CHAPTER 18

System Options

System Options in the OS/390 Environment

Portable system options for base SAS are documented in the *SAS Language Reference: Dictionary*. Only the base SAS system options that are specific to OS/390 or that have aspects specific to OS/390 are documented in this chapter. However, Table 18.1 on page 420 lists all the base SAS system options that are available under OS/390.

For information on system options that support a SAS product, such as SAS/ACCESS, SAS/CONNECT, or SAS/SHARE, see the documentation for that product.

For information about using SAS system options under OS/390, see “SAS System Options” on page 10.

For information on file specifications, see “Referring to External Files” on page 78 and “Ways of Allocating External Files” on page 69.

ALTLOG=

Specifies a destination for a copy of the SAS log

Default: none

Valid in: configuration file, SAS invocation

Category: Environmental Control: ENVFILES

OS/390 specifics: *file-specification*

Syntax

ALTLOG=*file-specification*

file-specification

identifies an external file. Under OS/390, it can be a valid DDname, a physical file name, or the name of a file stored in the directory structure of UNIX System Services. The DDname must have been previously associated with an external file using either a TSO ALLOCATE command or a JCL DD statement.

Details

The ALTLOG= system option specifies a destination to which a copy of the SAS log is written. Use the ALTLOG= option to capture the log output for printing.

See Also

- □ “ALTPRINT=” on page 328
- □ “Directing Output to an External File at SAS Invocation” on page 104

ALTPRINT=

Specifies a file for a copy of the SAS procedure output file

Default: none

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: *file-specification*

Syntax

ALTPRINT=*file-specification*

file-specification
identifies an external file. Under OS/390, it can be a valid DDname, a physical file name, or the name of a file stored in the directory structure of UNIX System Services. The DDname must have been previously associated with an external file using either a TSO ALLOCATE command or a JCL DD statement.

Details

Use the ALTPRINT= option to capture procedure output for printing.

See Also

□ "ALTLOG=" on page 328

□ "Directing Output to a Printer" on page 107

AUTOEXEC=

Specifies the autoexec file

Default: SASEXEC

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: *file-specification*

Syntax

AUTOEXEC=*file-specification* | NOAUTOEXEC

file-specification
identifies an external file. Under OS/390, it can be a valid DDname, a physical file name, or the name of a file stored in the directory structure of UNIX System Services. The DDname must have been previously associated with an external file using either a TSO ALLOCATE command or a JCL DD statement.

NOAUTOEXEC
disables AUTOEXEC, as if the *file-specification* was null.

Details

The autoexec file contains SAS statements that are executed automatically when you invoke SAS. The autoexec file can contain any SAS statements. For example, you can include LIBNAME statements for SAS data libraries that you access routinely in SAS sessions.

During initialization, SAS checks to see whether the SASEXEC DDname has been allocated. If so, SAS initializes AUTOEXEC= to SASEXEC, else it sets it to blank.

See Also

□ "Autoexec Files" on page 8

BLKALLOC

Causes SAS to set LRECL and BLKSIZE values for a SAS data library when it is allocated rather than when it is first accessed

Default: NOBLKALLOC

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

BLKALLOC | NOBLKALLOC

Details

If BLKALLOC is in effect, then SAS sets LRECL and BLKSIZE values for a SAS data library even if the data library is not opened during the SAS session in which it is allocated. Both LRECL and BLKSIZE are set to the value specified in the first of the following that has a value:

□ the SAS system option BLKSIZE=

□ the SAS system option BLKSIZE(OTHER)=

□ 6144.

BLKSIZE=

Specifies the default block size for SAS data libraries

Default: 0

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

BLKSIZE=*n* | *n*K | MIN | MAX | *hex*X

***n* | *n*K**
specifies default block size in bytes or kilobytes, respectively.

MIN
sets default block size to 1024.

MAX
sets default block size to 32,760.

***hex*X**
specifies default block size as a hexadecimal number of bytes.

Details

The BLKSIZE= option has an effect when you are creating a SAS data library. After the library is created, the block size is set. This option sets the physical block size of the library.

The BLKSIZE= data set option takes precedence over the BLKSIZE= system option. The BLKSIZE= system option takes precedence over the BLKSIZE(*device-type*)= system option.

BLKSIZE(*device-type*)=

Specifies the default block size for SAS data libraries by *device-type*

Default: varies by device type, see table at end of section

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

BLKSIZE(*device-type*)=*value*

device-type
specifies any valid specific device number, or DASD, DISK, or OTHER. DISK or DASD sets values for the device types 2301, 2303, 2305-1, 2305-2, 2314, 3330, 3330-1, 3340, 3350, 3375, 3380, 3390, and 9345.

OTHER
specifies a value that SAS uses when it is unable to determine the exact device type.

value
specifies the default block size. Valid values are

number
specifies the block size that SAS is to use for the device.

OPT
specifies that SAS is to choose an optimum block size for the device.

MAX or FULL
specifies that SAS is to use the maximum permitted block size for the device.

HALF, THIRD, FOURTH, or FIFTH
specifies that SAS is to use the largest value that results in obtaining two, three, four, and five blocks per track, respectively.

Details

The following example tells SAS to choose optimum block size values for all disk devices except 3380s, for which one-third track blocking is requested:

```
options blksize(disk)=opt
        blksize(3380)=third;
```

BLKSIZE(*device-type*)= accepts values in the range of 1024–56664.
Default values and ranges of values are as follows:

Device	Default	Minimum	Maximum
2301	6144	1024	20483
2303	4608	1024	4892
2305-1	6144	1024	14136
2305-2	6144	1024	14660
2314	6144	1024	7294
3330	6144	1024	13030
3330-1	6144	1024	13030
3340	6144	1024	8368
3350	6144	1024	19069
3375	8192	1024	35616
3380	6144	1024	47476
3390	6144	1024	56664
9345	6144	1024	46456
OTHER	6144	1024	56664

See Also

□ “Optimizing I/O” on page 151

CAPSOUT

Specifies that all output to print files is to be converted to uppercase

Default: NOCAPSOUT

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

CAPSOUT | NOCAPSOUT

CHARTYPE=

Specifies a character set or screen size to use for a device

Default: 0

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVDISPLAY

OS/390 specifics: all

Syntax

CHARTYPE=*cell-size* | *screen-size*

cell-size
specifies the character set number for an IBM 3290 terminal. Values are 1 for a 6 x 12 cell and 2 for a 9 x 16 cell.

screen-size
specifies the screen size for other Extended-Data-Stream (EDS) terminals. Values are 1 for a primary screen size and 2 for an alternate screen size.

Details

For an IBM 3290 terminal, the CHARTYPE= option specifies which character cell size to use. For other EDS terminals, it specifies which screen size to use. This option corresponds to the CHARTYPE option in SAS/GRAPH.

The default value, 0, indicates that the CHARTYPE= option is not applicable to the terminal you are using.

See Also

□ "Improving Screen Resolution on an IBM 3290 Terminal" on page 455

CLIST

Specifies that SAS will obtain its input from a CLIST

Default: NOCLIST

Valid in: configuration file, SAS invocation

Category: Environment Control: EXECMODES

OS/390 specifics: all

Syntax

CLIST | NOCLIST

Details

The CLIST option controls whether SAS obtains its input from the terminal directly (NOCLIST specified) or indirectly (CLIST specified) when running interactively under TSO. When CLIST is specified, you can use TSO CLISTs that include SAS statements after the TSO command that invokes SAS. NODMS must be specified if SAS is to obtain its primary input from a CLIST; otherwise, only input from files that are allocated to the terminal will come from a CLIST.

CONFIG=

Specifies a DDname for the configuration file

Default: CONFIG

Valid in: SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: *DDname*

Syntax

CONFIG=*DDname*

DDname
can be any valid DDname, up to eight characters; the DDname must have been previously associated with an external file using either a TSO ALLOCATE command or a JCL DD statement.

Details

The configuration file can contain any SAS system options except CONFIG=. If this option appears in the configuration file, it is ignored.

See Also

- □ "Configuration Files" on page 7
- □ *SAS Language Reference: Dictionary*

CONSOLELOG=

Specifies where SAS log output is directed if the log file is not available

Default: SASCLOG

Valid in: configuration file, SAS invocation

Category: Log and Procedure Output Control: LOGCONTROL

OS/390 specifics: default value, *file-specification*

Syntax

CONSOLELOG=*file-specification*

file-specification
identifies an external file. Under OS/390, it can be a valid DDname, a physical file, or the name of a file stored in the directory structure of UNIX System Services. The DDname must have been previously associated with an external file using either a TSO ALLOCATE command or a JCL DD statement.

Details

CONSOLELOG= specifies the location of a file that receives SAS log information when the log file of the SAS process is undefined or unavailable. This occurs most frequently during SAS invocation before the normal log file is allocated.

See Also

- □ *SAS Language Reference: Dictionary*
- □ "ALTLOG=" on page 328
- □ "LOG=" on page 375

DBCS

Enables double-byte character support

Default: NODBCS

Valid in: configuration file, SAS invocation

Category: Environment Control: LANGUAGECONTROL

OS/390 specifics: all

Syntax

DBCS | NODBCS

DBCS
uses two bytes for each character in the set.

NODBCS
does not use two bytes for each character in the set.

Details

DBCS indicates that all text, input, output, and data should be processed as if it is encoded in a double-byte character set. Double-byte character sets are used to represent text written in languages other than English.

See Also

- □ "DBCSLANG=" on page 336
- □ "DBCSTYPE=" on page 337

DBCSLANG=

Specifies the language of the double-byte character set

Default: none

Valid in: configuration file, SAS invocation

Category: Environment Control: LANGUAGECONTROL

OS/390 specifics: all

Syntax

DBCSLANG=*language-name*

language-name
specifies one of the following double-byte character sets:

CHINESE
Simplified Chinese language as used in the People's Republic of China.

HANZI
Alias for CHINESE.

JAPANESE
Japanese language.

KATAKANA
Japanese language with Katakana.

KOREAN
Korean language.

TAIWANESE
Taiwanese differs from traditional Chinese, but it uses the same characters.

UNKNOWN
Set automatically when an invalid value is specified.

Details

All values of DBCSLANG= require that you specify DBCSTYPE=IBM.

See Also

- □ "DBCS" on page 335
- □ "DBCSTYPE=" on page 337

DBCSTYPE=

Specifies the encoding sequence for double-byte character sets

Default: IBM
Valid in: configuration file, SAS invocation
Category: Environment Control: LANGUAGECONTROL
OS/390 specifics: *encoding_method*

Syntax

DBCSTYPE=*encoding_method*

encoding_method
specifies one of the following encoding methods for double-byte character sets:

FACOM
Specifies the Fujitsu encoding method (JEF code).

HITAC
Specifies the Hitachi encoding method (KEIS code).

IBM
Specifies the IBM encoding method.

See Also

- □ "DBCS" on page 335
- □ "DBCSLANG=" on page 336

DEVICE=

Specifies a terminal device driver for SAS/GRAPH software

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Graphics: GRAPHICS

OS/390 specifics: *device-driver-name*

Syntax

DEVICE=*device-driver-name*

Details

To see a list of available device drivers, use the GDEVICE procedure. If you are in the windowing environment, submit the following statements:

```
proc gdevice catalog=sashelp.devices;
run;
```

If you are running in interactive line mode, noninteractive mode, or batch mode, use the following statements:

```
proc gdevice catalog=sashelp.devices nofs;
list _all_;
run;
```

See Also

- □ *SAS/GRAPH Software: Reference*
- □ *SAS Language Reference: Dictionary*

DLINITDEFER

Suppresses synchronization of VTOC entry at library creation time

Default: NODLINITDEFER

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

DLINITDEFER | NODLINITDEFER

Details

Creating a new SAS data library normally causes the operating environment to close and reopen the library data set in order to update the DS1LSTAR (last record) value in the VTOC (volume table of contents). Updating the VTOC prevents data loss if the library is not properly closed, as may happen in the event of an OS/390 system crash. Unfortunately, at sites with a system exit or SMS management class that releases unused space (blocks/tracks) when a data set is closed*, a U315 abend may occur when creating a new data library. The associated error message text is of the form:

```
ERROR: Physical I/O error on SAS data library '<data set name>',
       on the volume VVVVVV JOBNAME,STEPNAME,DA,DDNAME,OO-OP,
       OUT OF EXTENT,NNNNNNNNNNNNN,EXCP
```

Note: This error and abend does not occur when creating temporary libraries (except for the WORK SAS library) or when the library allocation request includes the subparameter RLSE (RELEASE) SPACE. In these cases, the new library data set is not initially closed and reopened to update the VTOC. △

Specifying the system option DLINITDEFER prevents the error and abend by preventing the initial closing and reopening of the new library.

Specifying DLINITDEFER does involve a risk of data loss for newly initialized libraries that are not successfully closed due to an OS/390 system crash.

DLTRUNCHK

Enables checking for SAS data library truncation

Default: NODLTRUNCHK

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

DLTRUNCHK | NODLTRUNCHK

Details

When you open a SAS data library, DLTRUNCHK causes SAS to check for truncation. If the library appears to be truncated, you cannot complete the open of the library. If you are running a SAS/SHARE server, it is recommended that you specify this option.

If you specify NODLTRUNCHK, SAS does not check for truncation.

DSRESV

Requests exclusive use of shared disk volumes when accessing partitioned data sets on shared disk volumes

* For example, a management class that specifies `partial release = yes immediate`.

Default: NODSRESV

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

DSRESV | NODSRESV

DSRESV
reserves the device, which prevents other processors from accessing the volume on which the partitioned data set resides.

NODSRESV
enqueues the resources that are defined by the operating environment.

Details

The DSRESV option controls whether certain SAS utility procedures, such as PDSCOPY, issue the RESERVE macro instruction when they access partitioned data sets on shared disk volumes.

DYNALLOC

Controls whether SAS or the host sort utility allocates sort work data sets

Default: NODYNALLOC

Alias: DYN

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

DYNALLOC | NODYNALLOC

DYNALLOC
specifies that the host sort utility supports dynamic allocation of any necessary work files. Therefore, SAS does not attempt to allocate them.

NODYNALLOC
specifies that SAS will allocate sort work files. This may be necessary if the host sort utility does not support allocation. Some sort programs will not reallocate previously allocated work files even if the space requirements are greater.

See Also

- □ "SORT=" on page 394
- □ "SORTDEV=" on page 397
- □ "SORTUNIT=" on page 406
- □ "SORTWKDD=" on page 407
- □ "SORTWKNO=" on page 407

FILEBLKSIZE(*device-type*)=

Specifies the default maximum block size for external files

Default: varies by device type

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEBLKSIZE(*device-type*)=*value*

device-type
specifies any valid specific device number, as well as DASD, DISK, OTHER, SYSOUT, TAPE, and TERM.

