CHAPTER 12

Data Set Options

SAS Data Set Options under OpenVMS

SAS data set options control certain aspects of your SAS session, including the attributes of SAS files and data libraries.

Most SAS data set options are completely described in *SAS Language Reference: Dictionary*. Only the data set options that are specific to the OpenVMS operating environment are documented here. However, all the SAS data set options that are available under OpenVMS are listed in “Summary Table of SAS Data Set Options” on page 246.

Specifying Data Set Options

Data set options can be specified in parentheses following a data set name. Each option applies only to the SAS data set whose name it follows. RENAME= and KEEP= are examples of SAS data set options.

Some data set options can also be specified as engine or host options in the LIBNAME statement or function.

When one such option appears in a LIBNAME statement or function, it affects all data sets in that SAS data library. If the same option is specified both in the LIBNAME statement or function and after a data set name, the SAS System uses the value given after the data set name. For more information about the LIBNAME statement and about engine or host options, see “LIBNAME” on page 378.

Some SAS data set options have the same effect (and usually the same name) as system or statement options. For example, the BUFSIZE= data set option is analogous to the BUFSIZE= system option. In the case of overlapping options, the SAS System uses the following rules of precedence:

- data set option values (highest precedence)
- statement option values (precedence over system options)
- system option values (precedence over default values)
- default values (lowest precedence).

For more information about SAS system options under OpenVMS, see Chapter 18, “System Options,” on page 387.

Option Syntax

When you specify a data set option, use the following syntax:

data-set-name (*option-1 = value-1 option-2 = value-2*)

For those options that are valid as engine or host options in the LIBNAME statement or function, use the following syntax:

LIBNAME *libref <engine>* ’*SAS-data-library*’ *option-1=value-1 option-2=value-2 ...*;

Remember that not all data set options are valid as engine or host options. Also keep in mind that it is usually better to specify an engine in the LIBNAME statement than to let the SAS System determine which engine to use. This is important when you specify an engine or host option that is supported by two engines (for example, the ALQ= data set option) but which behaves differently in each context.

Summary Table of SAS Data Set Options

Table 12.1 on page 246 lists both the portable data set options and the data set options that are specific to OpenVMS. It describes each option and tells whether the option can be used for a data set that has been opened for input, output, or update. The See column tells you where to look for more detailed information about an option, for example:

COMP	See the description of the data set option in this section.
LR	See *SAS Language Reference: Dictionary*.
RDB	See *SAS/ACCESS Software for Relational Databases: Reference (Oracle Rdb Chapter)*.

The table also lists the engines with which the option is valid.

Table 12.1 Summary of SAS Data Set Options

Data Set Option	Description	Where Used	Engines	See
ALQ=	specifies the allocation quantity for a data set	output	V6, V7, V8, CONCUR	COMP
ALQMULT=	specifies the multiple of pages allocated for a data set	output	V6, V7, V8, CONCUR	COMP
ALTER=	assigns an alter password to a SAS file and enables access to a password-protected SAS file	input, output, update	V6, V7, V8, CONCUR	LR
BKS=	specifies the bucket size for a data set	output	CONCUR	COMP

