



SAS[®] Viya[™] 3.2 System Options: Reference

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS® Viya™ 3.2 System Options: Reference*. Cary, NC: SAS Institute Inc.

SAS® Viya™ 3.2 System Options: Reference

Copyright © 2017, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

March 2017

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

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

3.2-P1:lesysoptsfref

Contents

What's New in SAS Viya 3.2 System Options v

PART 1 About SAS System Options 1

Chapter 1 • What You Need to Know	3
Definition of System Options	3
Other System Option Documentation	3
Using SAS System Options	4
Comparisons	17

PART 2 SAS System Options 19

Chapter 2 • Dictionary of SAS System Options	21
SAS System Options Documented in Other SAS Publications	24
SAS System Options by Category	24
Dictionary	37

PART 3 SAS Functions and Statements That Process SAS System Options 235

Chapter 3 • SAS Functions That Process SAS System Options	237
Dictionary	237
Chapter 4 • SAS Statements That Process SAS System Options	243
Dictionary	243

PART 4 SAS Procedures That Process SAS System Options 245

Chapter 5 • OPTIONS Procedure	247
Overview: OPTIONS Procedure	247
Syntax: OPTIONS Procedure	248
Displaying a List of System Options	252
Displaying Information about One or More Options	253
Displaying Information about System Option Groups	255
Displaying Restricted Options	257
Results: OPTIONS Procedure	258
Examples: OPTIONS Procedure	259

PART 5 Appendix 265

Appendix 1 • Time Zone IDs and Time Zone Names	267
Area: Africa	267
Area: America–North, Central, and South	270
Area: Antarctica	279
Area: Asia	279
Area: Atlantic	283
Area: Australia	284
Area: Miscellaneous	286
Area: Europe	291
Area: Pacific	296
Recommended Reading	299
Index	301

What's New in SAS Viya 3.2 System Options

Connecting to SAS Viya Services

You use the [SERVICESBASEURL=](#) system option when you start SAS in batch or command line mode to connect to the SAS Viya services.

Byte Value Notation

For the system options whose values allow the byte notation K, M, G, or T, you can now specify KB, MB, GB, and TB.

Part 1

About SAS System Options

Chapter 1

What You Need to Know 3

Chapter 1

What You Need to Know

Definition of System Options	3
Other System Option Documentation	3
Using SAS System Options	4
Default Values	4
Changing the Default Option Values	4
Determining Which Settings Are in Effect	6
How Long System Option Settings Are in Effect	7
Restricted Options	8
Determining How a SAS System Option Value Was Set	10
Obtaining Descriptive Information about a System Option	12
Changing an Option Value By Using the INSERT and APPEND System Options	13
Resetting System Options to the Default or Starting Value	15
Order of Precedence for Processing SAS Configuration Files	16
Order of Precedence for Processing SAS System Options	17
Interaction with Data Set Options	17
Comparisons	17

Definition of System Options

System options are instructions that affect the processing of an entire SAS program or SAS session from the time the option is specified until it is changed. Examples of items that are controlled by SAS system options include the appearance of SAS output, the handling of some files that are used by SAS, the use of system variables, the processing of rows (observations) in SAS data sets, features of SAS initialization, and how SAS interacts with your operating environment.

Other System Option Documentation

Some SAS system options effect only individual SAS components or products. For example, SAS has system options specifically for Cloud Analytic Services and SAS/CONNECT. These system options reside in the documentation for these components and products. For links to these documents, see [“SAS System Options Documented in Other SAS Publications” on page 24](#) .

The documentation for all other system options is in this publication.

Using SAS System Options

Default Values

SAS system options are initialized with shipped default values when SAS is invoked. Configuration files can be created by a SAS administrator to define default system options values for a site. If you run your programs in batch or interactive line-mode, you can create your own configuration file to set default option values for your SAS session.

For information about creating a customized configuration file, see [“Customizing Your SAS Session By Using Configuration and Autoexec Files”](#) in *Batch and Line Mode Processing in SAS Viya*.

Changing the Default Option Values

Where You Can Change Default Option Values

You can change default option values using one of these methods:

- [in a configuration file](#)
- [on the SAS command line](#)
- [in an OPTIONS statement](#)
- [in an OPTIONS statement in the AUTOEXEC file](#)
- [setting the SASV9_OPTIONS environment variable](#)

Setting System Option Values That Include Spaces

If the value of a system option includes a space, you must enclose the value in quotation marks. The following examples show the correct syntax when you specify these options in a configuration file or on the command line:

```
-bufsize '3 k';  
-bottommargin '2 in';
```

If the value of a system option does not include a space, you do not need to enclose the value in quotation marks:

```
-bufsize 3k;  
-bottommargin 2in;
```

Setting System Options in a Configuration File

If you use the same unrestricted option settings frequently, then it is usually more convenient to specify the options in a configuration file. Typically, your SAS administrator creates a configuration file for your site. You can create a configuration file in your Home directory to set the system option values that you want to use. SAS has an [order of precedence](#) that it uses to process configuration files.

Specify system options in the file by preceding each with a hyphen. For ON or OFF options, just list the keyword corresponding to the appropriate setting. For options that accept values, list keyword identifying the option followed by the option value or, if appropriate, multiple values. All SAS system options can appear in a configuration file.

For more information, see [“Customizing Your SAS Session By Using Configuration and Autoexec Files”](#) in *Batch and Line Mode Processing in SAS Viya*.

Setting System Options on the Command Line

You can specify SAS system options in the SAS command. Precede each option with a hyphen:

```
sas -option1 -option2...
```

For ON or OFF options, list the keyword corresponding to the appropriate setting. For options that accept values, list the keyword that identifies the option, followed by the option value or, if appropriate, multiple values. Here are some examples:

```
sas -noaltlog
sas -autoexec /path/autoexec.sas
```

Settings that you specify in the SAS command last for the duration of the SAS session or, for those options that can be changed within the session, until you change them. All options can be specified in the SAS command.

Setting System Options Using an OPTIONS Statement within a SAS Session

Some options can be specified only when SAS starts. Most system options can be specified in an OPTIONS statement at any time during a SAS session except within data lines or parmcards. The options are set for the duration of the SAS session or until you change them. When you use the SAS Studio Background Submit feature, the options that you set in an OPTIONS statement do not effect the SAS Studio session. Background Submit creates a new SAS session.

The syntax for specifying system options in an OPTIONS statement is

```
OPTIONS option(s);
```

option(s)

specifies one or more SAS system options that you want to change.

Do not precede its name with a hyphen (-). If the option has an argument, use = after the option name. Here is an example:

```
options bufno=2g;
options maps='/usr/maps/mymaps';
```

To specify multiple options, separate each option using a space.

In general, use quotation marks to enclose filenames and pathnames specified in the OPTIONS statement. Do not use quotation marks otherwise. Any exceptions are discussed under the individual option.

By specifying either the INSERT or APPEND option in the OPTIONS statement, you can add a value to certain system options that name libraries or files, such as the FMTSEARCH option. For more information, see [“Changing an Option Value By Using the INSERT and APPEND System Options”](#) on page 13.


Setting System Options Using an OPTIONS Statement in an Autoexec File

An autoexec file contains SAS statements that are executed automatically when SAS is invoked. The autoexec file can be used to specify some SAS system options, as well as to assign librefs and filerefs to data sources that are used frequently. For example, your autoexec file could contain the following statements:

```
options bufno=2g;
```

```
filename rpt '/users/myid/data/report';
```

In general, use quotation marks to enclose filenames and pathnames specified in the `OPTIONS` statement. Do not use quotation marks otherwise. Any exceptions are discussed under the individual option.

In SAS Studio, you can add options to the Autoexec file by selecting  and **Edit Autoexec File**. Because not all options can be set with an `OPTIONS` statement in an Autoexec file, be sure to run the Autoexec file to test if the options were set.

For updating the Autoexec file for batch or interactive line modes, see [“Customizing Your SAS Session By Using Configuration and Autoexec Files”](#) in *Batch and Line Mode Processing in SAS Viya*

Set System Options Using an Environment Variable

Specify SAS system options in the `SASV9_OPTIONS` environment variable before you invoke SAS. Settings that you specify in the `SASV9_OPTIONS` environment variable affect SAS sessions that are started when the variable is defined.

For example, in the Korn shell, you would use the following:

```
export SASV9_OPTIONS='-fullstimer -nodate'
```

For information about defining Linux environment variables, see [“Defining Environment Variables in Linux Environments”](#) in *Batch and Line Mode Processing in SAS Viya*.

Processing System Options That Are Set More Than Once

If the same system option is set more than once in a configuration file or in the `SASV9_OPTIONS` environment variable, the most recent specification is the value that SAS uses. The other specifications are ignored. For example, the `BUFNO` option that is set to 3k is ignored in the following configuration file:

```
-bufno 3k
-bottommargin 2in
-bufno 4k
```

By default, if you specify the `MSG` or `SASAUTOS` system option more than once, the most recent specification is the value that SAS uses. If you want to add additional pathnames to the pathnames already specified by one of these options, you must use the `APPEND` or `INSERT` system option. For more information, see [“Changing an Option Value By Using the INSERT and APPEND System Options”](#) on page 13.

Specifying Hexadecimal Values

Hexadecimal values for system options must begin with a number (0-9), followed by an X. For example, the following `OPTIONS` statement sets the line size to 160 using a hexadecimal number:

```
options linesize=0a0x;
```

Character assignments for hexadecimal numbers require quotation marks:

```
options formchar='a0'x;
```

Determining Which Settings Are in Effect

To determine which settings are in effect for SAS system options, use one of the following:

OPLIST system option

Writes to the SAS log the system options that were specified on the SAS invocation command line. The OPLIST option can be specified only when SAS starts. SAS Studio sets this option to NOOPLIST.

VERBOSE system option

Writes to the SAS log the system options that were specified in the configuration file and on the SAS invocation command line.

OPTIONS procedure

Writes system option settings to the SAS log. To display the settings of system options with a specific functionality, such as error handling, use the GROUP= option: **proc options GROUP=errorhandling; run;** For more information, see [Chapter 5, “OPTIONS Procedure,” on page 247](#).

GETOPTION function

Returns the value of a specified system option. For example:

```
%put %sysfunc(getoption(datestyle,keyword));
```

VOPTION Dictionary table

Located in the Sashelp library, VOPTION contains a list of all current system option settings, a description of each option, the option type, whether the option is a portable or a host option, when the option can be set, and the group to which the option belongs. You can print the table by using the PRINT procedure or you can extract information from the VOPTION table by using SQL or the DATA step.

How Long System Option Settings Are in Effect

System Options in SAS Studio

In SAS Studio, the value for most system options applies to the next step and to all subsequent steps for the duration of the SAS session, or until you reset the system option setting.

SAS Studio sets several system options before and after each code submission to create a consistent environment. If you modify any of these options in your code, the values that you set do not persist. You must resubmit your OPTIONS statement each time you submit your code. A code submission is either all of the code in the programming tab or code that you highlight and submit.

To see in the log the options that SAS Studio sets, you can check the SAS Studio preference **Show generated code in the SAS log**.

SAS Studio sets these options before your code executes:

- DATE
- DFLANG
- DTRESET
- FIRSTOBS=1
- LOCALE
- NOSYNTAXCHECK
- NOTES
- NUMBER
- OBS=MAX

- PRINTERPATH=PDF
- VALIDMEMNAME=COMPAT
- VALIDVARNAME= is set to the **VALIDVARNAME** Preferences value.

For example, if you submit the following three lines of code, the value of the OBS option is 5 after SAS prints five rows of Sashelp.cars. If you submit each line of code separately, PROC PRINT prints the complete Sashelp.Cars data set instead of five rows because SAS Studio sets OBS=MAX each time code is submitted.

```
options obs=5;
proc print data=sashelp.cars;
run;
```

SAS Studio sets these system options after each code submission:

- NOTES
- SOURCE
- STIMER
- SYNTAXCHECK
- VALIDMEMNAME=COMPAT

System Options in Batch Mode

When you specify a SAS system option setting, the setting applies to the next step and to all subsequent steps for the duration of the SAS session, or until you reset the system option setting, as shown:

```
data one;
  set items;
run;

  /* option applies to all subsequent steps */
options obs=5;

  /* printing ends with the fifth observation */
proc print data=one;
run;

  /* the SET statement stops reading
  after the fifth observation */
data two;
  set items;
run;
```

To read more than five observations, you must reset the OBS= system option. For more information, see [“OBS= System Option” on page 142](#) .

Restricted Options

Restricted options are system options whose values are determined by the site administrator and cannot be overridden. The site administrator can create a restricted options table that specifies the option values that are restricted when SAS starts. Any attempt to modify a system option that is listed in the restricted options table results in a

message to the SAS log indicating that the system option has been restricted by the site administrator and cannot be updated.

PROC OPTIONS has two options that list restricted options:

RESTRICT lists the options that currently are restricted by your site administrator.

LISTRESTRICT lists the options that can be restricted by your site administrator.