DASD or DISK
sets values for the device types 2301, 2303, 2305-1, 2305-2, 2311, 2314, 2321, 3330, 3330-1, 3340, 3350, 3375, 3380, 3390, and 9345.

OTHER
specifies the value that SAS uses when it is unable to determine the exact device type.

SYSOUT
sets values for SYSOUT data sets.

TAPE
sets values for the 2400, 3400, 3480, 3490E, and 3590 device types.

TERM
sets values for data sets directed to the terminal.

value
specifies the default block size. Valid values are

number
specifies the block size that SAS is to use for the device.

OPT
tells SAS to choose an optimum block size for the device.

MAX or FULL
tells SAS to use the maximum permitted block size for the device.

HALF, THIRD, FOURTH, or FIFTH
tells SAS to use the largest value that results in obtaining two, three, four, and five blocks per track, respectively, (if a disk device) or the maximum permitted block size divided by two, three, four, and five, respectively (if not a disk device).

MIN
same as FIFTH above.

Details

The minimum value for FILEBLKSIZE(*device-type*)= is 5 and the maxmimum is 32,760. Default values are as follows:

Device	Default	Device	Default
2301	20483	3375	17600
2303	4892	3380	23476
2305-1	14136	3390	27998
2305-2	14660	3400	32760
2311	3625	3480	32760
2314	7294	3490E	32760
2321	2000	3590	32760
2400	32760	9345	22928
3330	13030	OTHER	6400
3330-1	13030	SYSOUT	264
3340	8368	TERM	264
3350	19069		

FILECC

Specifies whether to treat data in column 1 of a printer file as carriage-control data when reading the file

Default: NOFILECC

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 Specifics: all

Syntax

FILECC | NOFILECC

FILECC
specifies that data in column 1 of a printer file should be treated as carriage-control data.

NOFILECC
indicates that data in column 1 of a printer file should be treated as data.

FILEDEST=

Specifies the default printer destination

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEDEST=*printer-destination*

Details

The FILEDEST= system option specifies the default destination to be used for printer data sets when the DEST= option is omitted. This can occur when the FILENAME statement or FILENAME function does not have a DEST= value or when the form being used does not have a DEST= value.

See Also

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

FILEDEV=

Specifies the device name used for allocating new physical files

Default: SYSDA

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEDEV=*device-name*

Details

FILEDEV= specifies the device name to be used when dynamically allocating a new physical file if *device-type* or UNIT= is not specified in the FILENAME statement or FILENAME function, or if UNIT= is not specified in the LIBNAME statement or LIBNAME function. Device names are site-specific.

FILEDIRBLK=

Specifies the default directory block allocation for new partitioned data sets

Default: 6

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEDIRBLK=*n*

Details

The FILEDIRBLK= system option specifies how many directory blocks to allocate for a new partitioned data set when the SPACE= option is omitted from the FILENAME statement or FILENAME function.

See Also

- □ "FILESPPRI=" on page 349
- □ "FILESPSEC=" on page 350
- □ "FILEUNIT=" on page 352

FILEEXT=

Specifies how to handle file extensions when accessing members of partitioned data sets

Default: IGNORE

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEEXT=VERIFY | IGNORE | INVALID | ASIS

VERIFY
verifies that the part of the name after the period corresponds to the last level of the partitioned data set name.

IGNORE
ignores the part of the name after the period and specifies that only the part before the period is to be used.

INVALID
disallows any member name with an extension.

ASIS
accepts the member name as it is. These member names must conform to the naming conventions of partitioned data sets, as described below.

Details

For compatibility with SAS on other platforms, the FILEEXT= system option enables you to write portable SAS programs that will run on systems that support file extensions and on systems that do not support file extensions.

Portable SAS programs can access external files with file extensions when you run those programs in environments such as PC and UNIX. When you run those programs in OS/390, and when the program accesses members in partitioned data sets, the value of FILEEXT= determines how the file extensions are interpreted.

Member names in partitioned data sets must consist of one to eight alphanumeric characters starting with a letter or with one of the following national characters: $, #, @. A member name extension is an optional part of the member name that follows a period.

Example of FILEEXT=VERIFY

In this example, SAS verifies that the part of the name that follows the period corresponds to the last level of the partitioned data set name. If it does not, an error message is written to the SAS log:

```
options fileext=verify;

   /* allocate a PDS */
filename out2 'myid.fileext.sas' disp=old;
data _null_;

      /* the member name is 'versas'*/
   file out2(versas.sas);
        put 'text';
run;
```

Example of FILEEXT=IGNORE

Using the IGNORE value causes the extension, if present, to be ignored:

```
options fileext=ignore;

   /* allocate a PDS */
filename out2 'myid.fileext.testsrc' disp=old;
 data _null_;
```

```
      /* the member name is 'dotnd' */
   file out2(dotnd.some);
   put 'text';
run;
```

Example of FILEEXT=ASIS

With the ASIS parameter, the member name is accepted as-is:

```
options fileext=asis;

   /* allocate a PDS */
filename out2 'myid.fileext.testsrc' disp=old;
data _null_;

      /* the member name is 'mem.as' */
   file out2(mem.as);
   put 'text';
run;
```

FILEFORMS=

Specifies the default SYSOUT form for a print file

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Log and Procedure Output Control: LISTCONTROL

OS/390 specifics: all

Syntax

FILEFORMS=*operating-environment-form*

Details

The FILEFORMS= system option specifies a default operating environment form using one to four characters. The default form is used when a printer file is dynamically allocated if FORMS= is not specified in the FILENAME statement or FILENAME function.

Comparison

The FILEFORMS= option specifies operating environment forms, whereas the portable FORMS= system option specifies the name of the default form that is used by the SAS FORM subsystem. For information about the FORM subsystem and about the FORMS= system option, see *SAS Language Reference: Dictionary* and “Using the PRINT Command and the FORM Subsystem” on page 108 .

FILEMOUNT

Specifies whether an off-line volume is to be mounted

Default: FILEMOUNT

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEMOUNT | NOFILEMOUNT

Details

This option applies to the allocation of external files. It tells SAS what to do when an attempt is made to allocate a physical file on a volume that is offline.

If FILEMOUNT is in effect, a request is made to mount the volume. If NOFILEMOUNT is in effect, then the volume is not mounted and the allocation fails.

FILEMSGS

Controls whether you receive expanded dynamic allocation error messages when you are assigning a physical file

Default NOFILEMSGS

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEMSGS | NOFILEMSGS

Details

The FILEMSGS option applies to physical files that are referenced in either a FILENAME statement or function or a LIBNAME statement or function.

If FILEMSGS is in effect and you try to assign a data set that is allocated to another user, SAS generates detailed error messages explaining why the allocation failed. Under TSO, the messages are written to the display. The display is cleared and the messages appear. You must press ENTER to return to your session in the windowing environment. In batch mode, the messages are written to the job log.

If NOFILEMSGS is in effect, you will still receive some error messages in your SAS log, but they may not be as detailed.

FILENULL

Specifies whether zero-length records are written to external files

Default: FILENULL

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILENULL | NOFILENULL

FILENULL
allows zero-length records to be written to external files. This is the default value.

NOFILENULL
prevents zero-length records from being written to external files. This type of record is ignored.

Details

If your file transfer program cannot handle zero-length records, you should specify NOFILENULL before you create the file that you want to transfer.

FILEPROMPT

Controls whether you are prompted if you reference a data set that does not exist

Default: FILEPROMPT (interactive); NOFILEPROMPT (batch)

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEPROMPT | NOFILEPROMPT

FILEPROMPT
specifies that you want to be prompted. The prompt allows you to create the data set dynamically or to cancel the request. This is the default value.

NOFILEPROMPT
specifies that you do not want to be prompted. In this case, the data set is not created, and your LIBNAME or FILENAME statement or function fails.

Details

The FILEPROMPT option controls whether you are prompted if the physical file that is referenced in a FILENAME statement or function or a LIBNAME statement or function does not exist. This option has no effect in batch mode.

FILEREUSE

Specifies whether to reuse an existing allocation for a file that is being allocated to a temporary DDname

Default: NOFILEREUSE

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEREUSE | NOFILEREUSE

Details

If FILEREUSE is in effect and there is a request to allocate a file that is already allocated, the existing allocation is used whenever the new allocation would cause a temporary DDname (of the form @SAS*nnnn*) to be generated.

FILESPPRI=

Specifies the default primary space allocation for new physical files

Default: 1

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILESPPRI=*primary-space-size*

Details

The default primary space is allocated in units that are specified by the FILEUNIT= option. Use the FILESPSEC= option to specify secondary space allocation and the FILEDIRBLK= option to specify the number of directory blocks to be allocated.

The value of this option is used if you omit the SPACE= option from the FILENAME statement or function or LIBNAME statement or function when creating a new physical file.

The range of acceptable values for FILESPPRI= is 1-32760.

FILESPSEC=

Specifies the default secondary space allocation for new physical files

Default: 1

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILESPSEC=*secondary-space-size*

Details

The default secondary space is allocated in units that are specified by the FILEUNIT= system option. Use the FILESPPRI= option to specify primary space allocation, and use the FILEDIRBLK= option to specify the number of directory blocks to allocate.

The value of this option is used if you omit the SPACE= option in the FILENAME statement or function or LIBNAME statement or function when creating a new physical file.

The range of acceptable values is 0-32760.

FILESTAT

Specifies whether ISPF statistics will be written

Default: NOFILESTAT

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILESTAT | NOFILESTAT

Details

FILESTAT causes ISPF statistics to be written in the directory entry for a new member of a partitioned data set, or updated for an existing member that already contains ISPF statistics. NOFILESTAT suppresses ISPF statistics.

FILESYSOUT=

Specifies the default SYSOUT CLASS for a printer file

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Log and Procedure Output Control: LISTCONTROL

OS/390 specifics: all

Syntax

FILESYSOUT=*sysout-class*

sysout-class

is a single character (number or letter only). Valid classes are site dependent. At some sites, data center personnel may have set up a default class that cannot be overridden.

Details

The FILESYSOUT= option specifies the default SYSOUT CLASS that will be used when a printer file is allocated dynamically and when the SYSOUT= option is omitted from the FILENAME statement or FILENAME function.

FILESYSTEM=

Specifies the default file system used when the filename is ambiguous

Default: MVS

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILESYSTEM=MVS | HFS

MVS
specifies that the filesystem is native MVS, which includes partitioned data sets (PDS, PDSE).

HFS
specifies the hierarchical file system of UNIX System Services.

Details

The FILESYSTEM= system option specifies the file system that is used when a physical filename could reside in either file system. For example:

```
options filesystem='HFS';

/* resolves to a UNIX System Services */
/* path, such as /homedir/hfs.file    */

filename myhfs 'hfs.file';
```

See Also

□ "How SAS Determines Device Types" on page 79

FILEUNIT=

Specifies the default unit of allocation for new physical files

Default: CYLS

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEUNIT=*unit-type*

unit-type
specifies the unit of allocation. Valid values include BLK, BLKS, CYL, CYLS, TRK, and TRKS, or an integer. The default is CYLS. If an integer is specified, it is the block size that will be used for the allocation.

Details

The FILEUNIT= option specifies the default unit of allocation that will be used for new physical files if the SPACE= option is not specified in either the FILENAME statement or function or the LIBNAME statement or function.

FILEVOL=

Specifies which VOLSER to use for new physical files.

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

FILEVOL=*volser* | (*volser-1* . . . *volser-n*)

volser
specifies one to five volume serial numbers (VOLSERs); the VOLSERs can be separated by blanks or commas. A VOLSER is a six-character name of an OS/390 DASD or tape volume. The name contains one to six alphanumeric or national characters. VOLSERs are site-specific.

Details

The FILEVOL= option specifies the default VOLSER that will be used for allocating new physical files if the VOL= data set option is omitted from the FILENAME statement or function or the LIBNAME statement or function.

Parentheses are required if more than one VOLSER is specified.

FILSZ

Specifies that the host sort utility supports the FILSZ parameter

Default: FILSZ

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

FILSZ | NOFILSZ

FILSZ
specifies that the host sort utility supports the FILSZ parameter. SAS uses the FILSZ= option in the SORT control statement that it generates and passes to the sort program. FILSZ is more efficient than the SIZE parameter.

NOFILSZ
specifies that the host sort utility does not support the FILSZ parameter. SAS uses the SIZE= option in the SORT control statement that it generates and passes to the sort utility program.

Details

If a program product sort utility that supports the FILSZ parameter is installed, specifying the FILSZ option increases the sort efficiency.

See Also

- □ your site's sort utility documentation

FSBCOLOR

Specifies whether you can set background colors in SAS windows on vector graphics devices

Default: NOFSBCOLOR

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVDISPLAY

OS/390 specifics: all

Syntax

FSBCOLOR | NOFSBCOLOR

FSBCOLOR
enables you to set the background color in your SAS windows. For example, if you specify FSBCOLOR when you invoke SAS, you can issue commands such as the following in any SAS window:

```
color back blue
```

This command sets the background color to blue.

Use the FSBCOLOR option only on vector graphics devices. The FSBCOLOR system option is ignored if you specify it on a program symbols device, and you will receive an error message if you try to set the background color of a window.

NOFSBCOLOR
specifies that no background colors are to be used. This is the default value on all devices.

Details

Nongraphics terminals and *program symbols* graphics terminals, such as the IBM 3279, the PC 3270 emulators, and the Tektronix 4205, do not allow you to set the background color of individual windows; the background color is always black. *Vector* graphics terminals such as the IBM 3179G, 3192G, and 3472G allow you to set the background color.

FSBORDER=

Specifies what type of symbols are to be used in borders

Default: BEST

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVDISPLAY

OS/390 specifics: all

Syntax

FSBORDER=BEST | PS | APL | NONE

BEST
tells SAS to choose the border symbols based on the type of terminal you are using.

PS
tells SAS to use programmed symbols for border symbols in the windowing environment.

APL
tells SAS to use APL symbols.

NONE
indicates that no special border symbols are to be used (normal text is used).

Details

The FSBORDER= system option specifies what type of symbols are to be used in window borders and other widgets.

FSDEVICE=

Specifies the terminal device driver

Default: none

Alias: FSD=

Valid in: configuration file, SAS invocation

Catergory: Environment Control: ENVDISPLAY

OS/390 specifics: *device-name*

Syntax

FSDEVICE=*device-name*

Details

The value of the FSDEVICE= system option is needed to run windowing procedures. See "Terminal Support in the OS/390 Environment" on page 452 for a list of all devices that are supported by the SAS terminal-based interactive windowing system under OS/390.

See Also

□ *SAS Language Reference: Dictionary*

FSMODE=

Specifies the full-screen data stream type

Default: IBM

Valid in: configuration file, SAS invocation

Catergory: Environment Control: ENVDISPLAY

OS/390 specifics: all

Syntax

FSMODE=*data-stream-type*

data-stream-type
is the name of an acceptable data stream type. Valid values are

IBM
is the default.