Data Set Option	Description	Where Used	Engines	See
BUFNO=	specifies the number of buffers for processing a SAS data set	input, output, update	V6, V6TAPE, V7, V7TAPE, V8, V8TAPE	LR
BUFSIZE=	specifies permanent buffer size for output SAS data sets	output	V6, V6TAPE, V7, V7TAPE, V8, V8TAPE	LR, COMP
CACHENUM=	specifies the number of I/O data caches	input, output, update	V6, V7, V8	COMP
CACHESIZ=	specifies the size of each I/O data cache allocated for a file	input, output, update	V6, V7, V8	COMP
CNTLLEV=	specifies the level of shared access allowed to a SAS data set	input, update	V6, V7, V8, CONCUR	LR, COMP
COMPRESS=	compresses observations in an output SAS data set	output	V6, V7, V8	LR
DEQ=	specifies the default file-extension quantity for a data set	output, update	V6, V7, V8, CONCUR	COMP
DEQMULT=	specifies the number of pages to extend	output, update	V6, V7, V8, CONCUR	COMP
DLDMGACTION=	specifies what type of action to take when a SAS catalog in a SAS data library is detected as damaged	input, output, update		LR
DROP=	excludes variables from processing or from output SAS data sets	input, output, update	all	LR
ENCRYPT=	encrypts SAS data files	input, output, update	V6, V6TAPE, V7, V7TAPE, V8, V8TAPE	LR
FILECLOSE=	specifies the volume positioning to be performed when a SAS file residing on tape is closed	input, output	V6TAPE, V7TAPE, V8TAPE	LR, COMP
FIRSTOBS=	begins processing at a specified observation	input, output, update	all	LR
IDXNAME=	directs the SAS System to use a specific index to satisfy the conditions of a WHERE expression	input, output, update	V7, V7TAPE, V8, V8TAPE	LR

Data Set Option	Description	Where Used	Engines	See
IDXWHERE=	overrides the SAS System decision about whether to use an index to satisfy the conditions of a WHERE expression	input, output, update	V7, V7TAPE, V8, V8TAPE	LR
IN=	creates a variable that indicates whether a data set contributed data to the current observation	input	all	LR
INDEX=	defines indexes when creating a SAS data set	input, output, update	V6, V6TAPE, V7 , V7TAPE, V8, V8TAPE	LR
KEEP=	specifies variables for processing or writing to output SAS data sets	input, output, update	all	LR
LABEL=	specifies a label for a data set	input, output, update	all	LR
LOCKREAD=	specifies whether to read a record if a lock cannot be obtained for the record	input	CONCUR	COMP
LOCKWAIT=	indicates whether the SAS System should wait for a locked record	input	CONCUR	COMP
MBF=	specifies the multibuffer count for a data set	input, output, update	CONCUR	COMP
OBS=	specifies the last observation of a data set to process	input, output, update	all	LR
OUTREP=	specifies an operating environment's requirements vector for an output file	input, output, update	V7, V7TAPE, V8, V8TAPE	LR
PW=	assigns a read, write, or alter password to a SAS file and enables access to a password-protected SAS file	input, output, update	V6, V7, V8, CONCUR	LR
PWREQ=	controls the pop-up menu of a requestor window for a data set password	input, output, update	V6, V7, V8	LR
RDBLOCK=	specifies whether record-level locking or table-level locking is used when updating Oracle Rdb data	update	V7, V8	RDB
RDBCONST=	specifies when to evaluate constraints on Oracle Rdb data	input, update	V7, V8	RDB
READ=	assigns a read password to a SAS file and enables access to a read-protected SAS file	input, output, update	V6, V7, V8, CONCUR	LR
RENAME=	changes the name of a variable	input, update	all	LR
REPLACE=	overrides the REPLACE system option	output	all	LR
REUSE=	specifies reuse of space when observations are added to a compressed data set	output	V6, V7, V8	LR

Data Set Option	Description	Where Used	Engines	See
SORTEDBY=	specifies how the data set is currently sorted	input, output, update	V6, V7, V8	LR
TOBSNO=	specifies the number of observations to be transmitted in each multi-observation exchange with a SAS server	input, output, update	V7, V8	LR
TRANTAB=	specifies a translation table for character conversions	input, output, update	V7, V7TAPE, V8, V8TAPE	LR
TYPE=	specifies the data set type for input, update, and output data	input output update	all	LR
WHERE=	selects observations that meet the specified condition	input	all	LR
WHEREUP=	specifies whether to evaluate added observations and modified observations against a WHERE clause	input, output, update	V7, V7TAPE, V8, V8TAPE	LR
WRITE=	assigns a write password to a SAS file and enables access to a write-protected SAS file	input, output, update	V6, V7, V8, CONCUR	LR