To determine which system options are restricted by your site administrator, use the **RESTRICT** option of the **OPTIONS** procedure. The **RESTRICT** option displays the option's value, scope, and setting. In the following example, the SAS log shows that only one option, **CMPOPT**, is restricted:

```
proc options restrict;
run;
```

Log 1.1 *Restricted Option Information*

```
1  proc options restrict;
2  run;
   SAS (r) Proprietary Software Release V.03.02 TS1M0

Option Value Information For SAS Option BYLINE
  Value: NOBYLINE
  Scope: SAS Session
  How option value set: Site Administrator Restricted
```

The **OPTIONS** procedure displays this information for all options that are restricted. If your site administrator has not restricted any options, then the following message appears in the SAS log:

Your site administrator has not restricted any options.

You can use the **OPTIONS** procedure option **LISTRESTRICT** to view the options that your site administrator can restrict. These are not options that are restricted, but can be restricted. Your output might be different.

```
proc options listrestrict;
run;
```

Log 1.2 Partial Listing of Options That Can Be Restricted

```

13 proc options listrestrict ;
14 run;

SAS (r) Proprietary Software Release 9.4 TS1M4

Your Site Administrator can restrict the ability to modify the following Portable Options:

ANIMATION           Specifies whether to start or stop animation.
ANIMDURATION         Specifies the number of seconds that each animation frame displays.
ANIMLOOP            Specifies the number of iterations that animated images repeat.
ANIMOVERLAY         Specifies that animation frames are overlaid in order to view all frames.
APPLETLOC           Specifies the location of Java applets, which is typically a URL.
ARMAGENT            Specifies an ARM agent (which is an executable module or keyword, such
                    as LOG4SAS) that contains a specific implementation of the ARM API.
ARMLOC              Specifies the location of the ARM log.
ARMSUBSYS           Specifies the SAS ARM subsystems to enable or disable.
AUTOCORRECT         Automatically corrects misspelled procedure names and keywords, and
                    global statement names.
AUTOSAVELOC         Specifies the location of the Program Editor auto-saved file.
AUTOSIGNON          Enables a SAS/CONNECT client to automatically submit the SIGNON command
                    remotely with the RSUBMIT command.

BINDING             Specifies the binding edge type of duplexed printed output.
BUFNO               Specifies the number of buffers for processing SAS data sets.
BUFSIZE            Specifies the size of a buffer page for output SAS data sets.
BYERR               SAS issues an error message and stops processing if the SORT procedure
                    attempts to sort a _NULL_ data set.
BYLINE              Prints the BY line above each BY group.
BYSORTED            Requires observations in one or more data sets to be sorted in
                    alphabetic or numeric order.
CAPS                Converts certain types of input, and all data lines, into uppercase
                    characters.

```

For information, see [Chapter 5, “OPTIONS Procedure,”](#) on page 247.

The following system options cannot be restricted:

ALIGNSASIOFILES	INITSTMT	MSYMTABMAX	SSLPKCS12LOC
ALTLOG	INSERT	MVARSIZE	SSLPKSC12PASS
APPEND	_LAST_	OBJECTSERVER	SSPI
AUTOEXEC	LEFTMARGIN	ORIENTATION	STARTLIB
BOTTOMMARGIN	LINESIZE	PAGESIZE	SYSIN
CONFIG	LOG	PAPERSIZE	SYSPRINTFONT
CPUCOUNT	LOGPARM	PDFPASSWORD	TOPMARGIN
DATESTYLE	METAPASS	PRINT	TRANTAB
DBCS	METAPROTOCOL	PRINTERPATH	UBUFNO
DFLANG	METAREPOSITORY	RIGHTMARGIN	UBUFSIZE
DLDMGACTION	METASERVER	SASUSER	USER
FILELOCKWAITMAX	METAUSER	SOURCE	

Determining How a SAS System Option Value Was Set

System options can be set by using one of these methods:

- Shipped default
- SAS command line

- SAS configuration file
- Locale setting
- Environment variable
- Administrator restricted
- OPTIONS statement

To determine how a system option value was set, use either the OPTIONS procedure or the GETOPTION function:

- Use the OPTIONS procedure with the VALUE option specified in the OPTIONS statement. The VALUE option displays the specified option's value and scope.
- Use the GETOPTION function as an argument to the %SYSFUNC macro function:

```
%put %sysfunc(getoption(option-name, howset));
```

This log output shows the value of the CBUFNO option:

```
57          %put %sysfunc(getoption(cbufno, howset));
Shipped Default
```

This example shows how the option value for the system option CENTER was set using the OPTIONS procedure:

```
proc options option=center value;
run;
```

The following partial SAS log shows that the option value for CENTER was the shipped default.

Log 1.3 Option Value Information for the System Option CENTER

```
SAS (r) Proprietary Software Release V.03.02 TS1M0

Option Value Information For SAS Option CENTER
Value: CENTER
Scope: Default
How option value set: Shipped Default
```

If a SAS option is set from a configuration file, SAS displays the name of the configuration file that set the option.

Log 1.4 Option Value Information Showing an Option Set by a Configuration File

```
7  proc options option=work value;
8  run;

SAS (r) Proprietary Software Release V.03.02 TS1M0

Option Value Information For SAS Option WORK
Value: /tmp/SAS_workD54300006E06_server15179/
SAS_work775D00006E06_server15179
Scope: IOM ROOT COMP ENV
How option value set: Config File
Config file name:
/opt/sas/viya/SASFoundation/sasv9.cfg
```

If a SAS option is modified using the INSERT or APPEND system options, you can use the VALUE option in a PROC OPTIONS statement to show that the value was inserted or appended:

Log 1.5 *Option Value Information for an Option Modified by the INSERT and APPEND Options*

```

57      options insert=(fmtsearch "/home/cas");
58      options append=(fmtsearch "/home/sas");
59      proc options option=fmtsearch value; run;

SAS (r) Proprietary Software Release V.03.02 TS1M0

Option Value Information For SAS Option FMTSEARCH
Value: ('/HOME/CAS' WORK LIBRARY '/HOME/SAS')
Scope: IOM ROOT COMP ENV
How option value set: Options Statement
Value Inserted: '/HOME/CAS'

How option value set: Shipped Default
Value: WORK LIBRARY

How option value set: Options Statement
Value Appended: '/HOME/SAS'

```

If no value is assigned to a character system option, then SAS assigns the option a value of ' ' (a space between two single quotation marks) and **Option Value** displays a blank space.

Obtaining Descriptive Information about a System Option

You can quickly obtain basic descriptive information about a system option by specifying the DEFINE option in the PROC OPTIONS statement.

The DEFINE option writes the following descriptive information about a system option to the SAS log:

- the value of the option
- a description of the option
- the name and description of each system option group that the option is a part of
- type information, such as whether it is numeric or character, whether to expand a value that is an environment variable, and valid values for the option
- when in the SAS session that it can be set
- if it can be restricted by the system administrator

For example, the following statements write a message to the SAS log that contains descriptive information about the system option CASNWORKERS:

```

proc options option=casnworkers define;
run;

```

Log 1.6 Descriptive Information about the System Option CASWORKERS

```

57      proc options option=casworkers define;
58      run;

SAS (r) Proprietary Software Release V.03.02 TS1M0

CASWORKERS=ALL
Option Definition Information for SAS Option CASWORKERS
Group= CAS
Group Description: Cloud Analytic Services settings
Description: Specify the number of workers to use with a CAS session.
Type: The option value is of type CHARACTER
Maximum Number of Characters: 10
Casing: The option value is retained with original casing
Quotes: If present during "set", start and end quotes are retained
Parentheses: The option value does not require enclosure within parentheses. If present,
the parentheses are retained.
Expansion: Environment variables, within the option value, are not expanded
When Can Set: Startup or anytime during the SAS Session
Restricted: Your Site Administrator can restrict modification of this option

```

Changing an Option Value By Using the INSERT and APPEND System Options

You can use the INSERT and APPEND options to modify the values of these options:

Option	Where the Option Can Be Set
AUTOEXEC	configuration file, SAS invocation
FMTSEARCH	OPTIONS statement,
MSG	configuration file, SAS invocation
SASAUTOS	configuration file, SAS invocation, OPTIONS statement
SASHELP	configuration file, SAS invocation
SET	configuration file, SAS invocation

The value of these options is one or more libraries, files, or environment variables. You use the INSERT option to insert a value before the current value. You use the APPEND option to append a value to the end of the current value. Use the LISTINSERTAPPEND option in the PROC OPTIONS statement to see a list of options in the SAS log that you can use with the INSERT option and the APPEND option. Your output might be different.

```

proc options listinsertappend;
run;

```

```

56          proc options listinsertappend; run;

SAS (r) Proprietary Software Release V.03.02  TS1M0

Core options that can utilize INSERT and APPEND

AUTOEXEC          Specifies the location of the SAS AUTOEXEC files.
FMTSEARCH         Specifies the order in which format catalogs are searched.
SASAUTOS         Specifies the location of one or more autocall libraries.
SASHELP          Specifies the location of the Sashelp library.
SASSCRIPT        Specifies one or more locations of SAS/CONNECT server sign-on script files.

Host options that can utilize INSERT and APPEND

MSG              Specifies the path to the library that contains SAS messages.
SET             Defines an environment variable.

```

If you specify the INSERT option or the APPEND option after SAS starts, the syntax requires parentheses:

```
insert=(system-option-1=argument-1 system-option-n=argument-n)
```

```
append=(system-option-1=argument-1 system-option-n=argument-n)
```

The syntax for *system-option=argument* is the syntax that is required for the specified system option.

Here are two examples:

```
options insert=(fmtsearch="/u/home/cas");
options append=(set="/u/home/myprog2");
```

For more information, see [“INSERT= System Option” on page 99](#) and [“APPEND= System Option” on page 39](#).

You can use the VALUE option in the PROC OPTIONS statement to display values that have been inserted or appended to an option value:

```

57          proc options option=fmtsearch value; run;

          SAS (r) Proprietary Software Release V.03.02  TS1M0

Option Value Information For SAS Option FMTSEARCH
Value: ('/U/HOME/CAS' WORK LIBRARY)
Scope: IOM ROOT COMP ENV
How option value set: Options Statement
Value Inserted: '/U/HOME/CAS'

How option value set: Shipped Default
Value: WORK LIBRARY

```

SAS inserts or appends values as they are specified in the INSERT and APPEND system options. SAS does not check for duplicate values.

The INSERT and APPEND system options add only values to a system option's value. To delete a value from a system option, set the option to the value that you want.

Resetting System Options to the Default or Starting Value

Reset Options to the Default or Starting Value By Using the %SYSFUNC Macro Function and the GETOPTIONS Function

You can use macro processing and the GETOPTION function together to set a system option to the default value or to the value that was specified when SAS started.

You use the GETOPTION function DEFAULTVALUE option to set a system option its default value. You use the GETOPTION function STARTUPVALUE option to set a system option to the starting value.

The following code is an example of setting the DATESTYLE= system option to YMD and then to its default value:

```

/* Check the value of datestyle before we change it.          */
                                                                    */

/* The current value is DMY as this value was set in the      */
/* autoexec file when SAS Studio started.                    */
                                                                    */

    %put %sysfunc(getoption(datestyle,keyword));

/* Change the DATESTYLE value and check the change.          */
                                                                    */

    options datestyle='YMD';

    %put %sysfunc(getoption(datestyle,keyword));

/* Change DATESTYLE back to the default value and check it.  */
/* RESULT:  MDY                                              */
                                                                    */

    %let defstyl = %sysfunc(getoption(datestyle,keyword,defaultvalue)) ;
    options &defstyl; run;
    %put %sysfunc(getoption(datestyle,keyword));

```

The SAS log displays the following lines:

```

55
56      /* Check the value of datestyle before we change it.      */
57
58      /* The current value is DMY as this value was set in the      */
59      /* autoexec file when SAS Studio
started.                    */
60
61      %put %sysfunc(getoption(datestyle,keyword));
DATESTYLE=DMY
62
63      /* Change the DATESTYLE value and check the change.      */
64
65      options datestyle='YMD';
66
67      %put %sysfunc(getoption(datestyle,keyword));
DATESTYLE=YMD
68
69      /* Change DATESTYLE back to the default value and check it. */
70      /* RESULT:  MDY      */
71
72      %let defstyl =
%sysfunc(getoption(datestyle,keyword,defaultvalue)) ;
73      options &defstyl; run;
74      %put %sysfunc(getoption(datestyle,keyword));
DATESTYLE=MDY
75

```

For more information, see [GETOPTION function on page 237](#).

Order of Precedence for Processing SAS Configuration Files

SAS is shipped with a default configuration file in the !SASROOT directory. Your SAS administrator can edit this configuration file so that it contains whichever options are appropriate to your site.

If you run SAS programs in batch mode, you can also create one or more of your own configuration files.