FACOM | FUJITSU
specifies the F6683 data stream, which can be used for F6683 and F6653 terminals.

HITAC | HITACHI
specifies the T560/20 data stream, which can be used for T560/20, H2020, and H2050 terminals.

Details

The FSMODE= system option specifies the type of IBM 3270 data stream for the terminal. An incorrect setting of this option can cause a 3270 data stream program check or a system abend.

FULLSTATS

Specifies that additional statistics are to be written to the SAS log

Default: NOFULLSTATS

Alias: FULLSTIMER

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Log and Procedure Output Control: LOGCONTROL, System Administration: PERFORMANCE

OS/390 specifics: all

Syntax

FULLSTATS | NOFULLSTATS

Details

If the STATS and STIMER system options are in effect, the FULLSTATS system option causes the following statistics to be written to the SAS log after each step:

- CPU time
- Elapsed time
- Vector affinity time
- Vector usage time
- RSM Hiperspace time
- EXCP count.

Additional statistics can be written to the SAS log by specifying the MEMRPT system option.

See Also

- “MEMRPT” on page 377
- “STATS” on page 409
- “STIMER” on page 410
- “Collecting Performance Statistics” on page 150

GHFONT=

Specifies the default graphics hardware font

Default: none

Valid in: configuration file, SAS invocation

Catergory: Environment Control: ENVDISPLAY

OS/390 specifics: all

Syntax

GHFONT=*font-specification*

Examples of values for *font-specification* are

F6X9
specifies characters that are 6 pixels wide and 9 pixels high.

F9X12
specifies characters that are 9 pixels wide and 12 pixels high.

I6X9
specifies an italic font with characters that are 6 pixels wide and 9 pixels high.

See your system administrator for a complete list of fonts that are available to you.

Details

The GHFONT= option specifies the default hardware font in graphics. It applies only to vector graphics devices that support stroke precision in the vector graphics symbol set (for example, IBM terminals such as 3179G, 3192G, and 3472G).

This option is used with SAS software products where you can specify a smaller font and display more information in the tables on the display.

HELPLOC=

specifies the location of the text and index files for the facility that is used to view SAS help

Default: HELPDOC

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: default value

Syntax

HELPLOC = (*location-1*<,*location-2,...,location-n*>)

location
file specification of a specially formatted help data set known as an itemstore. For further information on developing and using your own help information, see “Using User-Defined Help” on page 27.

Details

Specifying a value for the HELPLOC= system option causes SAS to insert that value at the start of a concatenated list of values, the last of which is the default value HELPDOC. This allows you to access your own help information without losing access to SAS help. After two specifications of HELPLOC=, the value of the system option is of the following form:

```
helploc-specification-2, helploc-specification-1, HELPDOC
```

Both of the specifications in the example above could consist of concatenated lists of help itemstore file specifications.

See Also

- □ “ITEMS” on page 244

HSLXTNTS=

Specifies the size of each physical hiperspace that is created for a SAS data library

Default: 1,500

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

HSLXTNTS=*value*

Details

The HSLXTNTS= option specifies the size, in pages, of each physical hiperspace that is created for a SAS data library with the HIPERSPACE option in the LIBNAME statement or LIBNAME function. These physical hiperspaces are analogous to physical data set extents in that when one is filled, another is obtained. They are logically combined internally to form a single logical hiperspace representing a library.

The value that you specify must be in the range 0 to 2,147,483,647. If you specify 0, SAS uses the value 1,800. Check with your system administrator for any site-specific maximum number of pages you may have.

See Also

- □ “Optimizing I/O” on page 151
- □ *Tuning SAS Applications in the MVS Environment*, by Michael Raithel

HSMAXPGS=

Specifies the maximum number of hiperspace pages allowed in a SAS session

Default: 75,000

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

HSMAXPGS=*value*

Details

The HSMAXPGS= option specifies the maximum number of hiperspace pages that can be allocated in a single SAS session for all hiperspaces. The value of the HSMAXPGS= option is equal to the product of the values of the HSLXTNTS= and HSMAXSPC= options.

The value that you specify must be in the range 0 to 2,147,483,647. If you specify 0, SAS allocates 1,920 blocks of VIO to the library. Check with your system administrator for any site-specific maximum number of pages you may have.

If you are responsible for controlling resource use at your site and you are concerned with hiperspace usage, you can use the IBM SMF installation exit, IEFUSI, to limit the hiperspace resources that are available to users.

See Also

□ “Optimizing I/O” on page 151

□ *Tuning SAS Applications in the MVS Environment*, by Michael Raithel

HSMAXSPC=

Specifies the maximum number of hiperspaces allowed in a SAS session

Default: 50

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

HSMAXSPC=*value*

Details

The HSMAXSPC= option specifies the maximum number of physical hiperspaces (each of which has the size specified by the HSLXTNTS= option) that can be allocated in a single SAS session.

The value that you specify must be in the range 0 to 2,147,483,647. If you specify zero, SAS allocates 1,920 blocks of VIO for the library. Check with your system administrator for any site-specific maximum number of hiperspaces you may have.

See Also

- □ "Optimizing I/O" on page 151
- □ *Tuning SAS Applications in the MVS Environment*, by Michael Raithel

HSSAVE

Controls how often the DIV data set pages are updated when a DIV data set backs a hiperspace library

Default: HSSAVE

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

HSSAVE | NOHSSAVE

HSSAVE
specifies that the DIV data set pages are updated every time SAS writes to the hiperspace.

NOHSSAVE
specifies that the DIV data set pages are updated only when the library is closed. A SAS data library is closed when you clear the library specification or when you end your SAS session.

Details

Note: DIV data sets are also referred to as VSAM linear data sets. △

The HSSAVE default provides the best protection from data loss during programming. During execution of tested programs, you may wish to improve performance by specifying NOHSSAVE. The performance improvement results from a decrease in the number of I/O operations to the DIV data set. However, you should not specify NOHSSAVE unless you are willing to risk losing changes. You may lose changes if the library is not closed before a job terminates abnormally.

See Also

- □ "Optimizing I/O" on page 151
- □ *Tuning SAS Applications in the MVS Environment*, by Michael Raithel

HSWORK

Tells SAS to place the WORK data library in a hiperspace

Default: NOHSWORK

Valid in: configuration file, SAS invocation

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

HSWORK | NOHSWORK

Details

The HSWORK value indicates that a hiperspace should be used for the WORK data library. Specifying NOHSWORK indicates that the WORK data library will not be a hiperspace.

NOHSWORK is the default setting for this option, and this default is probably suitable for most of your programming needs. However, there may be times when you want to place the WORK data library in a hiperspace. For example, the performance of programs (with regard to elapsed time) that perform only output operations to the WORK data library may improve significantly when the WORK data library is a hiperspace library. The performance of programs that perform a mixture of input, output, and update operations usually does not show a significant improvement in elapsed time.

Note: The effect on performance of using a hiperspace for WORK data sets is site-dependent. Your system administrator may want to make recommendations based on investigation of this issue for your site. △

See Also

□ "Optimizing I/O" on page 151

□ *Tuning SAS Applications in the MVS Environment*, by Michael Raithel

ISPCAPS

Specifies whether to convert to uppercase all printable ISPF parameters (both variables and literals) that are used in CALL ISPEXEC and CALL ISPLINK

Default: NOISPCAPS

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPCAPS | NOISPCAPS

Details

If ISPCAPS is in effect, then the values of the variables or literals that are used as parameters will be passed to ISPF in uppercase.

If NOISPCAPS is in effect, then the caller must ensure that the parameters are in the proper case. The names of most ISPF parameters must be in uppercase.

The following example shows two ISPLINK calls. The first turns on the ISPCAPS form of the option. As a result, the parameters that are specified in lowercase in the second ISPLINK call will be passed to ISPF in uppercase.

```
DATA _NULL_;
   CALL ISPLINK('SAS','ISPCAPS');
   CALL ISPLINK('display', 'dmiem1');
   RUN;
```

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ "SAS Interface to ISPF" on page 120

ISPCHARF

Specifies whether SAS character variables with explicit informats or formats have their values converted by the informats or formats each time they are used as ISPF variables

Default: NOISPCHARF

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPCHARF | NOISPCHARF

Details

If ISPCHARF is specified, then formats and informats are used for SAS character variables that have been defined to ISPF via the SAS VDEFINE user exit. If NOISPCHARF is in effect, then formats and informats are not used for these SAS character variables.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- □ "SAS Interface to ISPF" on page 120

ISPCSR=

Specifies the name of the ISPF variable that is set by the SAS VDEFINE user exit to contain the name of a variable whose value is found to be invalid

Default: none

Valid in: configuration file, SAS invocation

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPCSR=*variable-name*

Details

The ISPF variables that are specified by both ISPCSR= and ISPMSG= are set by the SAS VDEFINE user exit whenever the exit finds an ISPF variable that has a zero length, or whenever the SAS informat that is associated with the variable finds the value invalid. SAS uses the VDEFINE user exit to define *variable-name* as a character variable length of eight, placing it in the explicit function pool.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- □ "SAS Interface to ISPF" on page 120

ISPEXECV=

Specifies the name of an ISPF variable which, when set, passes its value to an ISPF service

Default: none

Valid in: configuration file, SAS invocation

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPEXECV=*variable-name*

Details

When accessed, the variable contains the return code for the service request for which it was used. SAS uses the VDEFINE user exit to define *variable-name* as a character variable length of two, placing it in the explicit function pool.

For example, if ISPEXECV = SASEXEC, then you could do the following from an ISPF panel:

&SASEXEC = ’DISPLAY PANEL (XXX)’

IF (&SASEXEC ¬= ’00’) ...

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ “SAS Interface to ISPF” on page 120

ISPMISS=

Specifies the value for SAS character variables defined to ISPF via the SAS VDEFINE user exit when the associated ISPF variable that is being changed has a length of zero

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPMISS=*value*

Details

When the ISPF variable has a length of zero, the value of ISPMISS= is the value that will be assigned to SAS character variables defined to ISPF via the SAS VDEFINE user exit that have explicit formats or informats associated with them. The specified value must be one byte in length.

Note: The specified value is substituted only if the SAS system option ISPCHARF was in effect when the variable was identified to ISPF via VDEFINE. (See “ISPCHARF” on page 363.) △

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- □ "SAS Interface to ISPF" on page 120

ISPMSG=

Specifies the name of the ISPF variable that is set by the SAS VDEFINE user exit to contain a message ID for a variable with an invalid value that has also been identified to ISPF via VDEFINE

Default: none

Valid in: configuration file, SAS invocation

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPMSG=*variable-name*

Details

The ISPF variables that are specified by both ISPMSG= and ISPCSR= are set by the VDEFINE user exit whenever the exit finds an ISPF variable that has a zero length, or whenever the SAS informat that is associated with the variable finds the value invalid. The SAS VDEFINE user exit identifies *variable-name* to ISPF as a character variable length of eight, placing it in the explicit function pool.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- □ "SAS Interface to ISPF" on page 120

ISPNOTES

Specifies whether ISPF error messages are to be written to the SAS log

Default: NOISPNOTES

Category: Host Interfaces: ISPF

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

OS/390 specifics: all

Syntax

ISPNOTES | NOISPNOTES

Details

If ISPNOTES is specified, then ISPF error messages are written to the SAS log. If NOISPNOTES is in effect, then ISPF error messages are not written to the SAS log.

The ISPTRACE option overrides the NOISPNOTES option, so all messages are written to the SAS log when ISPTRACE is specified.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ “SAS Interface to ISPF” on page 120

ISPNZTRC

Specifies whether nonzero ISPF service return codes are to be written to the SAS log

Default: NOISPNZTRC

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF, Log and Procedure Output Control: LOGCONTROL

OS/390 specifics: all

Syntax

ISPNZTRC | NOISPNZTRC

Details

If ISPNZTRC is specified, nonzero ISPF service return codes are written to the SAS log. If NOISPNZTRC is in effect, then nonzero ISPF service return codes are not written to the SAS log.

To display all parameter lists and return codes in the SAS log, use the ISPTRACE option instead of ISPNZTRC.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ “SAS Interface to ISPF” on page 120

ISPPT

Specifies whether ISPF parameter value pointers and lengths are to be written to the SAS log

Default: NOISPPT

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF, Log and Procedure Output Control: LOGCONTROL

OS/390 specifics: all

Syntax

ISPPT | NOISPPT

Details

The ISPPT option is used for debugging. If ISPPT is specified, then ISPF parameter value pointers and lengths are displayed. If NOISPPT is in effect, then ISPF parameter value pointers and lengths are not displayed.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ "SAS Interface to ISPF" on page 120

ISPTRACE

Specifies whether the parameter lists that are passed to ISPF and the service return codes are to be written to the SAS log

Default: NOISPTRACE

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF, Log and Procedure Output Control: LOGCONTROL

OS/390 specifics: all

Syntax

ISPTRACE | NOISPTRACE

Details

If ISPTRACE is specified, then all ISPF service calls and return codes are written to the SAS log. Fixed binary parameters are written to the SAS log, converted to decimal display. After a VDEFINE or VDELETE service request, the list of currently defined SAS variables is written to the SAS log.

If NOISPTRACE is in effect, then ISPF service calls and return codes are not written to the SAS log.

Note: The ISPTRACE option can be set based on the value of the ISPF variable named DMITRACE. In the following example, if the DMITRACE value is YES, then

ISPTRACE will be in effect. If the DMITRACE value is NO, then NOISPTRACE will be in effect. △

```
CALL ISPLINK('DMI','*ISPTRACE');
```

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ "SAS Interface to ISPF" on page 120

ISPVDEFA

Specifies whether all current SAS variables are to be identified to ISPF via the SAS VDEFINE user exit

Default: NOISPVDEFA

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPVDEFA | NOISPVDEFA

Details

If ISPVDEFA is specified, then all current SAS variables are identified to ISPF via the SAS VDEFINE user exit. If an explicit VDEFINE service request is issued, then any variables that it specifies will be defined twice.

If NOISPVDEFA is in effect, then only those variables that are passed explicitly to the VDEFINE user exit will be defined.

To display information on ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ "SAS Interface to ISPF" on page 120

ISPVDLT

Specifies whether VDELETE is executed before each SAS variable is identified to ISPF via VDEFINE

Default: NOISPVDLT

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPVDLT | NOISPVDLT

Details

If ISPVDLT is specified, then each SAS variable is deleted from ISPF with the VDELETE user exit before it is identified to ISPF with VDEFINE. This prevents a SAS variable from being identified to ISPF more than once in any SAS DATA step.

If NOISPVDLT is in effect, then SAS variables are not deleted from ISPF before they are defined. This may cause SAS variables to be defined to ISPF more than once in a SAS DATA step.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ "SAS Interface to ISPF" on page 120

ISPVDTRC

Specifies whether to trace every VDEFINE for SAS variables.