ALQ=

Specifies how many disk blocks to initially allocate to a new SAS data set

Language element: data set option

Default: enough blocks for 10 data set pages

Engines: V6, V7, V8, CONCUR

Valid as: data set option, engine or host option

OpenVMS specifics: All aspects are host-specific

Syntax

ALQ=*allocation-quantity*

allocation-quantity

can range from 0 to 2,147,483,647 blocks. (A block is 512 bytes.) If you specify ALQ=0 with the V6, V7, or V8 engine, OpenVMS uses the minimum number of blocks that are required for a sequential file. If you specify ALQ=0 with the CONCUR engine, OpenVMS uses the minimum number of blocks that are required for a relative file. OpenVMS RMS always rounds the ALQ= value up to the next disk cluster boundary. The default value is enough space for 10 pages.

Details The ALQ= data set option is often used with the BUFSIZE= and DEQ= data set options to control buffer size and disk allocation for more efficient I/O. For example, if an application processes a large SAS data set sequentially, it may be more efficient to use option values like the following:

```
data mylib.a(bufsize=16384 alq=300 deq=96);
   . . . more data lines . . .
run;
```

These statements tell SAS to use a buffer size of 16,384 bytes for data set I/O and to allocate 300 disk blocks to the file initially; when more space is needed, it is allocated in 96-block chunks. This potentially reduces the number of file extensions required for the file. Unused disk blocks are deallocated and freed to the file system at close time. This reduces the number of times a file must be extended, potentially creates a more contiguous file, and reduces subsequent access time.

The order of precedence (from highest to lowest) for specifying initial-allocation values and file-extension values is

1 an ALQ= or DEQ= data set option specified for a specific data set
2 an ALQ= or DEQ= data set option specified in a LIBNAME statement or function
3 system default values.

The ALQ= data set option also corresponds to the **FAB$L_ALQ** field in the OpenVMS RMS structure. For information about the **FAB$L_ALQ** field, see *Guide to OpenVMS File Applications*.

See Also

□ Data set option: “ALQMULT=” on page 250
□ Data set option: “BUFSIZE=” on page 252
□ Data set option: “DEQ=” on page 256
□ Data set option: “DEQMULT=” on page 257
□ “Allocating Data Set Space Appropriately” on page 204
□ “Allocating File Space Appropriately” on page 209

ALQMULT=

Specifies the number of pages that are preallocated to a file

Language element: data set option

Default: 10

Engines V6, V7, V8, CONCUR

Valid as: data set option, engine option, global option

OpenVMS specifics: All aspects are host-specific

Syntax

ALQMULT=*n*

n

specifies the number of pages that are allocated for a file. The default value is 10.

Details The ALQMULT= data set option is related to the ALQ= data set option. With the ALQ= data set option, you specify a block value that is preallocated. With the ALQMULT= data set option, enough blocks are preallocated to accommodate the number of pages that you specify. With the ALQMULT= data set option you don't need to know how big the page size (buffer size) is for a file or to determine how many blocks are needed for the desired number of pages.

See Also

- □ Data set option: "ALQ=" on page 249
- □ Data set option: "BUFSIZE=" on page 252
- □ Data set option: "DEQMULT=" on page 257
- □ System option: "ALQMULT=" on page 396

BKS=

Specifies the bucket size for a new data set

Language element: data set option

Default: 32

Engines: CONCUR

Valid as: data set option, engine or host option

OpenVMS specifics: All aspects are host-specific

Syntax

BKS=*bucket-size*

bucket-size
can range from 0 to 63. If you specify a value of 0, SAS uses the minimum number of blocks that are needed to contain a single observation.

Details The BKS= data set option specifies the number of OpenVMS disk blocks in each bucket of a new data set. A *disk block* is 512 bytes. A *bucket* is a storage structure of a set number of blocks used for building and processing files of relative and indexed organization. A bucket contains one or more records or record cells. Buckets are the unit of contiguous transfer between RMS buffers and the disk.