Note: If you create a configuration file in your home directory and the NOUSERCONFIG= system option is set, the configuration file in your home directory is skipped. For more information, see [“USERCONFIG System Option” on page 209](#).

SAS reads option settings from each of these files in the following order:

1. sasv9.cfg in the !SASROOT directory.
2. sasv9_local.cfg in the !SASROOT directory.
3. .sasv9.cfg in your home directory. (Notice the leading period.)
4. sasv9.cfg in your home directory.
5. sasv9.cfg in your current directory.
6. any restricted configuration files. Restricted configuration files contain system options that are set by the site administrator and cannot be changed by the user. Options can be restricted globally, by group, or by user. For more information about restricted configuration files, see the configuration guide for the Linux environment.

For future releases of SAS, the names of these files will change accordingly.

For each system option, SAS uses the last setting that it encounters. Any other settings are ignored. For example, if the WORKPERMS system option is specified in sasv9.cfg

in the !SASROOT directory and in sasv9.cfg in your current directory, SAS uses the value specified in sasv9.cfg in your current directory.

Order of Precedence for Processing SAS System Options

If the same system option appears in more than one place, here is the order of precedence from highest to lowest:

1. restricted options table, if it exists
2. OPTIONS statement
3. autoexec file (that contains an OPTIONS statement)
4. SAS start-up command
5. SASV9_OPTIONS environment variable
6. configuration file specification
7. SAS system default settings.

Interaction with Data Set Options

Many system options and data set options share the same name and have the same function. System options remain in effect for all DATA and PROC steps in a SAS job or session until their settings are changed. A data set option, however, overrides a system option only for the particular data set in the step in which it appears.

In this example, the OBS= system option in the OPTIONS statement specifies that only the first 100 observations are read from any data set within the SAS job. The OBS= data set option in the SET statement, however, overrides the system option and specifies that only the first five observations are read from data set TWO. The PROC PRINT step uses the system option setting and reads and prints the first 100 observations from data set THREE:

```
options obs=100;

data one;
  set two(obs=500);
run;

proc print data=three;
run;
```

Comparisons

Note the differences between system options, data set options, and statement options.

system options

remain in effect for all DATA and PROC steps in a SAS job or current process unless they are respecified.

data set options

apply to the processing of the SAS data set with which they appear. Some data set options have corresponding system options or LIBNAME statement options. For an

individual data set, you can use the data set option to override the setting of these other options.

statement options

control the action of the statement in which they appear. Options in global statements, such as in the LIBNAME statement, can have a broader impact. For example, the LIBNAME= statement options affect all processing that is performed for a particular library.

Part 2

SAS System Options

Chapter 2

Dictionary of SAS System Options 21

Chapter 2

Dictionary of SAS System Options

SAS System Options Documented in Other SAS Publications	24
SAS System Options by Category	24
Dictionary	37
ALIGN SASIOFILES System Option	37
ALTLOG System Option	38
APPEND= System Option	39
AUTOCORRECT System Option	41
AUTOEXEC System Option	43
BUFNO= System Option	45
BUFSIZE= System Option	47
BYERR System Option	48
BYLINE System Option	49
BYSORTED System Option	50
CAPS System Option	51
CARDIMAGE System Option	52
CATCACHE= System Option	53
CBUFNO= System Option	54
CENTER System Option	55
CGOPTIMIZE= System Option	56
CHARCODE System Option	57
CHKPTCLEAN System Option	58
COMPRESS= System Option	59
CONFIG System Option	61
CPUCOUNT= System Option	62
CUID System Option	63
DATAPAGESIZE= System Option	64
DATASTMTCHK= System Option	65
DATE System Option	66
DATESTYLE= System Option	66
DECIMALCONV= System Option	67
DETAILS System Option	68
DKRICOND= System Option	69
DKROCOND= System Option	70
DLCREATEDIR System Option	71
DLDMGACTION= System Option	72
DSACCEL= System Option	73
DSNFERR System Option	74
DTRESET System Option	75
ECHO System Option	75
ECHOAUTO System Option	76

ERRORABEND System Option	77
ERRORBYABEND System Option	78
ERRORCHECK= System Option	79
ERRORS= System Option	80
FILELOCKS System Option	81
FILELOCKWAIT= System Option	83
FILELOCKWAITMAX= System Option	84
FILESYNC= System Option	85
FIRSTOBS= System Option	86
FMterr System Option	88
FMTSEARCH= System Option	88
FORMCHAR= System Option	91
FULLSTIMER System Option	92
HOSTINFOLONG System Option	95
IBUFNO= System Option	95
IBUFSIZE= System Option	97
INITSTMT= System Option	98
INSERT= System Option	99
INTERVALDS= System Option	101
INVALIDDATA= System Option	102
JREOPTIONS System Option	103
LABEL System Option	103
LABELCHKPT System Option	104
LABELCHKPTLIB= System Option	105
LABELRESTART System Option	107
LAST = System Option	108
LINESIZE= System Option	109
LOCKDOWN System Option	110
LOG System Option	111
LOGPARM= System Option	112
LRECL= System Option	117
LPTYPE System Option	118
MAXMEMQUERY System Option	118
MEMSIZE System Option	119
MERGENOBY System Option	122
METACONNECT= System Option	122
METAENCRYPTALG System Option	124
METAENCRYPTLEVEL System Option	125
METAPASS= System Option	125
METAPORT= System Option	127
METAPROFILE System Option	128
METAPROTOCOL= System Option	129
METAREPOSITORY= System Option	130
METASERVER= System Option	131
METASPN= System Option	132
METAUSER= System Option	133
MISSING= System Option	134
MSG System Option	135
MSGCASE System Option	136
MSGLEVEL= System Option	136
MSYMTABMAX System Option	138
MVARSIZE System Option	138
NEWS= System Option	139
NOTES System Option	140
NUMBER System Option	140
OBJECTSERVER System Option	141

OBS= System Option	142
OPLIST System Option	149
OVP System Option	150
PAGENO= System Option	151
PAGESIZE= System Option	152
PARM= System Option	152
PARMCARDS= System Option	153
PRINT System Option	154
PRINTCMD System Option	154
PRINTMSGLIST System Option	155
QUOTELENMAX System Option	156
REALMEMSIZE System Option	157
REPLACE System Option	158
REUSE= System Option	159
RSASUSER System Option	160
RTRACE System Option	161
RTRACELOC System Option	162
S= System Option	163
S2= System Option	165
S2V= System Option	168
SASHELP= System Option	169
SASAUTOS System Option	170
SASUSER= System Option	172
SERVICESBASEURL= System Option	173
SEQ= System Option	173
SET System Option	174
SETINIT System Option	175
SORTANOM System Option	176
SORTCUT System Option	177
SORTCUTP System Option	178
SORTDEV System Option	179
SORTDUP= System Option	179
SORTEQUALS System Option	180
SORTNAME System Option	181
SORTPARAM System Option	182
SORTPGM System Option	182
SORTSIZE= System Option	183
SORTVALIDATE System Option	184
SOURCE System Option	186
SOURCE2 System Option	186
SPOOL System Option	187
STARTLIB System Option	188
STDIO System Option	189
STEPCHKPT System Option	189
STEPCHKPTLIB= System Option	191
STEPRESTART System Option	192
STIMEFMT System Option	193
STIMER System Option	197
STRIPESIZE= System Option	198
SYNTAXCHECK System Option	200
SYSIN System Option	201
TERMINAL System Option	202
TERMSTMT= System Option	203
THREADS System Option	204
TIMEZONE= System Option	205
UBUFNO= System Option	207

UBUFSIZE= System Option	208
USER= System Option	209
USERCONFIG System Option	209
UTILLOC= System Option	210
VALIDFMTNAME= System Option	212
VALIDMEMNAME= System Option	213
VALIDVARNAME= System Option	215
VARINITCHK= System Option	217
VARLENCHK= System Option	218
VBUFSIZE= System Option	221
VERBOSE System Option	223
VNFERR System Option	224
WORK System Option	228
WORKINIT System Option	230
WORKPERMS System Option	231
WORKTERM System Option	231
XCMD System Option	232
YEARCUTOFF= System Option	233

SAS System Options Documented in Other SAS Publications

Some system options are documented with related subject matter in other SAS publications:

- [Cloud Analytic Services](#)
- [Encryption in SAS](#)
- [SAS Macro Language: Reference](#)
- [SAS National Language Support \(NLS\): Reference Guide](#)
- [SAS SQL Procedure User's Guide](#)
- [Universal Printing](#)

SAS System Options by Category

The system options by category represent the system options that appear in this document. The categories for SAS system options correspond to the SAS system option groups and subgroups:

Communications: E-mail	options associated with sending and receiving email using SAS
Communications: Networking and Encryption	options related to remote communication, shared settings, and encryption
Environment Control: Display	options to set SAS windows and display preferences
Environment Control: Error Handling	options associated with error conditions and error messages

Environment Control: Files	options to set SAS library and file location preferences
Environment Control: Help	options used to configure the SAS Help
Environment Control: Initialization and Operation	options that establish the SAS operating environment
Environment Control: Language Control	options to set language and translation preferences
Files: External Files	options that define how to process files that are not created by SAS
Files: SAS Files	options that define how to process SAS files
Input Control: Data Processing	options for data entry and data processing preferences
Input Control: Data Quality	options to configure the SAS Data Quality server
Graphics: Driver Settings	options that define devices, graphics, and map preferences
Log and Procedure Output Control: SAS Log	options that control the display of messages that are written to the SAS log
Log and Procedure Output Control: Procedure Output	options that define procedure output and display preferences
Log and Procedure Output Control: SAS Log and Procedure Output	options that control both SAS log and procedure output preferences
Log and Procedure Output Control: ODS Printing	options to define preferences for printing to ODS destinations
Log and Procedure Output Control: PDF	options to define preferences for PDF files
Log and Procedure Output Control: SVG	options to define preferences for SVG files
Log and Procedure Output Control: Animation	options to define preferences for Animating SVG files
Macro: SAS Macro	options that define SAS macro preferences
Sort: Procedure Options	options that define preferences for sorting SAS files
System Administration: Installation	options for defining site installation settings
System Administration: Memory	options that define computer memory preferences
System Administration: Performance	options that define performance preferences
System Administration: Code Generation	options that define preferences for generating SAS language statements
System Administration: Security	options for defining security settings

System Administration: SQL	options that define settings for the SQL procedure
System Administration: TK	options that are used in threaded processing

Category	Language Elements	Description
Communications: Metadata	METACONNECT= System Option (p. 122)	Identifies one named profile from the metadata connection profiles for connecting to a SAS 9 Metadata Server.
	METAENCRYPTALG System Option (p. 124)	Specifies the type of encryption to use when communicating with the SAS 9 Metadata Server.
	METAENCRYPTLEVEL System Option (p. 125)	Specifies the level of encryption when communicating with the SAS 9 Metadata Server.
	METAPASS= System Option (p. 125)	Specifies the password for the SAS 9 Metadata Server.
	METAPORT= System Option (p. 127)	Specifies the TCP port for the SAS 9 Metadata Server.
	METAPROFILE System Option (p. 128)	Identifies the XML document that contains connection profiles for the SAS 9 Metadata Server.
	METAPROTOCOL= System Option (p. 129)	Specifies the network protocol for connecting to the SAS 9 Metadata Server.
	METAREPOSITORY= System Option (p. 130)	Specifies the SAS 9 Metadata Repository to use with the SAS 9 Metadata Server.
	METASERVER= System Option (p. 131)	Specifies the host name or address of the SAS 9 Metadata Server.
Communications: Networking and Encryption	SERVICESBASEURL= System Option (p. 173)	Specifies the root URL for calls to SAS Viya services.
Environment Control: Display	CHARCODE System Option (p. 57)	Specifies whether specific keyboard combinations are substituted for special characters that are not on the keyboard.
	XCMD System Option (p. 232)	Specifies whether the X command is valid in the SAS session.
Environment Control: Error Handling	AUTOCORRECT System Option (p. 41)	Specifies whether SAS attempts to automatically correct misspelled procedure names, misspelled procedure keywords, or misspelled global statement names.
	BYERR System Option (p. 48)	Specifies whether SAS produces errors when the SORT procedure attempts to process a <code>_NULL_</code> data set.