Default: NOISPVDTRC

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF, Log and Procedure Output Control: LOGCONTROL

OS/390 specifics: all

Syntax

ISPVDTRC | NOISPVDTRC

Details

Tracing means that, as each SAS variable is identified to ISPF with the SAS VDEFINE user exit, its name, its VDEFINE length, and any nonzero ISPF return codes are written to the SAS log.

If NOISPVDTRC is in effect, then no information is written to the SAS log when a SAS variable is identified to ISPF via VDEFINE. The NOISPVDTRC setting is useful when many variables are defined with one service request because SAS actually issues multiple VDEFINE requests (one for each variable).

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- “SAS Interface to ISPF” on page 120

ISPVIMSG=

Specifies the ISPF message ID that is to be set by the SAS VDEFINE user exit whenever the informat for a variable returns a nonzero return code

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPVIMSG=*message-id*

Details

The message ID is stored in the ISPF variable that is specified by the ISPMSG= option.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- “SAS Interface to ISPF” on page 120

ISPVRMSG=

Specifies the ISPF message ID that is to be set by the SAS VDEFINE user exit whenever a variable has a null value

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPVRMSG=*message-id*

Details

The message ID is stored in the ISPF variable that is specified by the ISPMSG= option.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ "SAS Interface to ISPF" on page 120

ISPVTMSG=

Specifies the ISPF message ID that is to be displayed by the SAS VDEFINE user exit whenever the ISPVTRAP option is in effect

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPVTMSG=*message-id*

Details

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

□ "SAS Interface to ISPF" on page 120

ISPVTNAM=

Causes the information that is displayed by the ISPVTRAP option to be limited to information for the specified variable name only

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPVTNAM=*variable-name*

Details

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- □ "SAS Interface to ISPF" on page 120

ISPVTPNL=

Specifies the name of the ISPF panel that is to be displayed by the SAS VDEFINE user exit whenever the ISPVTRAP option is in effect

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPVTPNL=*panel*

Details

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- □ "SAS Interface to ISPF" on page 120

ISPVTRAP

Specifies whether the SAS VDEFINE user exit is to write information to the SAS log (for debugging purposes) each time it is entered

Default: NOISPVTRAP

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPVTRAP | NOISPVTRAP

Details

If ISPVTRAP is specified, the SAS VDEFINE user exit writes a message to the SAS log each time it is entered. If the parameters for the ISPVTPNL, ISPVTVARS, and ISPVTMSG options are set, it sets the ISPVTVARS variables and displays the ISPVTPNL panel with the ISPVTMSG message on it. If you press the END key on the information display, the option is set to NOISPVTRAP.

If NOISPVTRAP is in effect, the SAS VDEFINE user exit does not write information to the SAS log each time it is entered.

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- □ "SAS Interface to ISPF" on page 120

ISPVTVARS=

Specifies the prefix for the ISPF variables to be set by the SAS VDEFINE user exit whenever the ISPVTRAP option is in effect

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: ISPF

OS/390 specifics: all

Syntax

ISPVTVARS=*prefix*

Details

The numbers 0 through 5 are appended to this prefix to generate the ISPF variable names. These variables contain the following information:

*prefix*0 =	whether the variable is being read or written
*prefix*1 =	the name of the variable that is being updated
*prefix*2 =	the address of the parameter list for the VDEFINE user exit
*prefix*3 =	the address of the variable that is being updated
*prefix*4 =	the length of the variable that is being updated

*prefix*5 =	the value of the variable that is being updated.

For example, if ISPVTVARS=SASVT, then the variables SASVT0 - SASVT5 would be created. Possible values for these variables could be as follows:

SASVT0	READ (or WRITE)
SASVT1	MYVAR
SASVT2	083C1240
SASVT3	00450138
SASVT4	7
SASVT5	MYVALUE

To display the current settings of your ISPF options, use PROC OPTIONS GROUP=ISPF.

See Also

- □ “SAS Interface to ISPF” on page 120

LINESIZE=

Specifies the line size of SAS output

Default: the terminal’s width setting for interactive modes; 132 for noninteractive modes
Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window
Category: Log and Procedure Output Control: LOG_LISTCONTROL
OS/390 specifics: Default value

Syntax

LINESIZE=*char_width*

Details

Under OS/390, the default for interactive mode (windowing environment and interactive line mode) is the terminal’s width setting. For noninteractive mode and batch mode, the default is 132. The minimum value of *char_width* is 64; the maximum is 256.

See Also

- □ *SAS Language Reference: Dictionary*

LOG=

Specifies a file to which the SAS log is written when executing SAS programs outside the windowing environment

Default: SASLOG

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: *file-specification*

Syntax

LOG=*file-specification*

file-specification
identifies an external file. Under OS/390, it can be a valid DDname, a physical file name, or the name of a file stored in the directory structure of UNIX System Services. The DDname must have been previously associated with an external file using either a TSO ALLOCATE command or a JCL DD statement.

See Also

- □ "ALTLOG=" on page 328
- □ "Copying Output to an External File" on page 105
- □ *SAS Language Reference: Dictionary*

MEMLEAVE=

specifies the amount of memory in the user's region that is reserved exclusively for the use of the operating environment

Default: 512K

Valid in: configuration file, SAS invocation

Category: System Administration: MEMORY

OS/390 specifics: all

Syntax

MEMLEAVE=*n* | *n*K | *n*M | MIN | *hex*X

***n* | *n*K | *n*M**
specifies the amount of memory reserved in bytes, kilobytes, megabytes, or gigabytes, respectively.

MIN
specifies the amount of memory reserved as the minimum value, 0 bytes.

***hex*X**
specifies the amount of memory reserved as a hexadecimal number of bytes.

Details

MEMLEAVE= reserves memory in your region that will not be used by SAS. A minimum memory reservation is required so that the operating environment can perform cleanup activities in the event of an abnormal termination of SAS. You may need to reserve additional memory based on the amount of processing taking place in your region outside of SAS.

The MEMLEAVE= system option relates to the MEMSIZE= system option. To minimize out-of-memory conditions, do not specify a value for the MEMSIZE= option. Instead, leave the value of MEMSIZE= at its default value. This prevents the possibility of your MEMSIZE= value exceeding the size of your memory region. If this occurs, SAS uses up all of its memory before it runs its automatic memory recovery programs. By specifying MEMLEAVE= and leaving MEMSIZE= at its default value, you are ensured that you have memory reserved for cleanup and that memory recovery programs will run to help prevent out-of-memory conditions, regardless of the size of your region.

It should be noted that the value of MEMLEAVE= has no bearing on the values of the PROCLEAVE= and SYSLEAVE= system options. MEMLEAVE= reserves memory that is never used by SAS—it is used exclusively by the operating environment. PROCLEAVE= and SYSLEAVE= reserve SAS memory only.

See Also

- □ “MEMSIZE=” on page 378
- □ “PROCLEAVE=” on page 388
- □ “SYSLEAVE=” on page 412
- □ To adjust the size of your memory region, see the JCL documentation for your operating environment.

MEMRPT

Specifies whether memory usage statistics are to be written to the SAS log for each step

Default: MEMRPT

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: System Administration: MEMORY

OS/390 specifics: all

Syntax

MEMRPT | NOMEMRPT

MEMRPT
if the STATS option is in effect, MEMRPT specifies that memory usage statistics are to be written to the SAS log.

NOMEMRPT
specifies that memory usage statistics are not to be written to the SAS log.

Details

The STATS system option specifies that statistics are to be written to the SAS log. If STATS is in effect and MEMRPT is in effect, then the amount of memory used by the step and the total memory used by the SAS session are written to the SAS log for each step.

Additional statistics can be written to the SAS log by specifying the STIMER and FULLSTATS system options.

See Also

- □ "FULLSTATS" on page 356
- □ "STATS" on page 409
- □ "STIMER" on page 410
- □ "Collecting Performance Statistics" on page 150
- □ *SAS Language Reference: Concepts*

MEMSIZE=

Specifies the limit on the total amount of memory that SAS can use

Default: varies, see the following Details section

Valid in: configuration file, SAS invocation

Category: System Administration: MEMORY

OS/390 specifics: all

Syntax

MEMSIZE=*n* | *n* K | *n* M | *n* G | MIN | MAX | *hex*X

***n* | *n* K | *n* M | *n* G**
specifies total memory size in bytes (0 – 2,147,483,647), kilobytes (0 – 2,097,151), megabytes (0 – 2047), or gigabytes (0–2), respectively.

MIN
causes SAS to calculate the value of MEMSIZE= using the formula in the following Details section.

MAX
causes SAS to calculate the value of MEMSIZE= using the formula in the following Details section.

***hex*X**
specifies total memory size as a hexadecimal number of bytes.

Details

SAS calcuates the value of the MEMSIZE= system option using the formula below if any of the following are true:

- □ No value is specified for MEMSIZE=.
- □ If a value of 0 is specified for MEMSIZE=.
- □ If MIN or MAX is specified for MEMSIZE=.
- □ IF the value specified for MEMSIZE= exceeds the size of the user's REGION.

```
value_of_MEMSIZE_system_option = size_of_user_memory_region --
    value_of_MEMLEAVE_system_option
```

Using the default value for MEMSIZE= removes the possibility of MEMSIZE= values that exceed your memory region. When this is the case, SAS can run out of memory before running memory recovery programs. SAS begins running memory recovery programs when the size of SAS memory approaches the value of MEMSIZE=. If MEMSIZE= is greater than your entire memory region, the memory recovery programs may not run and SAS can run out of memory. For this reason, it is recommended that you not specify a value for MEMSIZE=. Instead, control memory usage by setting your region size and by using an appropriate value for the MEMLEAVE= option. For information on setting your region size, see the JCL information for your operating environment.

See Also

- □ "MEMLEAVE=" on page 376
- □ "Specify a value for MEMSIZE= when you invoke SAS" on page 156

MINSTG

Tells SAS whether to minimize its use of storage

Default: NOMINSTG

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: System Administration: MEMORY

OS/390 specifics: all

Syntax

MINSTG | NOMINSTG

MINSTG
tells SAS to minimize storage in use.

NOMINSTG
tells SAS not to minimize storage in use.

Details

The MINSTG system option tells SAS to minimize its use of storage by returning unused storage and deleting unused load modules at the termination of steps and pop-up windows. This option should be used on memory-constrained systems or when sharing the address space with other applications, such as ISPF split-screen or multisession products.

MSG=

Specifies the file specification of the partitioned data set that contains SAS messages

Default: SASMSG

Alias: SASMSG=

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: all

Syntax

MSG=*file-specification*

file-specification
identifies an external file. Under OS/390, it can be a valid DDname or a physical file name. The DDname must have been previously associated with an external file using either a TSO ALLOCATE command or a JCL DD statement.

Details

Under OS/390, the MSG= system option specifies the file that contains error, warning, and informational messages that are issued during a SAS session.

See Also

- □ "MSGCASE" on page 380
- □ "MSGLOAD" on page 381
- □ "MSGSIZE=" on page 381

MSGCASE

Specifies uppercase or lowercase message display

Default: NOMSGCASE

Valid in: configuration file, SAS invocation

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

MSGCASE | NOMSGCASE

Details

MSGCASE specifies that text taken from the message file is translated to uppercase for display.

See Also

- □ "MSG=" on page 380
- □ "MSGLOAD" on page 381
- □ "MSGSIZE=" on page 381

MSGLOAD

Enables closing and reloading of message files

Default: NOMSGLOAD

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Enironment Control: ENVFILES

OS/390 specifics: all

Syntax

MSGLOAD | NOMSGLOAD

Details

Specifying MSGLOAD closes and reloads message files.

See Also

- □ "MSG=" on page 380
- □ "MSGCASE" on page 380
- □ "MSGSIZE=" on page 381

MSGSIZE=

Specifies the size of the message storage cache

Default: 131,072

Valid in: configuration file, SAS invocation

Category: System Administration: MEMORY

OS/390 specifics: all

Syntax

MSGSIZE=*n* | *n*K | *n*M | *n*G | MIN | MAX | *hex*X

***n* | *n*K | *n*M | *n*G**
specifies message cache size in bytes, kilobytes, megabytes, or gigabytes, respectively.

MIN
sets message cache size to 0 and requires SAS to use the default value.

MAX
sets message cache size to 2,147,483,647.

***hex*X**
specifies message cache size as a hexadecimal number of bytes.

Details

The MSGSIZE= option is set during the installation process and normally is not changed after installation.

See Also

- □ “MSG=” on page 380
- □ “MSGCASE” on page 380
- □ “MSGLOAD” on page 381

MSYMTABMAX=

Specifies the maximum amount of memory available to the macro variable symbol table(s)

Default: 1,048,576 bytes

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Macro: MACRO

OS/390 specifics: default value

Syntax

MSYMTABMAX=*n* | *n*K | *n*M | *n*G | MIN | MAX | *hex*X

***n* | *n*K | *n*M | *n*G**
specifies symbol table size in bytes, kilobytes, megabytes, or gigabytes, respectively.

MIN
sets symbol table size to 0 and requires SAS to use the default value.

MAX
sets symbol table size to 2,147,483,647.

***hex*X**
specifies symbol table size as a hexadecimal number of bytes.

Details

Under OS/390, the default value for this option is 1,048,576 bytes.

See Also

□ *SAS Language Reference: Dictionary*

MVARSIZE=

Specifies the maximum size for macro variables that are stored in memory

Default: 8,192 bytes

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Macro: MACRO

OS/390 specifics: default value, range of available values

Syntax

MVARSIZE=*n* | *n*K | *n*M | *n*G | MIN | MAX | *hex*X

***n* | *n*K | *n*M | *n*G**
specifies maximum macro variable size in bytes, kilobytes, megabytes, or gigabytes, respectively.

MIN
sets maximum macro variable size to 0 and requires SAS to use the default value.

MAX
sets maximum macro variable size to 2,147,483,647.

***hex*X**
specifies maximum macro variable size as a hexadecimal number of bytes.

See Also

□ *SAS Language Reference: Dictionary*

NDSVOLS=

Specifies the VOLSER on which no SAS data library processing is to occur

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

NDSVOLS=*volser*

volser

specifies any valid VOLSER (volume serial number). A VOLSER is a six-character name of an OS/390 DASD volume. The name contains one to six alphanumeric or national characters. Volume serial numbers are site-specific.

Details

This option is used only for Version 5 SAS data libraries. With this option you can specify a volume serial that causes SAS to treat a data set reference to that volume as if you had specified _NULL_. This option is useful for production SAS jobs that, for example, need to initialize catalog generation data groups.

OPLIST

Writes to the SAS log the settings of all SAS system options that you specified when you invoked SAS

Default: NOOPLIST

Valid in: configuration file, SAS invocation

Category: Log and Procedure Output Control: LOGCONTROL

OS/390 specifics: information logged

Syntax

OPLIST | NOOPLIST

Details

Under OS/390, the OPLIST system option writes to the SAS log the settings of all options that were specified on the command line, up to 100 characters. It does not list the settings of system options that were specified in the configuration file.