The BKS= data set option is applied only when the data set is created. When deciding on the bucket size to use, consider whether the file will be accessed randomly (small bucket size), sequentially (large bucket size), or both ways (medium bucket size). For example, if you know that all the files in a SAS data library will be accessed randomly, by using the FSEDIT procedure for example, then use a small bucket size at the creation of each file to optimize the access. The following example uses the BKS= option in the LIBNAME statement so that the bucket size applies to all CONCUR data sets in that library:

```
libname test concur '[randomdir]' bks=30;
```

The BKS= data set option corresponds to the **FAB$B_BKS** field in OpenVMS RMS or to the FILE BUCKET_SIZE attribute when you use File Definition Language (FDL). For additional details, see *Guide to OpenVMS File Applications*.

See Also

□ "Using the CONCUR Engine" on page 154

BUFSIZE=

Specifies the size of the internal I/O buffer used for output data sets

Language element: data set option

Default: none

Engines: V6, V7, V8, V6TAPE, V7TAPE, V8TAPE

Valid as: data set option, SAS system option

OpenVMS specifics: integer range

Syntax

BUFSIZE=*integer*

integer

can range from 0 to the host maximum under OpenVMS. The engine tries to use a value in the range of 8,192 to 32,768 if possible. Specify values for BUFSIZE= in increments of 512.

Details The BUFSIZE= data set option specifies the page size (in bytes) for SAS files. A page of a data set is a logical unit that is used by the engine. The page size is a permanent attribute of the data set; under OpenVMS, it is always a multiple of 512 bytes. The page size is set when the file is created and cannot be changed thereafter. There is no default value; the value that the engine uses depends on the size of the observation.

For efficiency, use a larger BUFSIZE= value because it reduces the number of I/Os required to read or write the file. However, if *integer* is too large, disk space may be wasted. Pages must be written in full, even if they are only partially full of data. This means that if you set BUFSIZE= to a large value, 65,536 for example, and the last page contains only 4,000 bytes of data, more than 61,000 bytes of unused storage are written to disk, consuming approximately 120 disk blocks unnecessarily.

The system-dependent CACHESIZ= data set option is related to the BUFSIZE= data set option and can improve I/O performance without wasting disk space. For information about using the CACHESIZ= and BUFSIZE= options together, see "CACHESIZ=" on page 253.

See Also

- □ BUFSIZE= data set option in *SAS Language Reference: Dictionary*
- □ Data set option: "CACHESIZ=" on page 253
- □ "Setting Larger Buffer Size for Sequential Write and Read Operations" on page 207
- □ System option: "BUFSIZE=" on page 400

CACHENUM=

Specifies the number of I/O data caches used per SAS file

Language element: data set option

Default: 10

Engines: V6, V7, V8

Valid as: data set option, engine option, global option

OpenVMS specifics: All aspects are host-specific

Syntax

CACHENUM=*n*

n

specifies the number of I/O data-cache pages to be used per SAS file. The default value is 10.

Details The CACHENUM= data set option operates in conjunction with the CACHESIZ= data set option. Pages are cached in memory in a buffer that is the size of the CACHESIZ= value. The SAS System maintains up to the CACHENUM= value of these caches for each open file.

See Also

- □ Data set option: "CACHESIZ=" on page 253
- □ System option: "CACHENUM=" on page 401

CACHESIZ=

Controls the size of the I/O data cache that is allocated for a SAS file

Language element: data set option

Default: 65024

Engines: V6, V7, V8

Valid as: data set option, engine or host option

OpenVMS specifics: All aspects are host-specific

Syntax

CACHESIZ=*integer*

integer
can range from 0 to 65,024. By default, SAS uses a value that is an even multiple of the file's page size. If you specify a value of 0, SAS does not use a data cache. The default value is 65024.