Category	Language Elements	Description
	CHKPTCLEAN System Option (p. 58)	When SAS is in checkpoint mode or restart mode, specifies whether to erase the contents of the Work library after a batch program executes successfully.
	DKRICOND= System Option (p. 69)	Specifies the level of error detection to report when a variable is missing from an input data set during the processing of a DROP=, KEEP=, or RENAME= data set option.
	DKROCOND= System Option (p. 70)	Specifies the level of error detection to report when a variable is missing for an output data set during the processing of a DROP=, KEEP=, or RENAME= data set option.
	DSNFERR System Option (p. 74)	When a SAS data set cannot be found, specifies whether SAS issues an error message.
	ERRORABEND System Option (p. 77)	Specifies whether SAS responds to errors by terminating.
	ERRORBYABEND System Option (p. 78)	Specifies whether SAS ends a program when an error occurs in BY-group processing.
	ERRORCHECK= System Option (p. 79)	Specifies whether SAS enters syntax-check mode when errors are found in the LIBNAME, FILENAME, and %INCLUDE statements.
	ERRORS= System Option (p. 80)	Specifies the maximum number of observations for which SAS issues complete error messages.
	FMTERR System Option (p. 88)	When a variable format cannot be found, specifies whether SAS generates an error or continues processing.
	LABELCHKPT System Option (p. 104)	Specifies whether checkpoint-restart data for labeled code sections is to be recorded for batch programs.
	LABELCHKPTLIB= System Option (p. 105)	Specifies the libref of the library where the checkpoint-restart data is saved for labeled code sections.
	LABELRESTART System Option (p. 107)	Specifies whether to execute a batch program by using checkpoint-restart data for data collected at labeled code sections.
	QUOTELENMAX System Option (p. 156)	If a quoted string exceeds the maximum length allowed, specifies whether SAS writes a note to the SAS log.
	STEPCHKPT System Option (p. 189)	Specifies whether checkpoint-restart data for DATA and PROC steps is to be recorded for a batch program.
	STEPCHKPTLIB= System Option (p. 191)	Specifies the libref of the library where checkpoint-restart data for DATA and PROC steps is saved.
	STEPRESTART System Option (p. 192)	Specifies whether to execute a batch program by using checkpoint-restart data for DATA and PROC steps.
	SYNTAXCHECK System Option (p. 200)	Specifies whether to enable syntax check mode for multiple steps.

Category	Language Elements	Description
	VNFERR System Option (p. 224)	Specifies whether SAS issues an error or warning when a BY variable exists in one data set but not another data set when the other data set is <code>_NULL_</code> . This option applies when processing the SET, MERGE, UPDATE, or MODIFY statements.
Environment Control: Files	ALTLOG System Option (p. 38)	Specifies a location to write a copy of the SAS log.
	APPEND= System Option (p. 39)	Appends a value to the existing value of the specified system option.
	AUTOEXEC System Option (p. 43)	Specifies the SAS autoexec file.
	CONFIG System Option (p. 61)	Specifies the configuration file that is used when initializing or overriding the default values of SAS system options.
	FMTSEARCH= System Option (p. 88)	Specifies the order in which format catalogs are searched.
	INSERT= System Option (p. 99)	Inserts the specified value as the first value of the specified system option.
	LOG System Option (p. 111)	Specifies a destination for the SAS log when running in batch mode.
	MSG System Option (p. 135)	Specifies the library that contains the SAS error messages.
	NEWS= System Option (p. 139)	Specifies an external file that contains messages to be written to the SAS log, immediately after the header.
	PARM= System Option (p. 152)	Specifies a parameter string that is passed to an external program.
	PARMCARDS= System Option (p. 153)	Specifies the file reference to open when SAS encounters the PARMCARDS statement in a procedure.
	PRINT System Option (p. 154)	Specifies a destination for SAS output when running in batch mode.
	RSASUSER System Option (p. 160)	Specifies whether to open the Sasuser library for Read access or Read-Write access in Batch mode.
	RTRACELOC System Option (p. 162)	Specifies the pathname of the file to which the list of resources that are read or loaded during a SAS session is written.
	SASHELP= System Option (p. 169)	Specifies the location of the Sashelp library.
	SASAUTOS System Option (p. 170)	Specifies the autocall library.
	SASUSER= System Option (p. 172)	Specifies the SAS library to use as the Sasuser library.

Category	Language Elements	Description
	SET System Option (p. 174)	Defines a SAS environment variable.
	SYSIN System Option (p. 201)	Specifies the default location of SAS source code when running in batch mode.
	USER= System Option (p. 209)	Specifies the default permanent SAS library.
	USERCONFIG System Option (p. 209)	Specifies whether SAS processes the sasv9.cfg configuration file in a user's home directory.
	WORK System Option (p. 228)	Specifies the location of the Work library.
	WORKINIT System Option (p. 230)	Specifies whether to initialize the Work library at SAS invocation.
	WORKPERMS System Option (p. 231)	Sets the permissions of the SAS Work library when it is initially created.
	WORKTERM System Option (p. 231)	Specifies whether to erase the Work files when SAS terminates.
Environment Control: Initialization and Operation	INITSTMT= System Option (p. 98)	Specifies a SAS statement to execute after any statements in the AUTOEXEC= file and before any statements from the SYSIN= file.
	JREOPTIONS System Option (p. 103)	Identifies the Java Runtime Environment (JRE) options for SAS.
	LOCKDOWN System Option (p. 110)	Enables the ability to limit access to files and to specific SAS features for a SAS session executing in a batch or the workspace server.
	OBJECTSERVER System Option (p. 141)	Specifies whether SAS is to run as an Integrated Object Model (IOM) server.
	TERMINAL System Option (p. 202)	Specifies whether to associate a terminal with a SAS session.
	TERMSTMT= System Option (p. 203)	Specifies the SAS statements to execute when SAS terminates.
Environment Control: Language Control	DATESTYLE= System Option (p. 66)	Specifies the sequence of month, day, and year when ANYDTDTE, ANYDTCMT, or ANYDTTME informat data is ambiguous.
	DSACCEL= System Option (p. 73)	Specifies whether the DATA step is enabled for parallel processing in CAS.
	TIMEZONE= System Option (p. 205)	Specifies the user local time zone.
Files: External Files	FILELOCKS System Option (p. 81)	Specifies whether file locking is turned on or off and what action should be taken if a file cannot be locked.

Category	Language Elements	Description
	LRECL= System Option (p. 117)	Specifies the default logical record length to use for reading and writing external files.
	STARTLIB System Option (p. 188)	Specifies whether SAS assigns user-defined permanent librefs when SAS starts.
Files: SAS Files	ALIGNSASIOFILES System Option (p. 37)	Aligns output data on a page boundary to improve performance.
	BUFNO= System Option (p. 45)	Specifies the number of buffers to be allocated for processing SAS data sets.
	BUFSIZE= System Option (p. 47)	Specifies the permanent buffer size for output SAS data sets.
	CATCACHE= System Option (p. 53)	Specifies the number of SAS catalogs to keep open in cache memory.
	CBUFNO= System Option (p. 54)	Specifies the number of extra page buffers to allocate for each open SAS catalog.
	COMPRESS= System Option (p. 59)	Specifies the type of compression of observations to use for output SAS data sets.
	DATAPAGESIZE= System Option (p. 64)	Specifies how SAS determines the optimal buffer size for a SAS data set or utility file.
	DATASTMTCHK= System Option (p. 65)	Specifies which SAS statement keywords are prohibited from being specified as a one-level DATA step name to protect against overwriting an input data set.
	DKRCOND= System Option (p. 69)	Specifies the level of error detection to report when a variable is missing from an input data set during the processing of a DROP=, KEEP=, or RENAME= data set option.
	DKROCOND= System Option (p. 70)	Specifies the level of error detection to report when a variable is missing for an output data set during the processing of a DROP=, KEEP=, or RENAME= data set option.
	DLCREATEDIR System Option (p. 71)	Specifies to create a directory for the SAS library that is named in a LIBNAME statement if the directory does not exist.
	DLDMGACTION= System Option (p. 72)	Specifies the type of action to take when a SAS data set or a SAS catalog is detected as damaged.
	FILELOCKS System Option (p. 81)	Specifies whether file locking is turned on or off and what action should be taken if a file cannot be locked.
	FILELOCKWAIT= System Option (p. 83)	Sets the number of seconds that SAS waits for a locked file to become available.
	FILELOCKWAITMAX= System Option (p. 84)	Sets an upper limit on the time SAS waits for a locked file.

Category	Language Elements	Description
	FILESYNC= System Option (p. 85)	Specifies when operating system buffers that contain contents of permanent SAS files are written to disk.
	FIRSTOBS= System Option (p. 86)	Specifies the observation number or external file record that SAS processes first.
	IBUFNO= System Option (p. 95)	Specifies an optional number of extra buffers to be allocated for navigating an index file.
	IBUFSIZE= System Option (p. 97)	Specifies the buffer size for an index file.
	LAST = System Option (p. 108)	Specifies the most recently created data set.
	MERGENOBY System Option (p. 122)	Specifies the type of message that is issued when MERGE processing occurs without an associated BY statement.
	OBS= System Option (p. 142)	Specifies the observation that is used to determine the last observation to process, or specifies the last record to process.
	REPLACE System Option (p. 158)	Specifies whether permanently stored SAS data sets can be replaced.
	REUSE= System Option (p. 159)	Specifies whether SAS reuses space when observations are added to a compressed SAS data set.
	STRIPESIZE= System Option (p. 198)	Specifies one or more directory and size argument pairs that set the size of SAS data sets and utility files in that directory to the size of an I/O device stripe.
	UBUFNO= System Option (p. 207)	Specifies the number of buffers to use for utility files.
	UBUFSIZE= System Option (p. 208)	Specifies the buffer size for utility files.
	UTILLOC= System Option (p. 210)	Specifies one or more file system locations in which enabled threaded applications can store utility files.
	VALIDFMTNAME= System Option (p. 212)	Specifies the maximum size (32 characters or 8 characters) that user-created format and informat names can be before an error or warning is issued.
	VALIDMEMNAME= System Option (p. 213)	Specifies the rules for naming SAS data sets, SAS data views, and item stores.
	VALIDVARNAME= System Option (p. 215)	Specifies the rules for valid SAS variable names that can be created and processed during a SAS session.
	VARINITCHK= System Option (p. 217)	Specifies whether a DATA step stops or continues executing when a variable is not initialized and the type of message to write to the SAS log.

Category	Language Elements	Description
	VARLENCHK= System Option (p. 218)	Specifies the type of message to write to the SAS log when the input data set is read using the SET, MERGE, UPDATE, or MODIFY statements.
Input Control: Data Processing	BYSORTED System Option (p. 50)	Specifies whether observations in one or more data sets are sorted in alphabetic or numeric order or are grouped in another logical order.
	CAPS System Option (p. 51)	Specifies whether to convert certain types of input to uppercase.
	CARDIMAGE System Option (p. 52)	Specifies whether SAS processes source and data lines as 80-byte cards.
	DATESTYLE= System Option (p. 66)	Specifies the sequence of month, day, and year when ANYDTDTE, ANYDTCM, or ANYDTTME informat data is ambiguous.
	INTERVALDS= System Option (p. 101)	Specifies one or more interval name-value pairs, where the value is a SAS data set that contains user-defined intervals. The intervals can be used as arguments to the INTNX and INTCK functions.
	INVALIDDATA= System Option (p. 102)	Specifies the value that SAS assigns to a variable when invalid numeric data is encountered.
	S= System Option (p. 163)	Specifies the length of statements on each line of a source statement and the length of data on lines that follow a DATALINES statement.
	S2= System Option (p. 165)	Specifies the length of statements on each line of a source statement from a %INCLUDE statement, an AUTOEXEC= file, or an autocall macro file.
	S2V= System Option (p. 168)	Specifies the starting position to begin reading a file that is specified in a %INCLUDE statement, an autoexec file, or an autocall macro file with a variable length record format.
	SEQ= System Option (p. 173)	Specifies the length of the numeric portion of the sequence field in input source lines or data lines.
	SPOOL System Option (p. 187)	Specifies whether SAS statements are written to a utility data set in the Work library.
	STDIO System Option (p. 189)	Specifies whether SAS should use stdin, stdout, and stderr.
	VBUFSIZE= System Option (p. 221)	Specifies the size of the view buffer.
	YEARCUTOFF= System Option (p. 233)	Specifies the first year of a 100-year span that is used by date informats and functions to read a two-digit year.
Log and Procedure Output Control: Procedure Output	BYLINE System Option (p. 49)	Specifies whether to print BY lines above each BY group.

Category	Language Elements	Description
	CENTER System Option (p. 55)	Specifies whether to center or left-align SAS procedure output.
	DATE System Option (p. 66)	Specifies whether to print the date and time that a SAS program started.
	DECIMALCONV= System Option (p. 67)	Specifies the binary and decimal conversion methodology for formatting numeric data.
	DETAILS System Option (p. 68)	Specifies whether to include additional information when files are listed in a SAS library.
	DTRESET System Option (p. 75)	Specifies whether to update the date and time in the SAS log and in the procedure output file.
	FORMCHAR= System Option (p. 91)	Specifies the default output formatting characters.
	LABEL System Option (p. 103)	Specifies whether SAS procedures can use labels with variables.
	LINESIZE= System Option (p. 109)	Specifies the line size for the SAS log and for SAS procedure output.
	LPTYPE System Option (p. 118)	Specifies which Linux command and option settings are used to route files to the printer.
	MISSING= System Option (p. 134)	Specifies the character to print for missing numeric values.
	NUMBER System Option (p. 140)	Specifies whether to print the page number in the title line of each page of SAS PDF and RTF output.
	PAGENO= System Option (p. 151)	Resets the SAS output page number.
	PAGESIZE= System Option (p. 152)	Specifies the number of lines that compose a page of the SAS log.
	PRINTCMD System Option (p. 154)	Specifies the print command that SAS uses.
Log and Procedure Output Control: SAS Log	CPUID System Option (p. 63)	Specifies whether the CPU identification number is written to the SAS log.
	DATE System Option (p. 66)	Specifies whether to print the date and time that a SAS program started.
	DECIMALCONV= System Option (p. 67)	Specifies the binary and decimal conversion methodology for formatting numeric data.
	DETAILS System Option (p. 68)	Specifies whether to include additional information when files are listed in a SAS library.