See Also

- □ “VERBOSE” on page 416
- □ *SAS Language Reference: Dictionary*

PAGESIZE=

Specifies the page size of SAS output

Default: terminal screen size for the windowing environment; 21 for interactive line mode; 60 for noninteractive modes

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Log and Procedure Output Control: LOG_LISTCONTROL

OS/390 specifics: default value, range of available values

Syntax

PAGESIZE=*n* | *n*K | MAX | MIN | *hex*X

***n* | *n*K**
specifies the number of lines that compose a page in bytes, kilobytes, megabytes, or gigabytes, respectively.

MIN
sets the maximum number of lines that compose a page to 15, the minimum available under OS/390.

MAX
sets the maximum number of lines that compose a page to 32,767.

***hex*X**
specifies the maximum number of lines that compose a page as a hexadecimal number.

Details

Under OS/390, the windowing environment uses the terminal screen size to determine page size.

See Also

- □ *SAS Language Reference: Dictionary*

PFKEY=

Specifies which set of function keys to designate as primary

Default: PRIMARY

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVDISPLAY

OS/390 specifics: all

Syntax

PFKEY=*pfkey-set*

pfkey-set
specifies which set of 12 function keys is to be considered the primary set. Acceptable values include the following:

PRIMARY
specifies that the primary set be F13 through F24. Thus, F13 through F24 would have Version 5 settings; F1 through F12 would have Version 8 settings. You can use PRI as an alias for PRIMARY.

ALTERNATE
specifies that the primary set be F1 through F12. Thus F1 through F12 would have Version 5 settings; F13 through F24 would have Version 8 settings. You can use ALT as an alias for ALTERNATE

12
specifies that F1 through F12 exactly match F13 through F24. Thus, both F1 through F12 and F13 through F24 would have Version 5 settings. As a result, the KEYS window displays only F1 through F12.

Details

The PFKEY= option enables you to specify which set of 12 programmed function keys is to be considered primary.

The following values are displayed in the KEYS window when you specify PFKEY=PRIMARY. F1 through F12 are Version 8 settings; F13 through F24 are Version 5 settings.

Key	Definition	Key	Definition
F1	mark	F13	help
F2	smark	F14	zoom
F3	unmark	F15	zoom off; submit
F4	cut	F16	pgm; recall
F5	paste	F17	rfind
F6	store	F18	rchange
F7	prevwind	F19	backward
F8	next	F20	forward
F9	pmenu	F21	output
F10	command	F22	left

Key	Definition	Key	Definition
F11	keys	F23	right
F12	undo	F24	home

See Also

- □ *SAS Language Reference: Dictionary*

PGMPARM=

Specifies the parameter that is passed to the external program specified by the SYSINP= option

Default: none

Valid in: configuration file, SAS invocation

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

PGMPARM=’*string*’

string
can be up to 255 characters long. The quotation marks are optional unless the string contains blanks or special characters.

Details

The PGMPARM= option specifies the parameter that is passed to the external program specified by the SYSINP= option. For more information about using the PGMPARM= and SYSINP= options, contact your lcoal SAS Support Consultant.

PRINT=

Specifies the SAS output file when executing SAS programs outside the windowing environment

Default: SASLIST

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: *file-specification*

Syntax

PRINT=*file-specification*

file-specification

identifies an external file. Under OS/390, it can be a valid DDname, a physical file name, or the name of a file stored in the directory structure of UNIX System Services. The DDname must have been previously associated with an external file using either a TSO ALLOCATE command or a JCL DD statement.

See Also

□ "ALTPRINT=" on page 328

□ "Directing Output to a Printer" on page 107

□ *SAS Language Reference: Dictionary*

PRINTINIT

Initializes the SAS print file

Default: NOPRINTINIT

Valid in: configuration file, SAS invocation

Category: Log and Procedure Output Control: LISTCONTROL

OS/390 specifics: system response to PRINTINIT

Syntax

PRINTINIT | NOPRINTINIT

Details

Under OS/390, specifying PRINTINIT causes the SAS print file to become empty before SAS writes output to it. It also forces the file attributes to be correct for a print file. Specify NOPRINTINIT if a previous program or job step has already written output to the same file and you want to preserve that output.

See Also

□ *SAS Language Reference: Dictionary*

PROCLEAVE=

Specifies how much memory to leave unallocated for SAS procedures to use to complete critical functions during out-of-memory conditions

Default: (0,153600)

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: System Administration: MEMORY

OS/390 specifics: all

Syntax

PROCLEAVE=*n* | *n*K | *n*M | (*n* | *n*K | *n*M, *n* | *n*K | *n*M)

***n* |*n*K | *n*M**
: specifies in bytes, kilobytes, or megabytes how much memory to leave unallocated above the 16-megabyte line. Valid values are any integer from 0 to the maximum amount of available memory.

(*n* | *n*K | *n*M, *n* | *n*K | *n*M)
: specifies how much memory to reserve below the 16-megabyte line, followed by the amount of memory to reserve above the line.

Details

The PROCLEAVE= system option specifies an amount of memory to leave unallocated so that a procedure can terminate normally when error recovery code is initiated. If a procedure that demands large amounts of memory is failing, increase the number of bytes specified by PROCLEAVE=. This causes the failing procedure to use an algorithm that demands less memory. However, the procedure is also forced to use utility data sets, thereby increasing the execution time of the procedure.

See Also

□ "Use SYSLEAVE= and PROCLEAVE= to handle out-of-memory conditions" on page 157

REXXLOC=

Specifies the DDname of the REXX library to be searched when the REXXMAC option is in effect

Default: SASREXX

Valid in: configuration file, SAS invocation

Category: Host Interfaces: REXX

OS/390 specifics: all

Syntax

REXXLOC=*DDname*

Details

The REXXLOC= option specifies the DDname of the REXX library to be searched for any SAS REXX EXEC files, if the REXXMAC option is in effect.

See Also

- □ “SAS Interface to REXX” on page 136
- □ “REXXMAC” on page 390

REXXMAC

Enables or disables the REXX interface

Default: NOREXXMAC

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Host Interfaces: REXX

OS/390 specifics: all

Syntax

REXXMAC | NOREXXMAC

REXXMAC
enables the REXX interface. This means that when SAS encounters an unrecognized statement, it searches for a REXX EXEC file whose name matches the first word of the unrecognized statement. The REXXLOC= system option specifies the DDname of the REXX library to be searched.

NOREXXMAC
disables the REXX interface. This means that when SAS encounters an unrecognized statement, a "statement is not valid" error occurs.

See Also

- □ “SAS Interface to REXX” on page 136
- □ “REXXLOC=” on page 389

S=

For data lines that follow a CARDS statement and for SAS source statements, specifies which columns SAS should scan and which columns, if any, contain sequence numbers that should be ignored

Default: 0

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Input Control: INPUTCONTROL

OS/390 specifics: maximum length of value

Syntax

S=*n*

Details

Under OS/390, *n* can range from 0 to 32,760, which is the maximum length of records on OS/390.

Note: If *n* is 0, SAS uses the value of the SEQ= system option to determine whether the input contains sequence fields that should be ignored. Otherwise, SAS interprets *n* as the column in which to start scanning (for variable-length records) or stop scanning (for fixed-length records). △

See Also

□ *SAS Language Reference: Dictionary*

SASAUTOS=

Specifies the location of the autocall library

Default: SASAUTOS

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Macro: MACRO

OS/390 specifics: *file-specification*

Syntax

SASAUTOS=*file-specification* | (*file-specification-1* . . . *file-specification-n*)

file-specification

identifies the name of an external autocall library. Under OS/390, it can be any valid SAS fileref or a physical file name of a PDS or PDSE.

You can specify one or more autocall libraries at the same time. They will be searched in the order in which they are listed.

Details

SAS looks for autocall members in autocall libraries specified by SASAUTOS=. By default, SAS looks in the library associated with the SASAUTOS fileref. Once you specify the SASAUTOS= system option, that specification replaces the default. SAS no longer searches SASAUTOS unless you include it in the new specification for SASAUTOS=. To add SASAUTOS to your current list of autocall libraries, issue a statement like the following:

```
options sasautos=('your.autocall.lib'
                  'dept.autocall.lib' sasautos);
```

SAS searches the other autocall libraries before it searches the locations associated with the SASAUTOS fileref.

See Also

- □ *SAS Language Reference: Dictionary*
- □ "Specifying a User Autocall Library" on page 218

SASHELP=

Specifies the location of the SASHELP SAS data library

Default: SASHELP

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: *library-specification*

Syntax

SASHELP=*library-specification*

library-specification
can be any valid DDname or the name of the physical file that comprises a SAS data library; the DDname must have been previously associated with the SASHELP SAS data library, using either a TSO ALLOCATE command or a JCL DD statement.

Details

If the SASHELP= option is not specified, then the value SASHELP is used.

See Also

- □ *SAS Language Reference: Dictionary*

SASLIB=

Specifies the DDname for an alternate load library

Default: SASLIB

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: all

Syntax

SASLIB=*DDname*

DDname
is the DDname of a single load library or a concatenation of load libraries that SAS is to search before it searches the standard libraries. The *DDname* must be allocated before SAS is invoked.

Details

The SASLIB= option can be used to specify a load library that contains Version 5 formats, informats, and functions.

SASUSER=

Specifies the location of the SAS data library that contains the user profile catalog

Default: SASUSER

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: *library-specification*

Syntax

SASUSER=*library-specification*

library-specification
can be any valid DDname, the name of the physical file that comprises a SAS data library, or a UNIX System Services directory; the DDname must have been previously associated with the SASUSER SAS data library using either a TSO ALLOCATE command or a JCL DD statement.

Details

If a UNIX System Services directory is being used, it must have been created beforehand.

See Also

- □ “SASUSER Library” on page 9
- □ *SAS Language Reference: Dictionary*

SEQENGINE=

Specifies the default engine for sequential SAS data libraries

Default: SASV7SEQ

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

SEQENGINE=*sequential-engine*

sequential-engine
can have the following values:

TAPE or V8TAPE
specifies the engine for accessing sequential SAS data libraries in Version 8 tape format.

V7TAPE, V7SEQ, or SASV7SEQ
specifies the engine for accessing sequential SAS data libraries in Version 7 tape format.

V6SEQ, or SASV6SEQ
specifies the engine for accessing sequential SAS data libraries in Version 6 tape format.

Details

The SEQENGINE= system option specifies the engine that SAS will use to access an existing sequential format data library when an engine name is not explicitly stated in the LIBNAME statement or LIBNAME function.

See Also

- "Using the V8TAPE Engine" on page 59
- Appendix 1, *Accessing V5 and V5SEQ SAS Data Libraries*, in *SAS Companion for the MVS Environment*

SORT=

Specifies the minimum size of all allocated sort work data sets

Default: 0

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORT=*n* | *n*K | MAX | MIN | *hex*X

***n* | *n*K**
specifies the value of SORT= in bytes or kilobytes, respectively.

MIN
sets SORT= to 0.

MAX
sets SORT= to 32,767.

***hex*X**
specifies SORT= as a hexadecimal number.

Details

The SORT= option specifies the minimum size of all sort work files that SAS allocates. The units are specified by the SORTUNIT= option. If the DYNALLOC system option is specified, then any value that you specify for the SORT= option is ignored.

See Also

- □ “SORTUNIT=” on page 406
- □ “DYNALLOC” on page 340

SORTALTMSGF

Enables sorting with alternate message flags

Default: NOSORTALTMSGF

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTALTMSGF | NOSORTALTMSGF

Details

Specify SORTALTMSGF if the sort utility on your host requires non-standard flags for the message parameter. For example, the Fujitsu system sort utility requires alternate message flags.

SORTBLKMODE

Enables block mode sorting

Default: NOSORTBLKMODE

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTBLKMODE | NOSORTBLKMODE

Details

Specify SORTBLKMODE if the sort utility on your host supports block mode sorting.

SORTBUFMOD

Enables modification of the sort utility output buffer

Default: SORTBUFMOD

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTBUFMOD | NOSORTBUFMOD

Details

Specify NOSORTBUFMOD if the sort utility on your host does not support modification of its sort buffer.

SORTCUTP=

Specifies the number of bytes above which the host sort utility is used instead of the SAS sort program

Default: 4M

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTCUTP=*n* | *n*K | *n*M | *n*G | MAX | MIN | *hex*X

***n* | *n*K | *n*M | *n*G**
specifies the value of SORTCUTP= in bytes, kilobytes, megabytes, or gigabytes, respectively.

MIN
sets SORTCUTP= to 0.

MAX
sets SORTCUTP= to 2,147,483,647 bytes

***hex*X**
specifies SORTCUTP= as a hexadecimal number of bytes.

Details

The SORTCUTP= option specifies the number of bytes (or kilobytes, megabytes, or gigabytes) above which the external host sort utility is used instead of the SAS sort program, if SORTPGM=BEST is in effect.

The following equation computes the number of bytes to be sorted:

number-of-bytes=((*length-of-obs*)+(*length-of-all-keys*))**number-of-obs*

See Also

- □ "Efficient Sorting" on page 154
- □ "SORTPGM=" on page 403

SORTDEV=

Specifies the device name used for allocated sort work data sets

Default: SYSDA

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTDEV=*unit-device-name*

Details

The SORTDEV= option specifies the unit device name if SAS dynamically allocates the sort work file. (See "DYNALLOC" on page 340.) Use a generic device type unit name, such as 3390, rather than a group name, such as SYSDA. To determine the memory requirements, SAS must look up the device characteristics for the specified unit name. A group name might represent multiple device types, making it impossible to predict on which device type the sort work files will be allocated and, therefore, what the memory requirements will be.

For group names, the device characteristics of the WORK library are used. This may result in a warning message, unless NOSORTDEVWARN is in effect.

See Also

"SORTDEVWARN" on page 398

SORTDEVWARN

Enables device type warnings

Default: SORTDEVWARN

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTDEVWARN | NOSORTDEVWARN

Details

Specify NOSORTDEVWARN to disable warning messages sent when SORTDEV= specifies a generic or esoteric device type.

SORTEQOP

Specifies whether the host sort utility supports the EQUALS option

Default: SORTEQOP

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTEQOP | NOSORTEQOP

Details

The SORTEQOP option specifies whether the host sort utility accepts the EQUALS option. (The EQUALS option sorts observations that have duplicate keys in the original order.) If the utility does accept the EQUALS option, then SORTEQOP causes the EQUALS option to be passed to it unless you specify NOEQUALS in the PROC SORT statement. If NOSORTEQOP is in effect, then the EQUALS option is not passed to the host sort utility unless you explicitly specify the EQUALS option in the PROC SORT statement.

