginning and ending of a job, as well as what to do if the job step terminates abnormally. If you are specifying only *status*, you can omit the parentheses.

status

specifies the status of the physical file at the beginning of a job. Valid values are

NEW	a new data set is to be created.
OLD	the data set exists and is not to be shared.
SHR	the data set exists and may be shared.

The default for *status* is OLD.

normal-termination-disp

specifies disposition for the data set if the job using the data set terminates normally. If you omit the normal termination disposition value, the default is CATLG for new data sets or KEEP for existing data sets. Valid values are

DELETE	the data set is deleted at the end of the step.
KEEP	the data set is to be kept.
CATLG	the system should place an entry in the system catalog or user catalog.
UNCATLG	the system is to delete the entry in the system catalog or user catalog.

abnormal-termination-disp

specifies what to do if the job step terminates abnormally. The default is to take the action that is specified or implied by *normal-termination-disp*. Valid values are

DELETE	the data set is deleted at the end of the step.
KEEP	the data set is to be kept.
CATLG	the system should place an entry in the system catalog or user catalog.
UNCATLG	the system is to delete the entry in the system catalog or user catalog.

DLTRUNCHK | NODLTRUNCHK
: overrides the system option DLTRUNCHK for this LIBNAME statement assignment only.

EXTEND
: specifies that when SAS allocates this library, it will allocate it with a volume count that is one greater than the current number of DASD volumes on which the library resides. With this option, a single-volume library can be converted to a multivolume library, and existing multivolume libraries can be extended to another volume.

HFS
: specifies that the library is in the hierarchical file system of UNIX System Services. It is unnecessary to specify this option if the '*physical-filename*' on the LIBNAME statement contains a "/" or if the 'HFS:*data-set-name*' syntax is used.

HIPERSPACE
: specifies that the SAS data library be placed in a hiperspace rather than on disk. HIP is an alias for the HIPERSPACE option. For more information about this option, see "Optimizing I/O" on page 151

LABEL=(*subparameter-list*)
: enables you to specify for a tape or direct access data set the type and contents of the label of the tape or disk data set, as well as other information such as the retention period or expiration date for the data set.

The LABEL= option is identical to the JCL LABEL= parameter. Here is a simple example:

```
label=(3,SL,,,EXPDT=1999/123)
```

This label specification indicates the data set sequence number is 3, that it uses standard labels, and that it expires on the 123rd day of 1999. See *OS/390 JCL Reference* by IBM for complete information about how to use the LABEL= option, including which subparameters you can specify in *subparameter-list*.

LIKE='*physical-filename*'
: when allocating a new library, tells SAS to set the DCB attributes of the new library to the same values as those in the specified data set.

LINEAR
: specifies that this new library should be allocated as a VSAM linear data set. This will then be a permanent library, which uses the HIPERSPACE access method by way of the DIV (Data In Virtual) facility.

MGMTCLAS=*management-class-name*
: specifies a management class for an SMS data set. The name can have up to 8 characters. This option applies only to new data sets; it is ignored for existing data sets. The management class is predefined and controls how your data set is managed, such as how often it is backed up and how it is migrated.

The implementation of the MGMTCLAS= option is compatible with the SMS MGMTCLAS= JCL parameter. For complete information about this parameter, see *OS/390 JCL Reference* by IBM. Ask your system administrator which management class names are used at your site.

NOPROMPT
: for this assignment, specifies that even if the system option FILEPROMPT is in effect and if the library does not already exist, that no requestor window is displayed to ask if you want to create the library.

SPACE=(*unit*,(*primary*<,*secondary*>), <RLSE>,<*type*>, <ROUND>)

specifies how much disk space to provide for a data set that is being created. The space can be requested in terms of tracks, cylinders, or blocks, as follows:

unit

may be any of the following:

TRK	specifies that the space is to be allocated in tracks.
CYL	specifies that the space is to be allocated in cylinders.
blklen	specifies that the space is to be allocated in blocks whose block length is *blklen* bytes. The system computes how many tracks are allocated.

primary

specifies how many tracks, cylinders, or blocks to allocate.

secondary

specifies how many additional tracks, cylinders, or blocks to allocate if more space is needed. The system does not allocate additional space until it is needed.

RLSE

causes unused space that was allocated to an output data set to be released when the data set is closed. Unused space is released only if the data set is opened for output and if the last operation was a write operation.

type

can be any of the following:

CONTIG	means that the space to be allocated must be contiguous.
MXIG	means that the maximum contiguous space is required.
ALX	means that different areas of contiguous space are needed.