Category	Language Elements	Description
	DTRESET System Option (p. 75)	Specifies whether to update the date and time in the SAS log and in the procedure output file.
	ECHO System Option (p. 75)	Specifies a message to be echoed to stdout.
	ECHOAUTO System Option (p. 76)	Specifies whether the statements in the AUTOEXEC= file are written to the SAS log as they are executed.
	ERRORS= System Option (p. 80)	Specifies the maximum number of observations for which SAS issues complete error messages.
	FULLSTIMER System Option (p. 92)	Specifies whether to write all available system performance statistics and the datetime stamp to the SAS log.
	HOSTINFOLONG System Option (p. 95)	Specifies to print additional operating environment information in the SAS log when SAS starts.
	LINESIZE= System Option (p. 109)	Specifies the line size for the SAS log and for SAS procedure output.
	LOGPARM= System Option (p. 112)	Specifies when SAS log files are opened, closed, and, in conjunction with the LOG= system option, how they are named.
	MISSING= System Option (p. 134)	Specifies the character to print for missing numeric values.
	MSGCASE System Option (p. 136)	Specifies whether notes, warnings, and error messages that are generated by SAS are displayed in uppercase characters.
	MSGLEVEL= System Option (p. 136)	Specifies the level of detail in messages that are written to the SAS log.
	NEWS= System Option (p. 139)	Specifies an external file that contains messages to be written to the SAS log, immediately after the header.
	NOTES System Option (p. 140)	Specifies whether notes are written to the SAS log.
	NUMBER System Option (p. 140)	Specifies whether to print the page number in the title line of each page of SAS PDF and RTF output.
	OPLIST System Option (p. 149)	Specifies whether the settings of the SAS system options are written to the SAS log.
	OVP System Option (p. 150)	Specifies whether overprinting of error messages to make them bold, is enabled.
	PAGESIZE= System Option (p. 152)	Specifies the number of lines that compose a page of the SAS log.
	PRINTMSGLIST System Option (p. 155)	Specifies whether to print all messages to the SAS log or to print only top-level messages to the SAS log.
	RTRACE System Option (p. 161)	Produces a list of resources that are read or loaded during a SAS session.

Category	Language Elements	Description
	SOURCE System Option (p. 186)	Specifies whether SAS writes source statements to the SAS log.
	SOURCE2 System Option (p. 186)	Specifies whether SAS writes secondary source statements from included files to the SAS log.
	STIMEFMT System Option (p. 193)	Specifies the format that is used to display the time on FULLSTIMER and STIMER output.
	STIMER System Option (p. 197)	Specifies whether to write a subset of system performance statistics to the SAS log.
	VERBOSE System Option (p. 223)	Specifies whether SAS writes the system option settings to the SAS log.
Log and Procedure Output Control: SAS Log and Procedure Output	DATE System Option (p. 66)	Specifies whether to print the date and time that a SAS program started.
	DECIMALCONV= System Option (p. 67)	Specifies the binary and decimal conversion methodology for formatting numeric data.
	DETAILS System Option (p. 68)	Specifies whether to include additional information when files are listed in a SAS library.
	DTRESET System Option (p. 75)	Specifies whether to update the date and time in the SAS log and in the procedure output file.
	LINESIZE= System Option (p. 109)	Specifies the line size for the SAS log and for SAS procedure output.
	MISSING= System Option (p. 134)	Specifies the character to print for missing numeric values.
	NUMBER System Option (p. 140)	Specifies whether to print the page number in the title line of each page of SAS PDF and RTF output.
	PAGESIZE= System Option (p. 152)	Specifies the number of lines that compose a page of the SAS log.
Macro: SAS Macro	MSYMTABMAX System Option (p. 138)	Specifies the maximum amount of memory available to the macro variable symbol tables.
	MVARSIZE System Option (p. 138)	Specifies the maximum size for in-memory macro variables.
	SASAUTOS System Option (p. 170)	Specifies the autocall library.
Sort: Procedure Options	SORTANOM System Option (p. 176)	Specifies options for the host sort utility.
	SORTCUT System Option (p. 177)	Specifies the data size in number of observations above which SAS uses the host sort instead of the internal SAS sort.

Category	Language Elements	Description
	SORTCUTP System Option (p. 178)	Specifies the data size in bytes above which SAS uses the host sort instead of the internal SAS sort.
	SORTDEV System Option (p. 179)	Specifies the pathname used for temporary files created by the host sort utility.
	SORTDUP= System Option (p. 179)	Specifies whether the SORT procedure removes duplicate variables based on all variables in a data set or the variables that remain after the DROP or KEEP data set options have been applied.
	SORTEQUALS System Option (p. 180)	Specifies whether observations in the output data set with identical BY variable values are in a particular order.
	SORTNAME System Option (p. 181)	Specifies the name of the host sort utility.
	SORTPARM System Option (p. 182)	Specifies parameters for the host sort utility.
	SORTPGM System Option (p. 182)	Specifies whether to use the internal SAS sort utility or the host sort utility or to let SAS choose which sort utility to use.
	SORTSIZE= System Option (p. 183)	Specifies the amount of memory that is available to the SORT procedure.
	SORTVALIDATE System Option (p. 184)	Specifies whether the SORT procedure verifies if a data set is sorted according to the variables in the BY statement when a user-specified sort order is denoted in the sort indicator.
System Administration: Code Generation	CGOPTIMIZE= System Option (p. 56)	Specifies the level of optimization to perform during code compilation.
System Administration: Installation	SETINIT System Option (p. 175)	Specifies whether site license information can be altered.
System Administration: Memory	MAXMEMQUERY System Option (p. 118)	Specifies the maximum amount of memory that can be allocated per request for certain procedures.
	MEMSIZE System Option (p. 119)	Specifies the limit on the total amount of virtual memory that can be used by a SAS session.
	REALMEMSIZE System Option (p. 157)	Specifies the amount of real (physical) memory SAS can expect to allocate.
	SORTSIZE= System Option (p. 183)	Specifies the amount of memory that is available to the SORT procedure.
System Administration: Performance	BUFNO= System Option (p. 45)	Specifies the number of buffers to be allocated for processing SAS data sets.

Category	Language Elements	Description
	BUFSIZE= System Option (p. 47)	Specifies the permanent buffer size for output SAS data sets.
	CGOPTIMIZE= System Option (p. 56)	Specifies the level of optimization to perform during code compilation.
	COMPRESS= System Option (p. 59)	Specifies the type of compression of observations to use for output SAS data sets.
	CPUCOUNT= System Option (p. 62)	Specifies the number of processors that the thread-enabled applications should assume are available for concurrent processing.
	SORTSIZE= System Option (p. 183)	Specifies the amount of memory that is available to the SORT procedure.
	STRIPESIZE= System Option (p. 198)	Specifies one or more directory and size argument pairs that set the size of SAS data sets and utility files in that directory to the size of an I/O device stripe.
	THREADS System Option (p. 204)	Specifies that SAS uses threaded processing if available.
	UBUFNO= System Option (p. 207)	Specifies the number of buffers to use for utility files.
	UBUFSIZE= System Option (p. 208)	Specifies the buffer size for utility files.
	VBUFSIZE= System Option (p. 221)	Specifies the size of the view buffer.
System Administration: TK	DATAPAGESIZE= System Option (p. 64)	Specifies how SAS determines the optimal buffer size for a SAS data set or utility file.
	STRIPESIZE= System Option (p. 198)	Specifies one or more directory and size argument pairs that set the size of SAS data sets and utility files in that directory to the size of an I/O device stripe.

Dictionary

ALIGNSASIOFILES System Option

Aligns output data on a page boundary to improve performance.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Files: SAS Files

**PROC OPTIONS
GROUP=** SASFILES

Default: The shipped default is ALIGN SASIOFILES.

Note: This option cannot be restricted.

Syntax

[ALIGN SASIOFILES](#) | [NOALIGN SASIOFILES](#)

Syntax Description

ALIGN SASIOFILES

specifies to align output data on a page boundary.

NOALIGN SASIOFILES

specifies to write output data by using standard SAS practices.

Details

A SAS data set consists of a header that is followed by one or more pages of data. Normally, the header is 8K on Linux. The ALIGN SASIOFILES system option forces the header to be the same size as the data pages so that the data pages are aligned to boundaries that allow for more efficient I/O.

Aligning data on a page boundary causes the size of the file to increase, but performance improves because of fewer page accesses.

You can use the BUFSIZE= system option or the BUFSIZE= data set option to set the page size.

See Also

Data Set Options:

- “BUFSIZE= Data Set Option” in *SAS Viya Data Set Options: Reference*

System Options:

- “BUFSIZE= System Option” on page 47

ALTLOG System Option

Specifies a location to write a copy of the SAS log.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Files

**PROC OPTIONS
GROUP=** ENVFILES, LOGCONTROL

Default: None

Note: This option cannot be restricted by a site administrator. For more information, see “[Restricted Options](#)” on page 8.

Syntax

ALTLOG *file-specification* | NOALTLOG

Required Arguments

ALTLOG *file-specification*

specifies the location where a copy of the SAS log is to be written. The *file-specification* argument can be any valid Linux path to a directory, a filename, or an environment variable that is associated with a path. If you specify only the path to a directory, the SAS log is placed in a file in the specified directory. The name of the file is *filename.log*, where *filename* is the name of your SAS job. If you are running SAS interactively and specify only the path to a directory, the log is written to a file named *sas.log* within that path.

NOALTLOG

specifies that the SAS log is not copied.

Details

ALTLOG Basics

The ALTLOG system option specifies a destination to which a copy of the SAS log is written. All messages that are written to the SAS log are also written to the location specified in *file-specification*. You can use this option to capture log output for printing.

Note: You can use the LOG option in the PRINTTO procedure to redirect any portion of the log to an external file. The code for PROC PRINTTO does not appear in the SAS log for the current session, but it does appear in the SAS log that you created with the ALTLOG system option.

Note: When SAS is started with the OBJECTSERVER and NOTERMINAL system options and no log is specified, SAS discards all log and alternate log messages.

Using Directives with ALTLOG

Using directives in the ALTLOG system option enables you to control when log copies are open and closed, and how they are named, based on real-time events such as time, month, and day of week. For a list of directives, see “LOGPARM= System Option” on page 112.

See Also

“PRINTTO Procedure” in *SAS Viya Visual Data Management and Utility Procedures Guide*

APPEND= System Option

Appends a value to the existing value of the specified system option.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Environment Control: Files
PROC OPTIONS GROUP=	ENVFILES

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

APPEND=(*system-option-1*=*argument-1* <*system-option-2*=*argument-2* ...>)

Syntax Description

system-option

can be AUTOEXEC, FMTSEARCH, MSG, SASAUTOS, SASHELP, SASSCRIPT, or SET.

Note Some of these options are available only when SAS starts. These options can be specified in the APPEND= option only when the APPEND= option is specified in a configuration file, the SASV9_OPTIONS environment variable, or a SAS command.

argument

specifies a new value that you want to append to the current value of *system-option*.

argument can be any value that could be specified for *system-option* if *system-option* is set using the OPTIONS statement.

Restriction The arguments for the FMTSEARCH system option apply only to format catalogs. They do not apply to CAS format libraries. For information about CAS format library search order, see the [CAS statement FMTSEARCH= option](#).

Details

If you specify a new value for the AUTOEXEC, FMTSEARCH, MSG, SASAUTOS, SASSCRIPT, or SET system options, the new value replaces the value of the option. Instead of replacing the value, you can use the APPEND= system option to append a new value to the current value of the option.

For a list of system options that the APPEND= system option and the INSERT= system option support, including the system options that can be used when SAS starts, submit the following OPTIONS procedure:

```
proc options listinsertappend;
run;
```

Comparisons

The APPEND= system option adds a new value to the end of the current value of the AUTOEXEC, FMTSEARCH, MSG, SASAUTOS, SASSCRIPT, or SET system options. The INSERT= system option adds a new value as the first value of one of these system options.