Note that equals processing is the default for PROC SORT. Therefore, if NOSORTEQOP is in effect, and if you did not explicitly specify EQUALS, then the host sort interface must do additional processing to ensure that observations with identical keys will remain in the original order. This may adversely affect performance.

SORTLIB=

Specifies the name of the sort library

Default: SYS1.SORTLIB

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTLIB=*physical-filename*

physical-filename
specifies the name of a partitioned data set.

Details

The SORTLIB= option specifies the name of the partitioned data set (load library) that contains the host sort utility (other than the main module specified by the SORTPGM= or SORTNAME= option). This library is dynamically allocated to the DDname SORTLIB. If the host sort utility resides in a link list library or if the sort library is part of the JOBLIB, STEPLIB, or TASKLIB libraries, then this option is unnecessary and should not be specified.

SORTLIST

Enables passing of the LIST parameter to the host sort utility

Default: NOSORTLIST

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTLIST | NOSORTLIST

SORTLIST
tells SAS to automatically pass the LIST parameter to the host sort utility when the SORT procedure is invoked. The host sort utility uses the LIST parameter to determine whether or not to list control statements.

NOSORTLIST
tells SAS not to pass the LIST parameter to the host sort utility.

Details

The SORTLIST option controls whether the LIST parameter is passed to the host sort utility.

Note: If the default for your sort utility is to print messages, then NOSORTLIST has no effect. △

SORTMSG

Controls the class of messages to be written by the host sort utility

Default: NOSORTMSG

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTMSG | NOSORTMSG

SORTMSG
tells SAS to pass the MSG=AP parameter to the host sort utility.

NOSORTMSG
tells SAS to pass the MSG=CP parameter to the host sort utility, which means that only critical messages are written.

SORTMSG=

Specifies the DDname to be dynamically allocated for the message print file of the host sort utility

Default: SYSOUT

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTMSG=*DDname*

DDname
can be any valid DDname or a null string. The DDname will be dynamically allocated to either a SYSOUT data set (with class *) under batch or a terminal under TSO, and the DDname passed to the host sort utility.

Details

The SORTMSG= option specifies a DDname that will be dynamically allocated to either a SYSOUT data set (with class *) or a terminal.

SORTNAME=

Specifies the name of the host sort utility

Default: SORT

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTNAME=*host-sort-utility-name*

host-sort-utility-name
is any valid operating environment name. A valid operating environment name can be up to eight characters, the first of which must be a letter or national character ($, #, or @). The remaining characters, following the first, can be any of the above, or digits.

Details

The SORTNAME= option specifies the name of the host sort utility to be invoked if SORTPGM=HOST or if SORTPGM=BEST and the host sort utility is chosen instead of the SAS sort utility. See “SORTPGM=” on page 403 for information on sort utility selection.

SORTOPTS

Specifies whether the host sort utility supports the OPTIONS statement

Default: SORTOPTS

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTOPTS | NOSORTOPTS

Details

The SORTOPTS option specifies whether the host sort utility accepts the OPTIONS statement. The OPTIONS statement is generated by the host sort interface only if the 31-bit extended parameter list is requested via the SORT31PL option.

If the SORT31PL and NOSORTOPTS options are both specified, then not all of the available sort options can be passed to the host sort utility. This may cause the sort to fail. In particular, the sort work areas may not be used because the SORT option cannot be passed the value of the SORTWKDD= option.

You should therefore specify the DYNALLOC option, even though this may cause problems with multiple sorts within a single job. Older releases of some vendors' sort utilities dynamically allocate sort work files only if they are not already allocated. As a result, subsequent sorts might fail if they require more sort work space than the first sort.

SORTPARM=

Specifies a string of parameters to pass to your host sort utility

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTPARM='*string*'

string
is a string of parameters. It can contain up to 255 characters. Single quotes are optional unless *string* contains blanks or special characters.

Details

The string of parameters that you specify is appended to the OPTIONS statement that is generated by the SAS host sort interface. This enables you to specify options that are unique to the particular sort utility you are using. The sort utility must accept a 31-bit parameter list and an OPTIONS statement; otherwise, this option is ignored.

SORTPGM=

Specifies which sort utility to use

Default: BEST

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTPGM=*utility* | BEST | HOST | SAS

utility
can be any valid operating environment name that specifies the name of an accessible utility, except one of the three keywords for this option.

BEST
specifies a choice of sort utility of the data being sorted. The choice is made based on a comparison of the value of the SORTCUTP= option and a calculation of the number of bytes being sorted. If the number of bytes exceeds the value of SORTCUTP=, then the host sort utility is used. Otherwise, the SAS sort utility is used.

Details

The host sort utility may be more suitable than the sort utility supplied by SAS for SAS data sets that contain a large number of observations.

The name of the host sort utility is also given by the SORTNAME= system option.

See Also

□ "Efficient Sorting" on page 154

□ "SORTCUTP=" on page 396

□ "SORTNAME=" on page 401

SORTSHRB

Specifies whether the host sort interface can modify data in buffers

Default: SORTSHRB for all modes except batch; NOSORTSHRB for batch mode

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTSHRB | NOSORTSHRB

SORTSHRB

specifies that two or more tasks are likely to be sharing the data in buffers. If SORTSHRB is in effect, the host sort interface cannot modify data in buffers but must move the data first. This could have a severe performance impact, especially for large sorts.

SORTSHRB is the default value for the windowing environment, interactive line mode, and noninteractive mode, where it is quite likely that multiple tasks will be using the same data.

NOSORTSHRB

specifies that no tasks will be sharing the data in buffers. If NOSORTSHRB is in effect, the host sort interface can modify data in buffers. NOSORTSHRB is the default value for batch mode because it is unlikely that buffers will be shared during batch jobs, where larger sorts are usually run. If this is not suitable for your batch environment, be sure to specify SORTSHRB.

Details

SAS data sets can be opened for input by more than one SAS task (or window). When this happens, the buffers into which the data is read can be shared between the tasks. Because the host sort interface needs to modify the data before passing it to the host sort utility, and by default does this directly to the data in the buffers, data can be corrupted if more than one task is using the data in the buffers.

SORTSIZE=

Specifies the SIZE parameter that SAS is to pass to the sort utility

Default: MAX

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: valid values

Syntax

SORTSIZE=MAX | SIZE | *n* | *n*K | *n*M | *n*G

MAX

specifies that the characters MAX are to be passed to the system sort utility. This causes the sort utility to size itself. Not all sort utilities support this feature.

SIZE
specifies that the sort is to use the total amount of free space in the virtual machine minus the amount that is specified by the LEAVE= option in the PROC SORT statement.

n
specifies a number of bytes of memory to pass to the sort utility. If *n* is 0, the sort uses the default that was defined when it was installed.

***n*K**
specifies a number of kilobytes of memory to pass to the sort utility.

***n*M**
specifies a number of megabytes of memory to pass to the sort utility.

***n*G**
specifies a number of gigabytes of memory to pass to the sort utility.

SORTSUMF

Specifies whether the host sort utility supports the SUM FIELDS=NONE control statement

Default: SORTSUMF
Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window
Category: Sort: SORT
OS/390 specifics: all

Syntax

SORTSUMF | NOSORTSUMF

SORTSUMF
specifies that the host sort utility supports the SUM FIELDS=NONE control card.

NOSORTSUMF
specifies that the host sort utility does not support the SUM FIELDS=NONE control card. If NOSORTSUMF is in effect and the NODUPKEY option was specified when PROC SORT was invoked, then records that have duplicate keys are eliminated.

Details

If the NODUPKEY procedure option is specified when the SORT procedure is invoked, the SORTSUMF system option can be used to specify whether the host sort utility supports the SUM FIELDS=NONE control statement.

Note that duplicate keys are not the same as duplicate records. Duplicate keys can be eliminated with the NODUPKEY option, whereas duplicate records can be eliminated with the NODUP option in the PROC SORT statement.

SORTUADCON

Specifies whether the host sort utility supports passing a user address constant to the E15/E35 exits

Default: SORTUADCON

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTUADCON | NOSORTUADCON

SORTUADCON
specifies that the host utility supports passing a user address constant to the E15/ E35 exits.

NOSORTUADCON
specifies that the host sort utility does not support passing a user address constant to the E15/E35 exits.

SORTUNIT=

Specifies the allocation space units for sort work files

Default: CYLS

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTUNIT=CYL<S> | TRK<S> | BLK<S> | *n*

CYL<S>
specifies that the space units be cylinders. The space calculation for cylinder allocations requires that the characteristics of the device on which the allocations will be made need to be known. The device type is specified with the SORTDEV= option. The device type should be specified as generic, such as 3390, rather than esoteric, such as DISK. This is because when an esoteric name is specified, it is impossible to predict what device type will be used and thus the device characteristics.

TRK<S>
specifies that the space units be track(s). The space calculation for track allocations requires that the characteristics of the device on which the allocations will be made need to be known. The device type is specified with the SORTDEV= option. The device type should be specified as generic, such as 3390, rather than esoteric, such as DISK. This is because when an esoteric name is specified, it is impossible to predict what device type will be used and thus the device characteristics will also be unknown.

BLK<S>
specifies that the files will be allocated with an average block size equal to the record length rounded up to approximately 6K (6144). Therefore, if the input record length was 136, the average block size used for the allocation would be 6120.

n
is an integer that specifies the average block size.

Details

The SORTUNIT= option specifies the allocation space units to be used if SAS dynamically allocates the sort work files (see the DYNALLOC option).

SORTWKDD=

Specifies the prefix of sort work data sets

Default: SASS

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTWKDD=*prefix*

prefix
is a four-character, valid operating environment name, which must begin with a letter or a national character ($, #, or @), followed by letters, national characters, or digits.

Details

The SORTWKDD= option specifies the prefix to be used to generate the DDnames for the sort work files if SAS or the host sort utility dynamically allocates them (see “DYNALLOC” on page 340). The DDnames will be of the form *prefix*WK*nn*, where *nn* can be in the range of 01 to the value of the SORTWKNO= option, which is usually 3 and cannot exceed 6.

SORTWKNO=

Specifies how many sort work data sets to allocate

Default: 3

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORTWKNO=*n*

n

can be 0 through 6. If SORTWKNO=0 is specified, any existing sort work files are freed and none are allocated.

Details

The SORTWKNO= option specifies how many sort work files are to be allocated dynamically by either SAS or the SORT utility. (See “DYNALLOC” on page 340.)

SORT31PL

Controls what type of parameter list is used to invoke the host sort utility

Default: SORT31PL

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Sort: SORT

OS/390 specifics: all

Syntax

SORT31PL | NOSORT31PL

Details

If SORT31PL is in effect, a 31-bit extended parameter list is used to invoke the host sort utility. If NOSORT31PL is in effect, a 24-bit parameter list is used.

If SORT31PL is specified, then the SORTOPTS system option should also be specified. Also, because sorts that currently support a 31-bit parameter list also support the EQUALS option, the SORTEQOP system option should be specified in order to maximize performance.

STAE

Enables or disables a system ESTAE exit

Default: STAE

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Environment Control: ERRORHANDLING

OS/390 specifics: all

Syntax

STAE | NOSTAE

Details

The STAE option causes SAS's error trapping and handling to be activated by an ESTAE macro in the host supervisor.

STATS

Specifies whether statistics are to be written to the SAS log

Default: STATS
Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window
Category: System Administration: PERFORMANCE, Log and Procedure Output Control: LOGCONTROL
OS/390 specifics: all

Syntax

STATS | NOSTATS

STATS
tells SAS to write selected statistics to the SAS log.

NOSTATS
tells SAS not to write any statistics to the SAS log.

Details

The STATS system option specifies whether performance statistics are to be written to the SAS log. The statistics that are written to the log are determined by the MEMRPT, STIMER, and FULLSTATS system options.

See Also

□ "FULLSTATS" on page 356
□ "MEMRPT" on page 377
□ "STIMER" on page 410
□ "Collecting Performance Statistics" on page 150

STAX

Specifies whether to enable attention handling

Default: STAX

Valid in: configuration file, SAS invocation

Category: Environment Control: ERRORHANDLING

OS/390 specifics: all

Syntax

STAX | NOSTAX

STAX
causes attention handling to be activated by a STAX macro in the host supervisor.

NOSTAX
causes the SAS session to end when the attention key is pressed.

STIMER

Tells SAS whether to maintain system performance statistics

Default: STIMER

Valid in: configuration file, SAS invocation

Category: System Administration: PERFORMANCE

OS/390 specifics: all

Syntax

STIMER | NOSTIMER

STIMER
tells SAS to maintain the CPU time statistic. When the STATS option is also in effect, SAS writes the CPU time statistic to the SAS log.

NOSTIMER
tells SAS not to maintain the CPU time statistic.

Details

Additional statistics can be written to the SAS log by specifying the FULLSTATS or MEMRPT system options.

See Also

- □ "FULLSTATS" on page 356
- □ "MEMRPT" on page 377
- □ "STATS" on page 409
- □ "Collecting Performance Statistics" on page 150

SVC11SCREEN

Specifies whether to enable SVC 11 screening to obtain host date and time information

Default: NOSVC11SCREEN

Valid in: configuration file, SAS invocation

Category: System Administration: TESTING

OS/390 specifics: all

Syntax

SVC11SCREEN | NOSVC11SCREEN

SVC11SCREEN
causes SAS to issue SVC 11 to obtain the datetime.

NOSVC11SCREEN
causes SAS to use IBM's STCK instruction to obtain the datetime.

SYSIN=

Specifies the location of the primary SAS input data stream

Default: none (interactive), SYSIN (batch)

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: all

Syntax

SYSIN=*file-specification*

file-specification
identifies an external file. Under OS/390, it can be a valid DDname, a physical file name, or the name of a file stored in the directory structure of UNIX System

Services. The DDname must have been previously associated with an external file using either a TSO ALLOCATE command or a JCL DD statement.

Details

This option is applicable when you run SAS programs in noninteractive or batch mode.

SYSIN= is overridden by SYSINP= if a value for SYSINP= has been specified.

SYSINP=

Specifies the name of an external program that provides SAS input statements

Default: none

Valid in: configuration file, SAS invocation

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

SYSINP=*external-program-name*

external-program-name

identifies an external program, using eight characters or less.

Details

SAS calls this external program every time it needs a new SAS input statement. The PGMPARM= option (see "PGMPARM=" on page 387) enables you to pass a parameter to the program that you specify with the SYSINP= option.

The SYSINP= option overrides the SYSIN= system option.

SYSLEAVE=

Specifies how much memory to leave unallocated to ensure that SAS System tasks will be able to terminate successfully

Default: (0,153600)

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: System Administration: MEMORY

OS/390 specifics: all

Syntax

SYSLEAVE= *n* | *n*K | *n*M | (*n*| *n*K | *n*M, *n* | *n*K | *n*M)

***n* | *n*K | *n*M**
specifies in bytes, kilobytes, or megabytes how much memory space to leave unallocated above the 16-megabyte line. Unallocated space below the 16-megabyte line remains at its previous value, or at the default value. Valid values are any integer from 0 to the maximum amount of available space.