Details Pages of SAS files are cached in memory with each cache containing multiple pages. The CACHESIZ= data set option controls the size (in bytes) of the data cache used to buffer the I/O pages. Note that memory is consumed for each data cache, and multiple caches are used for each data set opened. Thus, the *disadvantage* of specifying extremely large CACHESIZ= values is large consumption of memory. The *advantage* of a larger CACHESIZ= value is that it reduces the actual number of disk I/Os required to read from or write to a file. For example, if you are reading a large data set, you can use the following statements:

```
libname test v8 '[mydir]';
data new;
   set test.big(cachesiz=65024);
   . . . more data lines . . .
run;
```

This DATA step reads the TEST.BIG data set in the most efficient manner.

If a data cache is used, then one disk I/O is the size of the CACHESIZ= value. If no data cache is allocated, then one disk I/O is the size of the BUFSIZE= value. The size of the page is controlled with the BUFSIZE= data set option.

Comparisons The CACHESIZ= and BUFSIZE= data set options are similar, but they have important differences. BUFSIZE= specifies the file's page size, which is a permanent attribute of the file. It can be set only when the file is created. CACHESIZ= is the size of the internal memory cache that is used for the life of the current file open. It can change any time the file is opened. Specifying a large BUFSIZE= value and CACHESIZ=0 improves I/O the same way that specifying a large CACHESIZ= value does. However, because only complete pages can be written to the file, if the actual data require less space than the specified BUFSIZE= value, the file uses more disk space than necessary.

For example, if you specify BUFSIZE=65024 and CACHESIZ=0, I/O is performed in increments of the page size. If the data actually require only 32,000 bytes of storage, then more than half the space allocated for the file is unused. If you specify BUFSIZE=32768 and CACHESIZ=65024, I/O is still performed in increments of 65,024 bytes; however, if the data require only 32,000 bytes, little space is wasted.

See Also

- □ Data set option: “BUFSIZE=” on page 252
- □ Data set option: “CACHENUM=” on page 253
- □ “Setting Larger Buffer Size for Sequential Write and Read Operations” on page 207
- □ System option: “CACHESIZ=” on page 402

CNTLLEV=

Specifies the level of shared access allowed to SAS data sets

Language element: data set option

Default: varies

Engines: V6, V7, V8, CONCUR

Valid as: data set option

OpenVMS specifics: syntax

Syntax

CNTLLEV=MEM | REC

MEM
demands exclusive access to the file.

REC
allows sharing of the file with record-level locking.

Details The CNTLLEV= data set option specifies the level at which shared update access to a SAS data set is denied.

With the CONCUR engine, you can request exclusive access to the file, or you can share access with record-level locking.

By default with the concurrency engine, data sets that are opened for input allow shared read access; data sets that are opened for output demand exclusive access; and data sets that are opened for update either allow shared read and shared write access or they retain exclusive access to the file, depending on the method that is used.

For example, you can allow other users both read and write access to a data set that you are opening for input only. By default, only shared read access is allowed. To enable record-level locking for the data set TEST.TWO, use the following statements:

```
libname test concur '[mydir]';
data test.one;
   set test.two(cntllev=rec);
run;
```

As another example, suppose you want to both plot and update your file simultaneously by using the FSEDIT procedure. By default, the PLOT procedure opens a data set for exclusive access, even though it only reads the file. To allow concurrent access, use the following statements:

```
libname test concur '[mydir]';
proc plot data=test.a(cntllev=rec);
```

```
run;
```

Note: The CNTLLEV= data set option is ignored for output files; output files always retain exclusive access. △

See Also

- □ CNTLLEV= data set option in *SAS Language Reference: Dictionary*

DEQ=

Tells OpenVMS how many disk blocks to add when it automatically extends a SAS data set during a write operation

Language element: data set option

Default: enough blocks for 5 data set pages

Engines: V6, V7, V8, CONCUR

Valid as: data set option, engine or host option

OpenVMS specifics: All aspects are host-specific

Syntax

DEQ=*default-file-extension-quantity*

default-file-extension-quantity
can range from 0 to 65,535 blocks. (A block is 512 bytes.) If you specify a value of 0, OpenVMS uses the process's default value. The default value is enough space for 5 pages.