Example

The following table shows the results of adding a value to the end of the FMTSEARCH= option value:

Current FMTSEARCH= Value	Value of APPEND= System Option	New FMTSEARCH= Value
(WORK LIBRARY)	(fmtsearch=(abc def))	(WORK LIBRARY ABC DEF)

See Also

- [“Changing an Option Value By Using the INSERT and APPEND System Options” on page 13](#)

System Options:

- [“INSERT= System Option” on page 99](#)

AUTOCORRECT System Option

Specifies whether SAS attempts to automatically correct misspelled procedure names, misspelled procedure keywords, or misspelled global statement names.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Default: The shipped default is AUTOCORRECT.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

AUTOCORRECT | **NOAUTOCORRECT**

Syntax Description

AUTOCORRECT

specifies that SAS attempts to automatically correct misspelled procedure names, misspelled procedure keywords, or misspelled global statement names.

NOAUTOCORRECT

specifies that SAS does not automatically attempt to correct misspelled procedure names, misspelled procedure keywords, or misspelled global statement names.

Details

In previous releases of SAS, SAS always attempted to correct misspellings. The AUTOCORRECT option enables you to turn off autocorrection.

When AUTOCORRECT is set and a procedure name, a procedure keyword, or a global statement name is misspelled in a SAS program, SAS attempts to interpret the misspelling when a program is compiled. If the attempt succeeds, SAS corrects the error,

prints a warning message to the log, and continues processing. If the error cannot be corrected, SAS writes an error message to the log.

When NOAUTOCORRECT is set, SAS writes the misspelling notification to the SAS log and ends the program.

Example

The following example shows a misspelled global statement name, a misspelled procedure option name, and a misspelled procedure name.

```
/* AUTOCORRECT is the default value */
options autocorrect;
data numbers;
  input x y z;
  datalines;
14.2  25.2  96.8
10.8  51.6  96.8
33.5  27.4  66.5
run;

optionss obs=1;

proc print ddata=numbers;
run;

options noautocorrect;

proc prints ddata=numbers;
run;
```



```

6  options autocorrect;
7  data numbers;
8      input x y z;
9      datalines;
NOTE: The data set WORK.NUMBERS has 3 observations and 3 variables.
NOTE: DATA statement used (Total process time):
      real time      2.75 seconds
      cpu time       0.64 seconds

13  run;
14
15  options obs=1;
      -----
      14
WARNING 14-169: Assuming the symbol OPTIONS was misspelled as options.

16
17  proc print ddata=numbers;
      -----
      1
WARNING 1-322: Assuming the symbol DATA was misspelled as ddata.

18  run;
NOTE: There were 1 observations read from the data set WORK.NUMBERS.
NOTE: PROCEDURE PRINT used (Total process time):
      real time      3.84 seconds
      cpu time       1.07 seconds

19
20  options noautocorrect;
21
22  proc prints ddata=numbers;
      -----
      181
ERROR 181-322: Procedure name misspelled.

23  run;
NOTE: The SAS System stopped processing this step because of errors.

```

AUTOEXEC System Option

Specifies the SAS autoexec file.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Files

PROC OPTIONS GROUP= ENVFILES

Default: autoexec.sas (see “Details” on page 44)

Syntax

-AUTOEXEC *file-specification* | **-NOAUTOEXEC**

-AUTOEXEC \ (*file-specification-1* <...*file-specification-n*>)

Syntax Description

AUTOEXEC *file-specification*

specifies the SAS autoexec file to be used instead of the default autoexec.sas file.

The *file-specification* argument can be a valid Linux filename or an environment variable that is associated with a pathname. If you specify multiple *file-specification* values, SAS executes all autoexec files.

-NOAUTOEXEC

specifies that SAS is not to process any autoexec files.

Details

The Autoexec File

The autoexec file contains SAS statements that are executed automatically when you invoke SAS or when you start another SAS process. The autoexec file can contain any SAS statements. For example, your autoexec file can contain LIBNAME statements for SAS libraries that you access routinely in SAS sessions. You can override the default autoexec file using the AUTOEXEC system option.

SAS looks for the AUTOEXEC system option in the following order. It uses the first AUTOEXEC system option that it finds:

1. in the command line
2. in the SASV9_OPTIONS environment variable
3. in the configuration file

SAS uses the first AUTOEXEC option that it encounters and ignores all others.

If neither the AUTOEXEC nor NOAUTOEXEC system option is found, SAS looks for the autoexec file in three directories in the following order:

1. your current directory
2. your home directory
3. the !SASROOT directory

SAS uses the first autoexec file that it finds to initialize the SAS session.

If you want to see the contents of the autoexec file for your session, use the ECHOAUTO system option when you invoke SAS. If you want to identify the data sources that the autoexec file is using, use the PROC OPTIONS statement:

```
proc options option=autoexec value;
run;
```

Adding Files to the Autoexec Option Value

You can use the INSERT system option to add an autoexec file as the first value of the AUTOEXEC system option. To add a file as the last value of the AUTOEXEC system option, use the APPEND system option. When you use the INSERT or APPEND options with the AUTOEXEC option, SAS concatenates the files and executes all files. For more information, see [“INSERT= System Option” on page 99](#) and [“APPEND= System Option” on page 39](#).

The autoexec file is always a Linux file. If your filename contains embedded blanks or special characters, you must enclose the filename in quotation marks. Otherwise, quotation marks are optional when one or more filenames are specified.

You can use the following syntax to create values for the AUTOEXEC system option::

```
-autoexec "(/path1/autoexec.sas /path2/autoexec.sas /path3/autoexec.sas) "
```

You can use the following syntax with the INSERT system option:

```
-insert autoexec "a.sas" -insert autoexec "b.sas"
```

You can use the following syntax with the APPEND system option:

```
-append autoexec "a.sas" -append autoexec "b.sas"
```

See Also

System Options:

- [“ECHOAUTO System Option” on page 76](#)

BUFNO= System Option

Specifies the number of buffers to be allocated for processing SAS data sets.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Files: SAS Files System Administration: Performance
PROC OPTIONS GROUP=	SASFILES PERFORMANCE
Default:	The shipped default is 1.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

```
BUFNO=n | nK | nM | nG | nT | hexX | MIN | MAX
```

Syntax Description

n | *nK* | *nM* | *nG* | *nT*

specifies the number of buffers to be allocated in multiples of 1: 1,024 (kilo); 1,048,576 (mega); 1,073,741,824 (giga); 1,099,511,627,776 (tera). For example, a value of **8** specifies 8 buffers, and a value of **3m** specifies 3,145,728 buffers.

Tip Use the notation that best fits the memory size of your system.

hexX

specifies the number of buffers as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **2dx** specifies 45 buffers.

MIN

sets the minimum number of buffers to 0, which causes SAS to use the minimum optimal value for the operating environment.

MAX

sets the number of buffers to the maximum possible number in your operating environment, up to the largest four-byte, signed integer, which is $2^{31}-1$, or approximately 2 billion.

Details

The number of buffers is not a permanent attribute of the data set; it is valid only for the current SAS session or job.

BUFNO= applies to SAS data sets that are opened for input, output, or update.

Using BUFNO= can improve execution time by limiting the number of input/output operations that are required for a particular SAS data set. However, the improvement in execution time comes at the expense of increased memory consumption.

The optimal value for buffer system options is dependent on your operating environment. Experiment with various buffers sizes to determine the optimal value for these system options.

You can estimate the number of buffers that you need from the data set page size and the amount of memory in your system. The data set page size can be specified by the BUFSIZE= system option or the BUFSIZE= data set option. If the default is used, SAS uses the minimal optimal page size for the operating environment. You can find the page size for a data set in the output of the CONTENTS procedure. Once you have the data set page size and the amount of memory available, you can estimate the number of buffers that you need. If the number of buffers is too large, SAS might not have enough memory to process the DATA or PROC step. You can change the page size for a data set by re-creating the data set by using the BUFSIZE= data set option.

Comparisons

- You can override the BUFNO= system option by using the BUFNO= data set option.
- SAS can allocate the number of buffers based on the number of data set pages and index file pages. For more information, see [“SASFILE Statement” in SAS Viya Statements: Reference](#).

See Also**Data Set Options:**

- [“BUFNO= Data Set Option” in SAS Viya Data Set Options: Reference](#)

Procedures:

- [“CONTENTS Procedure” in SAS Viya Visual Data Management and Utility Procedures Guide](#)

System Options:

- [“BUFSIZE= System Option” on page 47](#)
- [“UBUFNO= System Option” on page 207](#)
- [“DATAPAGESIZE= System Option” on page 64](#)

BUFSIZE= System Option

Specifies the permanent buffer size for output SAS data sets.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Files: SAS Files System Administration: Performance
PROC OPTIONS GROUP=	SASFILES PERFORMANCE
Default:	The shipped default is 0.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

BUFSIZE=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | **MAX**

Note: You can also use the KB, MB, GB, and TB syntax notations.

Syntax Description

n* | *nK* | *nM* | *nG* | *nT

specifies the buffer size in multiples of 1 (bytes); 1,024 (kilo); 1,048,576 (mega); 1,073,741,824 (giga); 1,099,511,627,776 (tera). You can specify decimal values for the number kilobytes, megabytes, or gigabytes. For example, a value of **8** specifies 8 bytes, a value of **.782k** specifies 801 bytes, and a value of **3m** specifies 3,145,728 bytes.

Note: If the system option and the data set option are not set, the default is 0. This causes SAS to use the minimum optimal buffer size for the operating environment. The BUFSIZE= system option is used in either of the following scenarios:

- if the BUFSIZE= data set option is not set
- if the BUFSIZE= data set option is set to zero

Range 1K–2G-1

Tip Use BUFSIZE=0 in order to reset the buffer size to the default value in your operating environment.

hexX

specifies the buffer size as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by hexadecimal characters (0–9, A–F), and then followed by an X. For example, the value **2dx** sets the page size to 45 bytes.

MAX

sets the buffer page size to 2,147,483,647.

Details

The buffer size is the amount of data that can be transferred from a single input/output operation to one buffer. The buffer size is a permanent attribute of the data set and is used when the data set is processed.

A larger buffer size can improve execution time by reducing the number of times SAS has to read from or write to the storage medium. However, the improvement in execution time comes at the expense of increased memory consumption.

The optimal value for buffer system options is dependent on your operating environment. Experiment with various buffers sizes to determine the optimal value for these system options.

If you specify a nonzero value when you create a SAS data set, the Base engine uses that value. If that value cannot hold at least one observation or is not a multiple of 1K, the engine rounds the value up to a multiple of 1K.

To change the buffer size, use a DATA step to copy the data set and either specify a new buffer or use the SAS default.

Note: If you use the COPY procedure to copy a data set to another library that is allocated with a different engine, the specified buffer size of the data set is not retained.

The buffer size of the utility files that SAS creates to process data can be set by using the UBUFSIZE= system option.

Comparisons

The BUFSIZE= system option can be overridden by the BUFSIZE= data set option.

The DATAPAGESIZE= system option specifies how SAS determines the optimal buffer size for a SAS data set.

See Also

Data Set Options:

- “BUFSIZE= Data Set Option” in *SAS Viya Data Set Options: Reference*

System Options:

- “ALIGN SASIOFILES System Option” on page 37
- “BUFNO= System Option” on page 45
- “DATAPAGESIZE= System Option” on page 64

BYERR System Option

Specifies whether SAS produces errors when the SORT procedure attempts to process a `_NULL_` data set.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

**PROC OPTIONS
GROUP=** ERRORHANDLING

Default: The shipped default is BYERR.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

BYERR | **NOBYERR**

Syntax Description

BYERR

specifies that SAS issue an error message and stop processing if the SORT procedure attempts to sort a `_NULL_` data set.

NOBYERR

specifies that SAS ignore the error message and continue processing if the SORT procedure attempts to sort a `_NULL_` data.

Details

The VNFERR system option sets the error flag for a missing variable when a `_NULL_` data set is used. The DSNFERR system option specifies how SAS responds when a SAS data set is not found.

See Also

System Options:

- [“DSNFERR System Option” on page 74](#)
- [“VNFERR System Option” on page 224](#)

BYLINE System Option

Specifies whether to print BY lines above each BY group.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: Procedure Output

**PROC OPTIONS
GROUP=** LISTCONTROL

Default: The shipped default is BYLINE.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

BYLINE | **NOBYLINE**

Syntax Description

BYLINE

specifies that BY lines are printed above each BY group.

NOBYLINE

suppresses the automatic printing of BY lines.

Details

Use NOBYLINE to suppress the automatic printing of BY lines in procedure output. You can then use #BYVAL, #BYVAR, or #BYLINE to display BYLINE information in a TITLE statement.

PROC PRINT performs its own BY line processing by displaying output for multiple BY groups on the same page.

NOBYLINE causes a page eject between BY groups. For PROC PRINT, the page eject between BY groups has the same effect as specifying the rightmost BY variable in the PAGEBY statement.

See Also

Statements:

- #BYVAL, #BYVAR, and #BYLINE arguments, “TITLE Statement” in *SAS Viya Statements: Reference*

BYSORTED System Option

Specifies whether observations in one or more data sets are sorted in alphabetic or numeric order or are grouped in another logical order.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Input Control: Data Processing

PROC OPTIONS GROUP= INPUTCONTROL

Default: The shipped default is BYSORTED.