(*n*| *n*K | *n*M, *n* | *n*K | *n*M)
specifies in bytes, kilobytes, or megabytes how much memory space to leave unallocated below and above the 16-megabyte line respectively. Valid values are any integer from 0 to the maximum amount of available space.

See Also

- □ "Use SYSLEAVE= and PROCLEAVE= to handle out-of-memory conditions" on page 157

SYSPREF=

Specifies a prefix for partially qualified physical file names

Default: user profile prefix for interactive, userid for batch

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

SYSPREF=*prefix*

Details

The SYSPREF= option specifies a prefix to be used in constructing a fully qualified physical file name from a partially qualified name. Wherever a physical name must be entered in quotation marks in SAS statements or in SAS windowing environment commands, you may enter a data set name in the form '*.rest.of-name*', and SAS inserts the value of the SYSPREF= option in front of the first period.

Unlike the user profile prefix, the SYSPREF= option may have more than one qualifier in its name. If, for example, SYSPREF=SAS.TEST, then '.SASDATA' is interpreted as 'SAS.TEST.SASDATA'. The maximum length of *prefix* is 42 characters.

If no value is specified for SYSPREF=, then SAS uses the user profile prefix (in the interactive environment) or the userid (in batch).

SYSPRINT=

Specifies the handling of output that is directed to the default print file

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Log and Procedure Output Control: LISTCONTROL

OS/390 specifics: all

Syntax

SYSPRINT= * | DUMMY | *DDname*

*****
terminates redirection of output.

DUMMY
suppresses output to the default print file.

DDname
causes output to the default print file to be redirected to the specified DDname.

S99NOMIG

Tells SAS whether to recall a migrated data set

Default: NOS99NOMIG

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: all

Syntax

S99NOMIG | NOS99NOMIG

Details

The S99NOMIG option tells SAS what to do when a physical file that you reference (in a FILENAME statement, for example) has been migrated. If S99NOMIG is in effect, then the data set is not recalled and the allocation fails. If NOS99NOMIG is in effect, the data set is recalled, and allocation proceeds as it would have if the data set had not been migrated.

TAPECLOSE=

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

Default: REREAD

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: SASFILES

OS/390 specifics: default value, valid values

Syntax

TAPECLOSE=REREAD | LEAVE | REWIND | DISP | FREE

REREAD
leaves the tape volume positioned at the tapemark that precedes the file that was just closed. REREAD overrides a FREE=CLOSE specification in control language. Specify TAPECLOSE=REREAD if you access one or more tape data libraries several times in a SAS program.

LEAVE
leaves the tape volume positioned at the tapemark that follows the file that was just closed. LEAVE overrides a FREE=CLOSE specification in control language. Specify TAPECLOSE=LEAVE if you are not repeatedly accessing the same tape libraries in a SAS program, but instead you are creating or accessing one or more tape libraries in a subsequent file on the same tape volume.

REWIND
rewinds the tape volume to the beginning of the tape. A FREE=CLOSE specification in control language overrides the REWIND specification. Specify TAPECLOSE=REWIND if you are not repeatedly accessing one or more tape libraries in a SAS program.

DISP
positions the tape volume according to the disposition that is specified in the operating environment's control language.

FREE
rewinds the tape volume to the beginning of the tape and deallocates the tape drive.

USER=

Specifies the location of the default SAS data library

Default: none

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: Environment Conrol: ENVFILES

OS/390 specifics: *library-specification*

Syntax

USER=*library-specification*

library-specification
can be a DDname that was previously associated with a SAS data library, the name of a physical file that comprises a SAS data library, or a UNIX System Services directory.

See Also

- □ "Directing Temporary SAS Data Sets to the USER Library" on page 16
- □ *SAS Language Reference: Dictionary*

VECTOR

Specifies whether vector facility instructions will be used (if they are available)

Default: NOVECTOR

Valid in: configuration file, SAS invocation

Category: System Administration: INSTALLATION

OS/390 specifics: all

Syntax

VECTOR | NOVECTOR

Details

The VECTOR option specifies that vector facility instructions will be used if your system has the IBM 3090 vector facility installed.

VERBOSE

Writes the settings of SAS system options either to the terminal or to the batch job log

Default: NOVERBOSE

Valid in: configuration file, SAS invocation

Category: Log and Procedure Output Control: LOGCONTROL

OS/390 specifics: data written and where it is written

Syntax

VERBOSE | NOVERBOSE

Details

If you specify the VERBOSE system option at SAS invocation, the settings of all SAS system options that were set at SAS invocation or in the configuration files will be displayed in the following order:

1. settings in the system configuration file

2 settings in the user configuration file, if you have one

3 settings at SAS invocation.

If you specify the VERBOSE system option in a configuration file, only the options that are processed after VERBOSE is encountered are displayed. In other words, VERBOSE can appear in a configuration file, but the resulting options list then includes only those options that follow it in the configuration file.

If you invoke SAS at a terminal, the settings are displayed at the terminal. If you invoke SAS as part of a batch job, the settings are written to the batch job log.

See Also

□ "OPLIST" on page 384

VSAMLOAD

Enables you to load a VSAM data set

Default: NOVSAMLOAD

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

VSAMLOAD | NOVSAMLOAD

Details

The VSAMLOAD option must be in effect in order to load an empty VSAM data set.

See Also

□ *SAS Guide to VSAM Processing*

VSAMREAD

Enables the user to read a VSAM data set

Default: VSAMREAD

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

VSAMREAD | NOVSAMREAD

Details

The VSAMREAD option enables you to process VSAM data sets with a SAS DATA step.

See Also

- □ *SAS Guide to VSAM Processing*

VSAMUPDATE

Enables you to update a VSAM data set

Default: NOVSAMUPDATE

Valid in: configuration file, SAS invocation, OPTIONS statement, OPTIONS window

Category: File Control: EXTFILES

OS/390 specifics: all

Syntax

VSAMUPDATE | NOVSAMUPDATE

Details

The VSAMUPDATE option must be in effect in order to update VSAM data sets. Specifying VSAMUPDATE implies VSAMREAD.

See Also

- □ *SAS Guide to VSAM Processing*

WORK=

Specifies the location of the SAS WORK library

Default: WORK

Valid in: configuration file, SAS invocation

Category: Environment Control: ENVFILES

OS/390 specifics: *library-specification*

Syntax

WORK=*library-specification*

library-specification
can be a DDname that was previously associated with a SAS data library or the name of a physical file that comprises a SAS data library.

See Also

- "WORK Library" on page 14
- *SAS Language Reference: Dictionary*

XCMD

Enables the use of operating environment commands in a SAS session

Default: XCMD

Valid in: configuration file, SAS invocation

Category: Input Control: INPUTCONTROL

OS/390 specifics: all

Syntax

XCMD | NOXCMD

Details

If XCMD is in effect, you can issue operating environment commands through any of the available SAS interfaces, including the X command or the X statement; TSO command, statement, function, or call function; SYSTEM function or call routine; %TSO macro, or %SYSEXEC macro.

Summary Table of SAS System Options

The following table lists all the SAS system options that are available to OS/390 SAS users. The table gives you the following information about each SAS system option:

- the option name
- the default if you do not specify the option and if the option does not appear in the configuration file or in your site's default options table or restricted options table
- where you can specify the option

- □ where to look for more information about the option.

The Specified In column indicates where you can set or change the option's value. The following abbreviations represent these circumstances:

SI	SAS invocation
CF	Configuration file
OS	OPTIONS statement
OW	OPTIONS window

Some options have different default values depending on the mode in which the SAS System is running. For these options, the following conventions are used to distinguish the default values:

(b)	the default value in batch or noninteractive mode.
(i)	the default value in interactive line mode.
(w)	the default value in windowing environment mode.

The See column tell you when and where you can specify a given option. The following abbreviations are used in the See column to specify sources of information.

ADB	*SAS/ACCESS Interface to ADABAS Software: Reference*
CAM	*Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*
COMP	*SAS Companion for the OS/390 Environment*
CONN	*SAS/CONNECT User's Guide*
DB2	*SAS/ACCESS Software for Relational Databases: Reference (DB2 under OS/390 Chapter)*
DATACOM	*SAS/ACCESS Interface to CA-DATACOM/DB: Reference*
HELP	SAS help facility
IDMS	*SAS/ACCESS Interface to CA-IDMS Software: Reference*
IMS	*SAS/ACCESS Interface to IMS-DL/I Software*
INST	installation instructions for SAS in the OS/390 environment
LR	*SAS Language Reference: Dictionary*
ORACLE	*SAS/ACCESS Software for Relational Databases: Reference (ORACLE Chapter)*
SHARE	*SAS/SHARE User's Guide*

When two references are listed in the "See" column, the first reference is the primary source of information and should be consulted first.

Table 18.1 Summary of All SAS System Options Available under OS/390

Option Name	Default	Specified In	See
ADBBYMD	R	SI CF	ADB
ADBDBID	0	SI CF	ADB
ADBDBMD	M	SI CF	ADB

ADBDEFW	0	SI CF	ADB
ADBDEL	N	SI CF	ADB
ADBDELIM	\	SI CF	ADB
ADBFMTL	500	SI CF	ADB
ADBISNL	5000	SI CF	ADB
ADBL3	N	SI CF	ADB
ADBMAXM	191	SI CF	ADB
ADBMAXP	9	SI CF	ADB
ADBMINM	1	SI CF	ADB
ADBNATAP		SI CF	ADB
ADBNATPW		SI CF	ADB
ADBNATUS		SI CF	ADB
ADBRECL	7500	SI CF	ADB
ADBSCHL	500	SI CF	ADB
ADBSECCC		SI CF	ADB
ADBSECDB	0	SI CF	ADB
ADBSECFL	16	SI CF	ADB
ADBSECPW		SI CF	ADB
ADBSPANS	*	SI CF	ADB
ADBSYSCC		SI CF	ADB
ADBSYSDB	0	SI CF	ADB
ADBSYSFL	15	SI CF	ADB
ADBSYSPW		SI CF	ADB
ADBTASK	S	SI CF	ADB
ADBUISN	Y	SI CF	ADB
ADBUPD	Y	SI CF	ADB
ADBVALL	300	SI CF	ADB
ALTLOG		SI CF	COMP, LR
ALTPRINT		SI CF	COMP, LR
APPLETLOC		all	LR
ASYNCHIO	NOASYNCIO (b), ASYNCHIO (i, w)	SI CF	LR
AUTHENCR	OPTIONAL	all	CAM
AUTOEXEC	SASEXEC	SI CF	COMP, LR
BATCH	NOBATCH	SI CF	LR
BINDING	DEFAULT	all	LR
BLKALLOC	NOBLKALLOC	all	COMP
BLKSIZE	0	all	COMP

BLKSIZE(DISK)	0	all	COMP
BLKSIZE(OTHER)	6144	all	COMP
BLKSIZE(2301)	6144	all	COMP
BLKSIZE(2303)	4608	all	COMP
BLKSIZE(2305-1)	6144	all	COMP
BLKSIZE(2305-2)	6144	all	COMP
BLKSIZE(2314)	6144	all	COMP
BLKSIZE(3330)	6144	all	COMP
BLKSIZE(3330-1)	6144	all	COMP
BLKSIZE(3340)	6144	all	COMP
BLKSIZE(3350)	6144	all	COMP
BLKSIZE(3375)	8192	all	COMP
BLKSIZE(3380)	6144	all	COMP
BLKSIZE(3390)	6144	all	COMP
BLKSIZE(9345)	6144	all	COMP
BMPREAD	N	SI CF	IMS
BNDLSUFX		SI CF	INST
BOTTOMMARGIN		all	LR
BUFNO	1	all	LR
BUFSIZE	0	all	LR
BYERR	BYERR	all	LR
BYLINE	BYLINE	all	LR
CAPS	NOCAPS	all	LR
CAPSOUT	NOCAPSOUT	all	COMP
CARDIMAGE	NOCARDIMAGE	all	LR
CATCACHE	0	SI CF	LR
CBUFNO	0	all	LR
CENTER	CENTER	all	LR
CHARCODE	NOCHARCODE	all	LR
CHARTYPE	0	SI CF	COMP
CLEANUP	CLEANUP	all	LR
CLIST	NOCLIST	SI CF	COMP
CMDMAC	NOCMDMAC	all	LR
CMPOPT	CMPOPT	all	LR
COLLATE	NOCOLLATE	all	LR
COLORPRINTING	COLORPRINTING	all	LR
COMAMID		all	CONN, CAM
COMAUX1		SI CF	CAM

COMAUX2		SI CF	CAM
COMPRESS	NO	all	LR
CONFIG	CONFIG	SI	COMP, LR
CONNECTREMOTE		all	LR
CONNECTSTATUS	CONNECTSTATUS	all	LR
CONNECTWAIT	CONNECTWAIT	all	LR
CONSOLELOG	SASCLOG	SI CF	COMP, LR
COPIES	1	all	LR
CPUID	CPUID	SI CF	LR
DATASTMTCHK	COREKEYWORDS	all	LR
DATE	DATE	all	LR
DB2DECPT	.	SI CF	DB2
DB2IN		all	DB2
DB2PKCHK	N	SI CF	DB2
DB2PLAN	SASDB2E7	all	DB2
DB2RRS	NODB2RRS	SI CF	DB2
DB2SSID	DB2	all	DB2
DB2UPD	Y	SI CF	DB2
DBCS	NODBCS	SI CF	COMP
DBCSLANG		SI CF	COMP
DBCSTYPE	IBM	SI CF	COMP
DDBDBN		SI CF	DDB
DDBDELIM	\	SI CF	DDB
DDBLOAD	0	SI CF	DDB
DDBLOCK	0	SI CF	DDB
DDBMASK	#	SI CF	DDB
DDBMISS		SI CF	DDB
DDBPW		SI CF	DDB
DDBSPANS	*	SI CF	DDB
DDBSV	PROD	SI CF	DDB
DDBTASK	2	SI CF	DDB
DDBTRACE	0	SI CF	DDB
DDBUPD	Y	SI CF	DDB
DDBURT		SI CF	DDB
DDBUSER		SI CF	DDB
DETAILS	NODETAILS	all	LR
DEVICE		all	LR, COMP
DFLANG	ENGLISH	all	LR