Details You should set the value to at least 10 percent of the normal data set size. A large value results in fewer file extensions during the life of the file; a small value results in many file extensions over the life of the file. Note that record access is slowed when a file has many noncontiguous file extensions.

Example

Suppose you specify the DEQ= data set option as follows:

```
libname test '[mydir]';
data test.a(alq=1000 deq=1000);
   . . . more data lines . . .
run;
```

Specifying DEQ=1000 indicates that if more space is needed for the file, that space is allocated in 1,000-block chunks. Note that OpenVMS always rounds the DEQ= value up to the next disk-cluster boundary.

See Also

- □ Data set option: “ALQ=” on page 249
- □ Data set option: “DEQMULT=” on page 257
- □ “Allocating Data Set Space Appropriately” on page 204
- □ “Allocating File Space Appropriately” on page 209

DEQMULT=

Specifies the number of pages to extend a SAS file

Language element: data set option

Default: 5

Engines: V6, V7, V8, CONCUR

Valid as: data set option, engine option, global option

OpenVMS specifics: All aspects are host-specific

Syntax

DEQMULT=*n*

n

specifies the number of pages to extend a SAS file. The default value is 5.

Details The DEQMULT= data set option controls how many pages worth of disk blocks are added to a file each time it has to be extended. By default, enough space for 5 pages is added.

See Also

- □ Data set option: “ALQ=” on page 249
- □ Data set option: “ALQMULT=” on page 250
- □ Data set option: “DEQ=” on page 256
- □ System option: “DEQMULT=” on page 408

FILECLOSE=

Specifies the default CLOSE disposition for a SAS data set on tape

Language element: data set option

Default: LEAVE

Engines: V6TAPE, V7TAPE, V8TAPE

Valid as: data set option

OpenVMS specifics: valid values for *disposition*; syntax

Syntax

FILECLOSE=*disposition*

disposition

can have one of the following values:

REWIND	rewinds to the beginning of the tape after closing each member.
LEAVE	performs no tape positioning when a member is closed.
FREE	rewinds and dismounts the tape when the current member is closed. Note that under OpenVMS, the device is dismounted, not deallocated or "freed."
REREAD	rewinds to the beginning of the file.

Details The FILECLOSE= data set option overrides the TAPECLOSE= system option.

See Also

- □ FILECLOSE= data set option in *SAS Language Reference: Dictionary*
- □ System option: "TAPECLOSE=" on page 446

LOCKREAD=

Specifies whether to read a record if a lock cannot be obtained for the record

Language element: data set option

Default: NO

Engines: CONCUR

Valid as: data set option

OpenVMS specifics: All aspects are host-specific

Syntax

LOCKREAD=YES | NO

YES

indicates that the record must be locked when read. If another process already has the record locked, the read operation fails, and you receive a message that the record is locked by another process.

NO

indicates that records do not have to be locked when they are read. If another process already has the record locked, the data are read but are not locked by the reading process.

Details The LOCKREAD= data set option controls whether SAS reads a record when the record is locked by another process. This option applies only to data sets that are accessed for input. Examples of accessing a data set for input include using the SET statement and the PRINT and FSBROWSE procedures. This data set option does not apply when you are using, for example, the FSEDIT procedure, which accesses a data set for update.

If you specify LOCKREAD=YES and your job tries to access a locked record, the read operation fails. The severity of the failure depends on your job. If you are using FSBROWSE, and you try to move to a locked observation, a message displays saying the observation is unavailable. However, you can still move to other observations that are not locked. A more severe failure occurs if you are using a SET statement in the DATA step. If the DATA step encounters a locked record, the DATA step fails, and your data set will probably be incomplete.

CAUTION:

> **Use LOCKREAD=NO with care.** When you specify LOCKREAD=NO, the SAS System reads locked records. This can result in obsolete data being read. If you know that the data that you are reading are changing often, do not use LOCKREAD=NO. △

LOCKREAD= is closely related to the LOCKWAIT= data set option, which controls whether SAS waits for a locked record to become available.