Note: This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

BYSORTED | **NOBYSORTED**

Syntax Description

BYSORTED

specifies that observations in a data set or data sets are sorted in alphabetic or numeric order.

Requirement	When you use the BYSORTED option, observations must be ordered or indexed according to the values of BY variables.
Interaction	If both the BYSORTED system option and the NOTSORTED statement option in a BY statement are specified, then the NOTSORTED option in the BY statement takes precedence over the BYSORTED system option.
Tip	If BYSORTED is specified, then SAS assumes that the data set is ordered by the BY variable. BYSORTED should be used if the data set is ordered by the BY variable for better performance.

NOBYSORTED

specifies that observations with the same BY value are grouped together but are not necessarily sorted in alphabetic or numeric order.

Note If a procedure ignores the NOTSORTED option in a BY statement, then it ignores the NOBYSORTED system option also.

Tips When the NOBYSORTED option is specified, you do not have to specify NOTSORTED in every BY statement to access the data sets.

NOBYSORTED is useful if you have data that falls into other logical groupings such as chronological order or linguistic order. NOBYSORTED allows BY processing to continue without failure when a data set is not actually sorted in alphabetic or numeric order.

Details

The requirement for ordering or indexing observations according to the values of BY variables is suspended for BY-group processing when you use the NOBYSORTED option. By default, BY-group processing requires that your data be sorted in alphabetic or numeric order. If your data is grouped in any other way but alphabetic or numeric, then you must use the NOBYSORTED option to allow BY-processing to continue without failure.

See Also**Statements:**

- NOTSORTED option, “NOTSORTED” in *SAS Viya Statements: Reference*

CAPS System Option

Specifies whether to convert certain types of input to uppercase.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Input Control: Data Processing
PROC OPTIONS GROUP=	INPUTCONTROL
Default:	The shipped default is NOCAPS.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[CAPS](#) | [NOCAPS](#)

Syntax Description

CAPS

specifies that SAS translate lowercase characters to uppercase in these types of input:

- data following CARDS, CARDS4, DATALINES, DATALINES4, and PARMCARDS statements
- text enclosed in single or double quotation marks
- values in VALUE and INVALUE statements in the FORMAT procedure
- titles, footnotes, variable labels, and data set labels
- constant text in macro definitions
- values of macro variables
- parameter values passed to macros.

Note Data read from external files and SAS data sets are not translated to uppercase.

NOCAPS

specifies that lowercase characters that occur in the types of input that are listed above are not translated to uppercase.

CARDIMAGE System Option

Specifies whether SAS processes source and data lines as 80-byte cards.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Input Control: Data Processing

PROC OPTIONS GROUP= INPUTCONTROL

Default: The shipped default is NOCARDIMAGE.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[CARDIMAGE](#) | [NOCARDIMAGE](#)

Syntax Description

CARDIMAGE

specifies that SAS source and data lines be processed as if they were punched card images—all exactly 80 bytes long and padded with blanks. That is, column 1 of a line is treated as if it immediately followed column 80 of the previous line.

Therefore, *tokens* can be split across lines. (A *token* is a character or series of characters that SAS treats as a discrete word.)

Strings in quotation marks (literal tokens) that begin on one line and end on another are treated as if they contained blanks out to column 80 of the first line. Data lines longer than 80 bytes are split into two or more 80-byte lines. Data lines are not truncated regardless of their length.

NOCARDIMAGE

specifies that SAS source and data lines not be treated as if they were 80-byte card images. When NOCARDIMAGE is in effect, the end of a line is always treated as the end of the last token, except for strings in quotation marks. Strings in quotation marks can be split across lines. Other types of tokens cannot be split across lines under any circumstances. Strings in quotation marks that are split across lines are not padded with blanks.

Example

Consider the following DATA step:

```
data;
  x='A
  B';
run;
```

If CARDIMAGE is in effect, the variable X receives a value that consists of 78 characters: the A, 76 blanks, and the B. If NOCARDIMAGE is in effect, the variable X receives a value that consists of two characters: AB, with no intervening blanks.

CATCACHE= System Option

Specifies the number of SAS catalogs to keep open in cache memory.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is 0.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

CATCACHE=*n* | *n*K | MIN | MAX

Syntax Description***n* | *nK***

specifies the number of open-file descriptors to keep in cache memory in multiples of 1 (*n*) or 1,024 (*nK*). You can specify decimal values for the number of kilobytes. For example, a value of 8 specifies 8 open-file descriptors, a value of .782k specifies 801 open-file descriptors, and a value of 3k specifies 3,072 open-file descriptors.

If *n* > 0, SAS places up to that number of open-file descriptors in cache memory instead of closing the catalogs.

MIN

sets the number of open-file descriptors that are kept in cache memory to 0.

MAX

sets the number of open-file descriptors that are kept in cache memory to 32,767.

Tip The recommended maximum setting for this option is 10.

Details

Use the CATCACHE= system option to tune an application by avoiding the overhead of repeatedly opening and closing the same SAS catalogs.

CAUTION:

When using both the CBUFNO= and CATCACHE= options, if one of the option's value is set higher than zero, the other option must be set to zero.

See Also**System Options:**

- [“CBUFNO= System Option” on page 54](#)

CBUFNO= System Option

Specifies the number of extra page buffers to allocate for each open SAS catalog.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Files: SAS Files

PROC OPTIONS GROUP= SASFILES

Default: The shipped default is 0.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

CBUFNO=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Syntax Description

n* | *nK* | *nM* | *nG* | *nT

specifies the number of extra page buffers in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). For example, a value of **8** specifies 8 bytes, and a value of **3m** specifies 3,145,728 bytes.

MIN

sets the number of extra page buffers to 0.

MAX

sets the number of extra page buffers to 20.

hexX

specifies the number of extra page buffers as a hexadecimal number. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **0ax** sets the number of extra page buffers to 10 buffers.

Details

The CBUFNO= option is similar to the BUFNO= option that is used for SAS data set processing.

The optimal value for buffer system options is dependent on your operating environment. Experiment with various buffers sizes to determine the optimal value for these system options.

Increasing the value for the CBUFNO= option might result in fewer I/O operations when your application reads very large objects from catalogs. Increasing this value also comes with the normal tradeoff between performance and memory usage. If memory is a serious constraint for your system, you should not increase the value of the CBUFNO= option. Do not increase the value of the CBUFNO= option if you have increased the value of the CATCACHE= option.

CAUTION:

When using both the CBUFNO= and CATCACHE= options, if one of the option's value is set higher than zero, the other option must be set to zero.

CENTER System Option

Specifies whether to center or left-align SAS procedure output.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: Procedure Output

PROC OPTIONS GROUP= LISTCONTROL

Alias: CENTRE

Default: The shipped default is CENTER.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

CENTER | **NOCENTER**

Syntax Description

CENTER

centers SAS procedure output.

NOCENTER

left aligns SAS procedure output.

CGOPTIMIZE= System Option

Specifies the level of optimization to perform during code compilation.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Categories: System Administration: Performance
System Administration: Code Generation

PROC OPTIONS GROUP= PERFORMANCE
CODEGEN

Alias: CGOPT

Default: The shipped default is 3.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

CGOPTIMIZE=0 | **1** | **2** | **3**

Syntax Description

0

specifies not to perform optimization.

1

specifies to perform stage 1 optimization. Stage 1 optimization removes redundant instructions, missing value checks, and repetitive computations for array subscriptions; detects patterns of instructions and replaces them with more efficient sequences.

2

specifies to perform stage 2 optimization. Stage 2 performs optimizations that pertain to the SAS register.

Interaction Stage 2 optimization for a large DATA step program can result in a significant increase in compilation time and thus overall execution time.

3

specifies to perform full optimization, which is a combination of stages 1 and 2.

CHARCODE System Option

Specifies whether specific keyboard combinations are substituted for special characters that are not on the keyboard.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Environment Control: Display
PROC OPTIONS GROUP=	ENVDISPLAY
Default:	The shipped default is NOCHARCODE.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

CHARCODE | **NOCHARCODE**

Syntax Description

CHARCODE

allows certain character combinations to be substituted for special characters that might not be on your keyboard.

NOCHARCODE

does not allow substitutions for certain keyboard characters.

Details

If you do not have the following symbols on your keyboard, you can use these character combinations to create the symbols that you need when CHARCODE is active:

Symbol	Characters
back quotation mark (`)	?:
backslash (\)	?,
left brace ({)	?{(
right brace (})	?)
logical not sign (¬ or ^)	?=
left square bracket ([)	?<
right square bracket (])	?>
underscore (_)	?-

Symbol	Characters
vertical bar ()	?/

Example

This statement produces the output [TEST TITLE]:

```
title '?<TEST TITLE?>';
```

CHKPTCLEAN System Option

When SAS is in checkpoint mode or restart mode, specifies whether to erase the contents of the Work library after a batch program executes successfully.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Error Handling
PROC OPTIONS GROUP=	ERRORHANDLING
Default:	The shipped default is NOCHKPTCLEAN.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

CHKPTCLEAN | **NOCHKPTCLEAN**

Syntax Description

CHKPTCLEAN

specifies that the files in the Work library are to be erased after a batch program successfully executes in checkpoint mode or restart mode.

NOCHKPTCLEAN

specifies that the files in the Work library are not to be erased after a batch program successfully executes in checkpoint mode or restart mode.

Details

Typically, checkpoint mode or restart mode is started with the NOWORKTERM and NOWORKINIT system options set. When these options are set, the Work library is preserved between SAS sessions. You can use the CHKPTCLEAN system option to erase all files from the Work library if you no longer need the files after your batch program successfully runs in checkpoint mode or restart mode.

This option is effective only when the following conditions are met:

- SAS is in either checkpoint mode or restart mode. SAS enters checkpoint mode when the STEPCHKPT option or the LABELCHKPT option is set. SAS enters restart mode when the STEPRESTART option or the LABELRESTART option is set.

- The checkpoint library is Work.
- The program runs successfully in batch mode.

If the program does not run successfully, the files in the Work library are not erased, regardless of whether the CHKPTCLEAN option is set.

Comparisons

The CHKPTCLEAN option erases the contents of the Work library after the successful completion of a batch program only in checkpoint mode or restart mode.

The WORKTERM option erases the contents of the Work library when a SAS session ends.

See Also

System Options:

- [“LABELCHKPT System Option” on page 104](#)
- [“LABELRESTART System Option” on page 107](#)
- [“STEPCHKPT System Option” on page 189](#)
- [“STEPRESTART System Option” on page 192](#)
- [“WORKTERM System Option” on page 231](#)

COMPRESS= System Option

Specifies the type of compression of observations to use for output SAS data sets.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Categories: Files: SAS Files
System Administration: Performance

PROC OPTIONS GROUP= SASFILES
PERFORMANCE

Default: The shipped default is NO.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

COMPRESS=NO | YES | CHAR | BINARY

Syntax Description

NO
specifies that the observations in a newly created SAS data set are uncompressed (maintaining fixed-length records).

YES | CHAR

specifies that the observations in a newly created SAS data set are compressed (producing variable-length records) by using RLE (Run Length Encoding). RLE compresses observations by reducing repeated runs of the same character (including blanks) to two-byte or three-byte representations.

Alias ON

Tip Use this compression algorithm for character data.

BINARY

specifies that the observations in a newly created SAS data set are compressed (producing variable-length records) by using RDC (Ross Data Compression). RDC combines run-length encoding and sliding-window compression to compress the file by representing repeated byte patterns more efficiently.

Tip This method is highly effective for compressing medium to large (several hundred bytes or larger) blocks of binary data (character and numeric variables). Because the compression function operates on a single record at a time, the record length needs to be several hundred bytes or larger for effective compression.

Details

Compressing a file is a process that reduces the number of bytes required to represent each observation. Advantages of compressing a file include reduced storage requirements for the file and fewer I/O operations necessary to read or write to the data during processing. However, more CPU resources are required to read a compressed file (because of the overhead of uncompressing each observation), and there are situations when the resulting file size might increase rather than decrease.

Use the COMPRESS= system option to compress all output data sets that are created during a SAS session. Use the option only when you are creating SAS data files (member type DATA). You cannot compress SAS views, because they contain no data.

Once a file is compressed, the setting is a permanent attribute of the file, which means that to change the setting, you must re-create the file. That is, to uncompress a file, specify COMPRESS=NO for a DATA step that copies the compressed file.

Note: For the COPY procedure, the default value CLONE uses the compression attribute from the input data set for the output data set. If the engine for the input data set does not support the compression attribute, then PROC COPY uses the current value of the COMPRESS= system option. For more information about CLONE and NOCLONE, see the COPY Statement options, “[CLONE | NOCLONE](#)” in *SAS Viya Visual Data Management and Utility Procedures Guide*.