DKRICOND	ERROR	all	LR
DKROCOND	WARN	all	LR
DLDMGACTION	REPAIR	all	LR
DLINITDEFER	NODLINITDEFER	all	COMP
DLIREAD	N	SI CF	IMS
DLTRUNCHK	NODLTRUNCHK	all	COMP
DMR	NODMR	SI CF	CONN
DMS	NODMS (i, b); DMS (w)	SI CF	LR
DMSEXP	NODMSEXP	SI CF	LR
DOCLOC		SI CF	LR
DSNFERR	DSNFERR	all	LR
DSRESV	NODSRESV	all	COMP
DUPLEX	NODUPLEX	all	LR
DYNALLOC	NODYNALLOC	all	COMP
ECHOAUTO	NOECHOAUTO	SI CF	LR
ENCRKEY		SI CF	INST
ENGINE		SI CF	LR
ERRORABEND	NOERRORABEND	all	LR
ERRORCHECK	NORMAL	all	LR
ERRORS	20	all	LR
EXPLORER	NOEXPLORER	SI CF	LR
FILEBLKSIZE(DISK)	0	all	COMP
FILEBLKSIZE(OTHER)	6400	all	COMP
FILEBLKSIZE(SYSOUT)	264	all	COMP
FILEBLKSIZE(TAPE)	0	all	COMP
FILEBLKSIZE(TERM)	264	all	COMP
FILEBLKSIZE(2301)	20483	all	COMP
FILEBLKSIZE(2303)	4892	all	COMP
FILEBLKSIZE (2305-1)	14136	all	COMP
FILEBLKSIZE (2305-2)	14660	all	COMP
FILEBLKSIZE(2311)	3625	all	COMP
FILEBLKSIZE(2314)	7294	all	COMP
FILEBLKSIZE(2321)	2000	all	COMP
FILEBLKSIZE(2400)	32760	all	COMP
FILEBLKSIZE(3330)	13030	all	COMP
FILEBLKSIZE (3330-1)	13030	all	COMP

FILEBLKSIZE(3340)	8368	all	COMP
FILEBLKSIZE(3350)	19069	all	COMP
FILEBLKSIZE(3375)	17600	all	COMP
FILEBLKSIZE(3380)	23476	all	COMP
FILEBLKSIZE(3390)	27998	all	COMP
FILEBLKSIZE(3400)	32760	all	COMP
FILEBLKSIZE(3480)	32760	all	COMP
FILEBLKSIZE(3490E)	32760	all	COMP
FILEBLKSIZE(3590)	32760	all	COMP
FILEBLKSIZE(9345)	22928	all	COMP
FILECC	NOFILECC	all	COMP
FILEDEST		all	COMP
FILEDEV	SYSDA	all	COMP
FILEDIRBLK	6	all	COMP
FILEEXT	IGNORE	all	COMP
FILEFORMS		all	COMP
FILEMOUNT	FILEMOUNT	all	COMP
FILEMSGS	NOFILEMSGS	all	COMP
FILENULL	FILENULL	all	COMP
FILEPROMPT	FILEPROMPT (i); NOFILEPROMPT (b)	all	COMP
FILEREUSE	NOFILEREUSE	all	COMP
FILESPPRI	1	all	COMP
FILESPSEC	1	all	COMP
FILESTAT	NOFILESTAT	all	COMP
FILESYSOUT		all	COMP
FILESYSTEM	MVS	all	COMP
FILEUNIT	CYLS	all	COMP
FILEVOL		all	COMP
FILSZ	FILSZ	all	COMP
FIRSTOBS	1	all	LR
FMTERR	FMTERR	all	LR
FMTSEARCH	(WORK LIBRARY)	all	LR
FORMCHAR	|—-|+|—+=|-/\<>*	all	LR
FORMDLIM		all	LR
FORMS	DEFAULT	all	LR
FSBCOLOR	NOFSBCOLOR	SI CF	COMP
FSBORDER	BEST	SI CF	COMP

FSDEVICE		SI CF	LR, COMP
FSMODE	IBM	SI CF	COMP
FULLSTATS	NOFULLSTATS	all	COMP
GHFONT		SI CF	COMP
GISMAPS		all	LR
GWINDOW	GWINDOW	all	LR
HELPLOC	HELPDOC	SI CF	COMP, LR
HSLXTNTS	1500	all	COMP
HSMAXPGS	75000	all	COMP
HSMAXSPC	50	all	COMP
HSSAVE	HSSAVE	all	COMP
HSWORK	NOHSWORK	SI CF	COMP
ICSRSLV	ONLY	SI CF	CONN
IDMWHST	I	SI CF	IDMS
IMPLMAC	NOIMPLMAC	all	LR
IMSBPAGN	*	all	IMS
IMSBPCPU	0	all	IMS
IMSBPDCA	0	all	IMS
IMSBPIN	*	all	IMS
IMSBPNBA	0	all	IMS
IMSBPOBA	0	all	IMS
IMSBPOPT	C	all	IMS
IMSBPOUT	*	all	IMS
IMSBPPAR	0	all	IMS
IMSBPSTI	0	all	IMS
IMSBPUPD	Y	SI CF	IMS
IMSDEBUG	N	all	IMS
IMSDLBKO	*	all	IMS
IMSDLBUF	16	all	IMS
IMSDLDBR	*	all	IMS
IMSDLEXC	0	all	IMS
IMSDLFMT	P	all	IMS
IMSDLIRL	*	all	IMS
IMSDLIRN	*	all	IMS
IMSDLLOG	0	all	IMS
IMSDLMON	N	all	IMS
IMSDLSRC	0	all	IMS
IMSDLSWP	*	all	IMS

IMSDLUPD	Y	SI CF	IMS
IMSID	*	SI CF	IMS
IMSIOB	*	all	IMS
IMSREGTP	DLI	SI CF	IMS
IMSSPIE	0	all	IMS
IMSTEST	0	all	IMS
IMSWHST	N	SI CF	IMS
INITCMD		SI CF	LR
INITSTMT		SI CF	LR
INVALIDDATA	.	all	LR
ISPCAPS	NOISPCAPS	all	COMP
ISPCHARF	NOISPCHARF	all	COMP
ISPCSR		SI CF	COMP
ISPEXECV		SI CF	COMP
ISPMISS		all	COMP
ISPMSG		SI CF	COMP
ISPNOTES	NOISPNOTES	all	COMP
ISPNZTRC	NOISPNZTRC	all	COMP
ISPPT	NOISPPT	all	COMP
ISPTRACE	NOISPTRACE	all	COMP
ISPVDEFA	NOISPVDEFA	all	COMP
ISPVDLT	NOISPVDLT	all	COMP
ISPVDTRC	NOISPVDTRC	all	COMP
ISPVIMSG		all	COMP
ISPVRMSG		all	COMP
ISPVTMSG		all	COMP
ISPVTNAM		all	COMP
ISPVTPNL		all	COMP
ISPVTRAP	NOISPVTRAP	all	COMP
ISPVTVARS		all	COMP
LABEL	LABEL	all	LR
LAST	_NULL_	all	LR
LEFTMARGIN		all	LR
LINESIZE	width of terminal (i, w); 132 (b)	all	COMP, LR
LOG	SASLOG	SI CF	COMP, LR
LUFIRST	1	SI CF	CAM
LULAST	9	SI CF	CAM

LUNAME		SI CF	CAM
LUPOOL	USER	SI CF	CAM
LUPREFIX		SI CF	CAM
LU62MODE		all	CAM
MACRO	MACRO	SI CF	LR
MAPS	MAPS	all	LR
MAUTOSOURCE	MAUTOSOURCE	all	LR
MEMLEAVE	524288	SI CF	COMP
MEMRPT	MEMRPT	all	COMP
MEMSIZE	varies, see dictionary	SI CF	COMP
MERGENOBY	NOWARN	all	LR
MERROR	MERROR	all	LR
MFILE	NOMFILE	all	LR
MINSTG	NOMINSTG	all	COMP
MISSING	.	all	LR
MLOGIC	NOMLOGIC	all	LR
MPRINT	NOMPRINT	all	LR
MRECALL	NOMRECALL	all	LR
MSG	SASMSG	SI CF	COMP
MSGCASE	NOMSGCASE	SI CF	COMP
MSGLEVEL	N	all	LR
MSGLOAD	NOMSGLOAD	all	COMP
MSGSIZE	131072	SI CF	COMP
MSTORED	NOMSTORED	all	LR
MSYMTABMAX	1048576	all	LR, COMP
MULTENVAPPL	NOMULTENVAPPL	all	LR
MVARSIZE	8192	all	LR, COMP
NDSVOLS		all	COMP
NETENCRYPT		all	CONN, SHARE
NETENCRYPTALGORITHM		all	CONN, SHARE
NETENCRYPTKEYLEN	0	all	CONN, SHARE
NETMAC	NETMAC	all	CONN, SHARE
NEWS		SI CF	LR
NOTES	NOTES	all	LR
NUMBER	NUMBER	all	LR
OBJECTSERVER	NOOBJECTSERVER	SI CF	LR
OBS	2147483647	all	LR
OPLIST	NOOPLIST	SI CF	COMP, LR

OPRESTRICTIONS		SI CF	INST
ORIENTATION	PORTRAIT	all	LR
OVP	NOOVP	all	LR
PAGENO	1	all	LR
PAGESIZE	terminal screen size (w); 21 (i); 60 (b)	all	LR, COMP
PAPERDEST		all	LR
PAPERSIZE	LETTER	all	LR
PAPERSOURCE		all	LR
PAPERTYPE	PLAIN	all	LR
PARM		all	LR
PARMCARDS	SASPARM	all	LR
PFKEY	PRIMARY	SI CF	COMP
PGMPARM		SI CF	COMP
PRINT	SASLIST	SI CF	COMP, LR
PRINTERPATH		all	LR
PRINTINIT	NOPRINTINIT	SI CF	COMP, LR
PRINTMSGLIST	PRINTMSGLIST	all	LR
PROBSIG	0	all	LR
PROCLEAVE	(0, 153600)	all	COMP
PSUPISA	174080	SI CF	INST
PSUPOSA	20480	SI CF	INST
REPLACE	REPLACE	all	LR
REUSE	NO	all	LR
REXXLOC	SASREXX	SI CF	COMP
REXXMAC	NOREXXMAC	all	COMP
RIGHTMARGIN		all	LR
RSASUSER	NORSASUSER	SI CF	LR
S	0	all	LR, COMP
SASAUTOS	SASAUTOS	all	COMP, LR
SASCMD		all	LR
SASFRSCR		all	CONN, INST
SASHELP	SASHELP	SI CF	COMP, LR
SASLIB	SASLIB	SI CF	COMP
SASMSTORE		all	LR
SASSCRIPT		all	CONN, INST
SASUSER	SASUSER	SI CF	COMP, LR
SEQ	8	all	LR

SEQENGINE	SASV7SEQ	all	COMP
SERROR	SERROR	all	LR
SETINIT	NOSETINIT	SI CF	LR
SKIP	0	all	LR
SMF	NOSMF	SI CF	INST
SMFEXIT		SI CF	INST
SMFTYPE	128	SI CF	INST
SOLUTIONS	SOLUTIONS	SI CF	LR
SORT	0	all	COMP
SORTALTMSGF	NOSORTALTMSGF	all	COMP
SORTBLKMODE	NOSORTBLKMODE	all	COMP
SORTBUFMOD	SORTBUFMOD	all	COMP
SORTCUTP	4194304	all	COMP
SORTDEV	SYSDA	all	COMP
SORTDEVWARN	SORTDEVWARN	all	COMP
SORTDUP	PHYSICAL	all	LR
SORTEQOP	SORTEQOP	all	COMP
SORTLIB	SYS1.SORTLIB	all	COMP
SORTLIST	NOSORTLIST	all	COMP
SORTMSG	NOSORTMSG	all	COMP
SORTMSG	SYSOUT	all	COMP
SORTNAME	SORT	all	COMP
SORTOPTS	SORTOPTS	all	COMP
SORTPARM		all	COMP
SORTPGM	BEST	all	COMP
SORTSEQ		all	LR
SORTSHRB	SORTSHRB (i,w); NOSORTSHRB (b)	all	COMP
SORTSIZE	2147483647	all	COMP, LR
SORTSUMF	SORTSUMF	all	COMP
SORTUADCON	SORTUADCON	all	COMP
SORTUNIT	CYLS	all	COMP
SORTWKDD	SASS	all	COMP
SORTWKNO	3	all	COMP
SORT31PL	SORT31PL	all	COMP
SOURCE	NOSOURCE	all	LR
SOURCE2	NOSOURCE2	all	LR
SPOOL	NOSPOOL	all	LR

STAE	STAE	all	COMP
STARTLIB	NOSTARTLIB	SI CF	LR
STATS	STATS	all	COMP
STAX	STAX	SI CF	COMP
STIMER	STIMER	SI CF	COMP
SUBSYSID	SAS0	SI CF	INST
SUMSIZE	8388608	all	LR
SVC0R15	4	SI CF	INST
SVC0SVC	109	SI CF	INST
SVC11SCREEN	NOSVC11SCREEN	SI CF	COMP
SYMBOLGEN	NOSYMBOLGEN	all	LR
SYNCHIO	SYNCHIO (b); NOSYNCHIO (i,w)	SI CF	LR
SYSIN	none (i,w); SYSIN (b)	SI CF	COMP
SYSINP		SI CF	COMP
SYSLEAVE	(0, 153600)	all	COMP
SYSPARM		all	LR
SYSPREF	user profile prefix (i, w); userid (b)	all	COMP
SYSPRINT		all	COMP
S2	0	all	LR
S99NOMIG	NOS99NOMIG	all	COMP
TAPECLOSE	REREAD	all	COMP
TBUFSIZE	0	all	CONN
TCPIPMCH		SI CF	CAM
TCPIPPRF		SI CF	CAM
TCPPORTFIRST	0	all	CAM
TCPPORTLAST	0	all	CAM
TERMINAL	TERMINAL	SI CF	LR
TOPMARGIN		all	LR
TRAINLOC		SI CF	LR
TRANTAB		all	CONN, LR
USER		all	COMP, LR
USERXIT1		SI CF	INST
USERXIT2		SI CF	INST
VALIDVARNAME	V7	all	LR
VECTOR	NOVECTOR	SI CF	COMP, INST
VERBOSE	NOVERBOSE	SI CF	COMP
VMCTLISA	163840	all	INST

VMLOADI	0	all	INST
VMLOADO	0	all	INST
VMNSISA	0	SI CF	INST
VMNSOSA	0	SI CF	INST
VMPAISA	262144	all	INST
VMPAOSA	131072	all	INST
VMPBISA	262144	all	INST
VMPBOSA	131072	all	INST
VMTAISA	262144	all	INST
VMTAOSA	131072	all	INST
VMTBISA	262144	all	INST
VMTBOSA	131072	all	INST
VNFERR	VNFERR	all	LR
VSAMLOAD	NOVSAMLOAD	all	COMP
VSAMREAD	VSAMREAD	all	COMP
VSAMUPDATE	NOVSAMUPDATE	all	COMP
WORK	WORK	SI CF	COMP, LR
WORKINIT	WORKINIT	SI CF	LR
WORKTERM	WORKTERM	all	LR
XCMD	XCMD	SI CF	COMP
YEARCUTOFF	1920	all	LR

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.