ROUND

specifies that the allocated space must be equal to an integral number of cylinders when the *unit* specified was a block length. If *unit* was specified as TRK or CYL, the system ignores ROUND.

If SPACE is not defined, its values are taken from the SAS system options FILEUNIT=, FILESPPRI=, and FILESPSEC=, in the following form:

```
SPACE=(FILEUNIT,(FILESPPRI,FILESPSEC))
```

The default specifications are as follows:

```
SPACE=(CYL,(1,1))
```

STORCLAS=*storage-class-name*

specifies a storage class for an SMS data set. The name can have up to 8 characters. This option applies only to new data sets; it is ignored for existing data sets. The storage class is predefined and controls which device your SMS data set is stored on, such as disk or tape.

The implementation of the STORCLAS= option is compatible with the SMS STORCLAS= JCL parameter. For full details on this parameter, refer to *OS/390 JCL Reference* by IBM. Ask your system administrator which storage class names are used at your site.

UNIT=*value* | (*value, n*)

where *n* is the number of units for multivolume data libraries; can name one of several different devices. Some likely values are DISK and SYSDA. Additional valid values may be defined at your site.

VOLSER=*value* | (*value-1*, ..., *value-n*)

specifies up to 5 disk volume serial numbers. If VOLSER= is not specified, its value is taken from the SAS system option FILEVOL= (see “FILEVOL=” on page 353 for details).

WAIT=*n*

specifies how long the SAS System waits for a data set that is held by another job or user before the LIBNAME statement fails. The value *n* specifies a length of time in clock minutes. If the data set becomes free before *n* minutes expire, then the LIBNAME statement is processed as usual. The dynamic allocation request is retried internally every 15 seconds.

When you use the WAIT= option, you must also specify the engine name in the LIBNAME statement if you are accessing uncataloged data libraries or libraries that do not reside on disk. Otherwise, you do not have to specify the engine name.

For batch jobs using WAIT=, also specify the FILEMSGS option (see “FILEMSGS” on page 347), which causes a message to be written to the system log for each allocation attempt, thus allowing system operators to determine why the job is waiting.

See Also

- □ LIBNAME function, see “LIBNAME” on page 199
- □ “Allocating SAS Data Libraries Internally” on page 34
- □ “Listing Your Current Librefs” on page 46
- □ “LIBNAME” on page 439
- □ *SAS Language Reference: Dictionary*

OPTIONS

Changes the value of one or more SAS system options

Valid: anywhere

OS/390 specifics: *options*

Syntax

OPTIONS *options-1* <. . . *option-n*>;

Details

Some of the options that you can specify are host-specific. “Summary Table of SAS System Options” on page 419 describes all of the system options that are available in

SAS under OS/390. Descriptions of the portable system options are provided in *SAS Language Reference: Dictionary*.

Some system options can be changed only when you invoke SAS, not in an OPTIONS statement. “Summary Table of SAS System Options” on page 419 tells where each system option can be specified.

See Also

- □ “SAS System Options” on page 10
- □ *SAS Language Reference: Dictionary*

SYSTASK LIST

Lists asynchronous tasks

Valid: anywhere

OS/390 specifics: all

Syntax

SYSTASK LIST <_ALL_ | *taskname*> <STATE>;

ALL
specifies all active tasks in the system. A task is *active* if it is running, or if it has completed and has not been waited for using the WAITFOR statement on the remote host that submitted the task.

STATE
displays the status of the task, which can be Start Failed, Running, or Complete.

taskname
requests information for one remotely submitted task. If the task name contains a blank character, enclose *taskname* in quotes.

Details

Task names can be listed with the SYSTASK LIST statement. These task names are assigned on other hosts and are supplied to the OS/390 SAS session via RSUBMIT commands or statements in SAS/CONNECT software.

See Also

- □ “WAITFOR” on page 322
- □ *SAS/CONNECT User’s Guide*

TITLE

Specifies title lines for SAS output

Valid: anywhere

OS/390 specifics: maximum length of title

Syntax

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

Details

Under OS/390, the maximum title length is determined by the value of the LINESIZE= system option. The maximum value of LINESIZE= is 256. Titles longer than the value of LINESIZE= are truncated.

Note: No space is permitted between TITLE and the number *n*. △

See Also

□ *SAS Language Reference: Dictionary*

TSO

Issues a TSO command or invokes a CLIST or a REXX exec during a SAS session

Valid: anywhere

OS/390 specifics: all

Syntax

TSO <*command* >;

command

can be a system command enclosed in quotes, an expression whose value is a system command, or the name of a character variable whose value is a system command. Under OS/390, "system command" includes TSO commands, CLISTs, and REXX execs.