See Also

- □ Data set option: “LOCKWAIT=” on page 259

LOCKWAIT=

Indicates whether SAS should wait for a locked record

Language element: data set option

Default: NO

Engines: CONCUR

Valid as: data set option

OpenVMS specifics: All aspects are host-specific

Syntax

LOCKWAIT=YES | NO | *n*

YES

tells OpenVMS RMS to wait when it is requesting access to locked records. If you specify LOCKWAIT=YES, OpenVMS RMS waits forever. Even if LOCKREAD=NO is in effect, the record is not read.

NO

tells OpenVMS RMS not to wait for a locked record to become available. If a record is locked and LOCKREAD=NO is in effect, the SAS System reads the record anyway. This can result in obsolete data being read. If LOCKREAD=YES is in effect, the I/O request fails.

n

tells OpenVMS RMS to wait for *n* seconds for a locked record. If the record becomes available within *n* seconds, the record is read. If the record does not become available and LOCKREAD=YES is in effect, the I/O request fails. If the record does not become available and LOCKREAD=NO is in effect, the record is read even though it is locked. Note, however, that you may read obsolete data in this case. The value of *n* can range from 1 to 255.

Details The LOCKWAIT= data set option indicates whether OpenVMS RMS should wait when requesting access to locked records for read, write, or update access. It is used in conjunction with the LOCKREAD= data set option.

See Also

□ Data set option: “LOCKREAD=” on page 258

MBF=

Specifies the multibuffer count for a data set

Language element: data set option

Default: 2 for files opened for update; 1 for files opened for input or output

Engines: CONCUR

Valid as: data set option, engine or host option

OpenVMS specifics: All aspects are host-specific

Syntax

MBF=*multibuffer-count*

multibuffer-count

is the number of I/O buffers that you want OpenVMS RMS to allocate for a particular file. The value can range from 0 to 127, and it must be an integer. If you specify MBF=0, the process’s default value is used.

Details A multibuffer count of 1 is adequate to process your data set sequentially. However, for random access, it is more efficient to use more than one buffer to store various parts of your file in memory. For example, to tell OpenVMS RMS to allocate five buffers for every file in a data library, use the following statement:

```
libname test concur '[mydir]' mbf=5;
```

The MBF= data set option corresponds to the **FAB$B_MBF** field in OpenVMS RMS or to the CONNECT MULTIBUFFER_COUNT attribute when you are using FDL. For additional details about the **FAB$B_MBF** field, see *Guide to OpenVMS File Applications*.

See Also

- □ "Using the CONCUR Engine" on page 154

WRKCACHE=

Specifies the size of the I/O data cache allocated for a file in the WORK data library

Language element: data set option

Default: 65024

Engines: V7, V8

Valid as: data set option, global option

OpenVMS specifics: valid values for *n*

Syntax

WRKCACHE=*n* | *n*K | *n*M | *n*G | MIN | MAX | hex

n
specifies the size of the data cache in bytes (*n*), kilobytes (*n*K), megabytes (*n*M), and gigabytes (*n*G). The default is 65,024 bytes.

MIN
sets the size of the I/O data cache to 0.

MAX
sets the size of the I/O data cache to 65024. This number is the default.

hex
specifies the size of the I/O data cache as a hexadecimal number. This number must be followed by an X, as in *hexadecimal-value*X.

Details The WRKCACHE= data set option controls the size (in bytes) of each I/O data cache used for all files in the WORK data library. The value of *n* must be a positive integer.

See Also

- □ System option: "WRKCACHE=" on page 451
- □ Data set option: "CACHESIZ=" on page 253

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for the OpenVMS Environment, Version 8,* Cary, NC: SAS Institute Inc., 1999. 518 pp.

SAS® Companion for the OpenVMS Environment, Version 8

1–58025–526–4

1st printing, October 1999

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

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