Details

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

SAS executes the TSO statement immediately. Under OS/390, TSO is an alias for the X statement. On other operating environments, the TSO statement has no effect, whereas the X statement is always processed.

You can use the TSO statement to issue most TSO commands or to execute CLISTs or REXX execs. However, you cannot issue the TSO commands LOGON and LOGOFF, and you cannot execute CLISTs that include the TSO ATTN statement.

Note: You cannot use the TSO statement in a batch job. △

TSOEXEC TSOEXEC is a TSO command that is used to invoke authorized commands. At OS/390 sites that run under later releases of TSO/E, you can invoke authorized commands such as RACF commands by submitting the following statement:

```
tso tsoexec authorized-command ;
```

For more information, see the IBM document *TSO Extensions Command Reference*.

Entering TSO Submode You can also use the TSO statement to go into TSO submode during a SAS session.

To start the submode, place the TSO statement in your program without specifying any options. (In the windowing environment, enter TSO submode by issuing `TSO` as a command-line command. See “TSO” on page 449.) When the statement is executed, SAS goes into TSO submode and prompts you for TSO commands. Any commands that you enter in TSO submode are processed by TSO; they are not processed as SAS statements. They can be any length; however, if the command is longer than one line, you must enter a TSO continuation symbol.

To return to the SAS session, enter `RETURN`, `END`, or `EXIT`. Any characters that follow the RETURN, END, or EXIT subcommand are ignored. An END command that occurs within a CLIST terminates the command procedure without ending the TSO submode.

See Also

- □ Statement: “X” on page 323
- □ Functions: “SYSTEM” on page 202 and “TSO” on page 204
- □ CALL routines: “CALL SYSTEM” on page 180 and “CALL TSO” on page 181
- □ Command: “TSO” on page 449
- □ “SAS Interface to REXX” on page 136 .

WAITFOR

Suspends execution of the current SAS session until the specified tasks finish executing

Valid: anywhere

OS/390 specifics: all

Syntax

WAITFOR <_ANY | _ALL_> *taskname1* <*taskname2* ...*tasknameX* >
<TIMEOUT=*seconds*>;

taskname

specifies the name of the remotely submitted task(s) that you wish to complete execution before resuming execution of SAS. You cannot use wildcards to specify task names. Resumption of the SAS session depends first on the value of the TIMEOUT= option, second on the execution state of the specified task(s).

ANY | _ALL_

suspends execution of the current SAS session until either one or all of the specified remote tasks finishes execution. The default setting is _ANY_, which means that as soon as one of the specified task(s) completes execution, the WAITFOR statement will finish execution. Note again that resumption of execution is primarily dependent on the TIMEOUT= option.

TIMEOUT=*seconds*

specifies the maximum number of seconds that WAITFOR should suspend the current SAS session, regardless of the execution state of any or all specified tasks. The SAS session resumes execution at the end of the TIMEOUT= period even if specified tasks are still executing. If you do not specify the TIMEOUT= option and you do not specify any task names, WAITFOR suspends execution of the SAS session indefinitely. If you specify tasks name(s) and you do not specify a TIMEOUT= value, the SAS session resumes execution when the specified tasks complete execution. Specifying TIMEOUT= without specifying task names suspends SAS execution for the specified number of seconds.

Details

Task names can be listed with the SYSTASK LIST statement. These task names are assigned on other hosts and are supplied to the OS/390 SAS session via RSUBMIT commands or statements in SAS/CONNECT software.

The SYSRC macro variable contains the return code for the WAITFOR statement. If a WAITFOR statement cannot execute successfully, the SYSRC macro variable will contain a non-zero value. For example, the WAITFOR statement may contain syntax errors. If the number of seconds specified with the TIMEOUT option elapses, then the WAITFOR statement finishes executing and the SYSRC macro variable is set to a non-zero value if:

□ you specify a single task that does not finish executing.

□ you specify more than one task and the _ANY_ option (which is the default setting), but none of the tasks finishes executing.

□ you specify more than one task and the _ALL_ option, and any one of the tasks does not finish executing.

See Also

□ “SYSTASK LIST” on page 320

□ *SAS/CONNECT User’s Guide*

X

Issues an operating environment command during a SAS session

Valid: anywhere

OS/390 specifics: issues a TSO command or invokes a CLIST or a REXX exec

Syntax

X <*command*>;

Details

Under OS/390, the X and TSO statements are identical; on other operating environments, the TSO statement has no effect, whereas the X statement is always processed. See “TSO” on page 321 for more information.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for the OS/390 Environment, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS® Companion for the OS/390® Environment, Verison 8

1–58025–523–X

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

1st printing, November 1999

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