In general, COMPRESS=CHAR provides good compression when single bytes repeat; COMPRESS=BINARY provides good compression when strings of bytes repeat. However, it is more costly to look for strings of bytes that repeat than to look for single bytes that repeat. For examples, see “[Compress=CHAR](#)” in *SAS Viya Data Set Options: Reference* and “[COMPRESS=BINARY](#)” in *SAS Viya Data Set Options: Reference*.

Comparisons

The COMPRESS= system option can be overridden by the COMPRESS= option in the LIBNAME statement and the COMPRESS= data set option.

The data set option POINTOBS=YES, which is the default, determines that a compressed data set can be processed with random access (by observation number)

rather than sequential access. With random access, you can specify an observation number in the POINT= option in the SET and MODIFY statements.

When you create a compressed file, you can also specify REUSE=YES (as a data set option or system option) in order to track and reuse space. With REUSE=YES, new observations are inserted in space freed when other observations are updated or deleted. When the default REUSE=NO is in effect, new observations are appended to the existing file.

POINTOBS=YES and REUSE=YES are mutually exclusive. That is, they cannot be used together. REUSE=YES takes precedence over POINTOBS=YES. That is, if you set REUSE=YES, SAS automatically sets POINTOBS=NO.

See Also

Data Set Options:

- “COMPRESS= Data Set Option” in *SAS Viya Data Set Options: Reference*
- “POINTOBS= Data Set Option” in *SAS Viya Data Set Options: Reference*
- “REUSE= Data Set Option” in *SAS Viya Data Set Options: Reference*

Statements:

- “LIBNAME Statement” in *SAS Viya Statements: Reference*

System Options:

- “REUSE= System Option” on page 159

CONFIG System Option

Specifies the configuration file that is used when initializing or overriding the default values of SAS system options.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Files
PROC OPTIONS GROUP=	ENVFILES
Default:	sasv9.cfg (see “Order of Precedence for Processing SAS Configuration Files” in <i>Batch and Line Mode Processing in SAS Viya</i>)
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

CONFIG *file-specification* | NOCONFIG

Required Arguments

CONFIG *file-specification*

specifies a configuration file to be read. The *file-specification* must resolve to a valid operating system filename.

NOCONFIG

specifies that any previous CONFIG specification should be ignored and that the default system options should be used.

Details

Configuration files contain system option specifications that execute automatically whenever SAS is invoked.

Specifying a configuration file disables the default configuration file list.

See Also

[“Using SAS System Options” on page 4](#)

CPUCOUNT= System Option

Specifies the number of processors that the thread-enabled applications should assume are available for concurrent processing.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: System Administration: Performance

PROC OPTIONS GROUP= PERFORMANCE

Default: If a system has four or more processors, the default value is 4. If a system has fewer than four processors, the default is ACTUAL.

Interaction: If the THREADS system option is set to NOTTHREADS, the CPUCOUNT= option has no effect.

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

CPUCOUNT= 1 - 1024 | ACTUAL

Syntax Description**1-1024**

is the number of CPUs that SAS assumes are available for use by thread-enabled applications.

Tips The value is typically set to the actual number of CPUs available to the current process by your configuration.

Setting CPUCOUNT= to a number greater than the actual number of available CPUs might result in reduced overall performance of SAS.

ACTUAL

returns the number of physical processors that are associated with the operating system where SAS is executing. If the operating system is executing in a partition, the value of the CPUCOUNT= option is the number of physical processors that are associated with the operating system in that partition.

Tips This number can be less than the physical number of CPUs if the SAS process has been restricted by system administration tools.

Setting CPUCOUNT= to ACTUAL at any time causes the option to be reset to the number of physical processors that are associated with the operating system at that time. If the operating system is executing in a partition, the value of the CPUCOUNT= option is the number of physical processors that are associated with the operating system in that partition.

If your system supports Simultaneous Multi-Threading (SMT), hyperthreading, or Chip Multi-Threading (CMT), the value of the CPUCOUNT= option represents the number of such threads on the system.

Details

Certain procedures have been modified to take advantage of multiple CPUs by threading the procedure processing. The Base SAS engine also uses threading to create an index. The CPUCOUNT= option provides the information that is needed to make decisions about the allocation of threads.

Changing the value of CPUCOUNT= affects the degree of parallelism each thread-enabled process attempts to achieve. Setting CPUCOUNT= to a number greater than the actual number of available CPUs might result in reduced overall performance of SAS.

Comparisons

When the related system option THREADS is in effect, threading is active where it is available. The value of the CPUCOUNT= option affects the performance of THREADS by suggesting how many system CPUs are available for use by thread-enabled SAS procedures.

See Also

System Options:

- [“THREADS System Option” on page 204](#)
- [“UTILLOC= System Option” on page 210](#)

CPUID System Option

Specifies whether the CPU identification number is written to the SAS log.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: SAS Log

PROC OPTIONS GROUP= LOGCONTROL

Default: The shipped default is CPUID.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

CPUID | **NOCPUID**

Syntax Description

CPUID

specifies that the CPU identification number is printed at the top of the SAS log after the licensing information.

NOCPUID

specifies that the CPU identification number is not written to the SAS log.

DATAPAGESIZE= System Option

Specifies how SAS determines the optimal buffer size for a SAS data set or utility file.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Categories: Files: SAS Files
System Administration: TK

PROC OPTIONS GROUP= SASFILES
TK

Default: The shipped default is CURRENT.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

DATAPAGESIZE=[COMPAT93](#) | [CURRENT](#)

Syntax Description

COMPAT93

specifies that SAS 9.3 optimization processes are used to determine the buffer size for SAS data sets or utility files.

CURRENT

specifies that the optimization processes for the current release of SAS are used to determine the buffer size for SAS data sets or utility files.

Details

When the BUFSIZE= or UBUFSIZE= system options are set to 0, SAS uses the optimal buffer size for an operating environment. The increase in buffer size might increase the size of the data set or utility file. If the current optimization processes are not ideal for your SAS session, use DATAPAGESIZE=COMPAT93 for the optimization processes that were used prior to SAS 9.4.

See Also

System Options:

- “BUFSIZE= System Option” on page 47
- “UBUFSIZE= System Option” on page 208

DATASTMTCHK= System Option

Specifies which SAS statement keywords are prohibited from being specified as a one-level DATA step name to protect against overwriting an input data set.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Files: SAS Files

PROC OPTIONS GROUP= SASFILES

Default: The shipped default is COREKEYWORDS.

Note: This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

DATASTMTCHK=COREKEYWORDS | ALLKEYWORDS | NONE

Syntax Description

COREKEYWORDS

prohibits certain words as one-level SAS data set names in the DATA statement. They can appear as two-level names. The following keywords cannot appear as one-level SAS data set names:

- MERGE
- RETAIN
- SET
- UPDATE.

For example, SET is not acceptable in the DATA statement, but SAVE.SET and WORK.SET are acceptable.

ALLKEYWORDS

prohibits any keyword that can begin a statement in the DATA step (for example, ABORT, ARRAY, INFILE) as a one-level data set name in the DATA statement.

NONE

provides no protection against overwriting SAS data sets.

Details

If you omit a semicolon in the DATA statement, you can overwrite an input data set if the next statement is SET, MERGE, or UPDATE. Different, but significant, problems arise when the next statement is RETAIN. DATASTMTCHK= enables you to protect yourself against overwriting the input data set.

DATE System Option

Specifies whether to print the date and time that a SAS program started.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Log and Procedure Output Control: SAS Log and Procedure Output Log and Procedure Output Control: SAS Log Log and Procedure Output Control: Procedure Output
PROC OPTIONS GROUP=	LOG_LISTCONTROL LISTCONTROL LOGCONTROL
Default:	The shipped default is DATE.
Interaction:	SAS Studio sets this option to DATE before each code submission. For more information, see “System Options in SAS Studio” on page 7 .
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

DATE | **NODATE**

Syntax Description

DATE

specifies that the date and the time at which the SAS program started are printed at the top of each page of the SAS log and any output that is created by SAS.

NODATE

specifies that the date and the time are not printed.

DATESTYLE= System Option

Specifies the sequence of month, day, and year when ANYDTE, ANYDTM, or ANYDTTME informat data is ambiguous.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Environment Control: Language Control Input Control: Data Processing
PROC OPTIONS GROUP=	INPUTCONTROL LANGUAGECONTROL
Default:	The default value is determined by the value of the LOCALE= system option.
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

DATESTYLE= MDY | YMD | DMY | LOCALE

Syntax Description

MDY

specifies that SAS set the order as month, day, year.

YMD

specifies that SAS set the order as year, month, day.

DMY

specifies that SAS set the order as day, month, year.

LOCALE

specifies that SAS set the order based on the value that corresponds to the LOCALE= system option value and is one of the following: MDY | YMD | DMY.

Details

System option DATESTYLE= identifies the order of month, day, and year. The default value is LOCALE. The default LOCALE system option value is English. Therefore, the default DATESTYLE order is MDY.

To get the default settings for each locale option value, see [Locale Values](#).

See Also

Informats:

- “ANYDTDTE Informat” in *SAS Viya Formats and Informats: Reference*
- “ANYDXTDM Informat” in *SAS Viya Formats and Informats: Reference*
- “ANYDXTME Informat” in *SAS Viya Formats and Informats: Reference*

System Options:

- “LOCALE System Option” in *SAS Viya National Language Support: Reference Guide*

DECIMALCONV= System Option

Specifies the binary and decimal conversion methodology for formatting numeric data.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Log and Procedure Output Control: SAS Log and Procedure Output Log and Procedure Output Control: SAS Log Log and Procedure Output Control: Procedure Output
PROC OPTIONS GROUP=	LOG_LISTCONTROL LISTCONTROL LOGCONTROL
Default:	The shipped default is COMPATIBLE.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

DECIMALCONV=[COMPATIBLE](#) | [STDIEEE](#)

Syntax Description

COMPATIBLE

specifies to convert and format decimal values by using the conversion methodology that is compatible with releases prior to SAS 9.4. Use this option when SAS output might be processed by existing applications that are sensitive to the details of formatting.

Alias COMPAT

STDIEEE

specifies to convert and format decimal values by using the IEEE Standard for Floating-Point Arithmetic 754-2008. Using the STDIEEE argument improves the accuracy and readability of floating-point numbers. In some cases, more significant digits can be displayed in the same field width.

Details

Here are some of the specific improvements in decimal conversion and formatting when DECIMALCONV=STDIEEE:

- The BEST w . format might use scientific notation rather than fixed-point notation for fewer than three digits of precision. For example, a field with a width of 7 that was displayed as 0.00027 in previous releases might be displayed as 2.68E-4.
- For very short widths, sometimes the BEST w . format omits the decimal point for scientific-notation output in order to gain one or two significant digits (for example, display 137E7 instead of 1.4E9). Although this formatting can be used when DECIMALCONV= is set to either value, it is used more frequently when DECIMALCONV=STDIEEE.
- Although other formats, including $w.d$, Ew . and $Dw.d$ might be affected by the DECIMALCONV= setting, the changes are most noticeable when the BEST w . format is used.

See Also

Formats:

- [“BEST Format” in SAS Viya Formats and Informats: Reference](#)

DETAILS System Option

Specifies whether to include additional information when files are listed in a SAS library.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Categories: Log and Procedure Output Control: SAS Log and Procedure Output

Log and Procedure Output Control: SAS Log
 Log and Procedure Output Control: Procedure Output

**PROC OPTIONS
 GROUP=** LOG_LISTCONTROL
 LISTCONTROL
 LOGCONTROL

Default: The shipped default is NODETAILS.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[DETAILS](#) | [NODETAILS](#)

Syntax Description

DETAILS

includes additional information when some SAS procedures display a listing of files in a SAS library.

NODETAILS

does not include additional information.

Details

The DETAILS specification sets the default display for these components of SAS:

- the CONTENTS procedure
- the DATASETS procedure.

The type and amount of additional information that displays depends on which procedure you use.

DKRCOND= System Option

Specifies the level of error detection to report when a variable is missing from an input data set during the processing of a DROP=, KEEP=, or RENAME= data set option.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Categories: Files: SAS Files
 Environment Control: Error Handling

**PROC OPTIONS
 GROUP=** ERRORHANDLING
 SASFILES

Default: The shipped default is ERROR.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

DKRCOND=ERROR | WARN | WARNING | NOWARN | NOWARNING

Syntax Description**ERROR**

sets the error flag and writes an error message to the SAS log when a variable is missing from an input data set during the processing of a DROP=, KEEP=, or RENAME= data set option.

WARN | WARNING

writes a warning message to the SAS log when a variable is missing from an input data set during the processing of a DROP=, KEEP=, or RENAME= data set option.

NOWARN | NOWARNING

does not write a warning message to the SAS log when a variable is missing from an input data set during the processing of a DROP=, KEEP=, or RENAME= data set option.

Example

In the following statements, if the variable X is not in data set B and DKRICOND=ERROR, SAS sets the error flag to 1 and displays error messages:

```
data a;
  set b(drop=x);
run;
```

See Also**System Options:**

- [“DKROCOND= System Option” on page 70](#)

DKROCOND= System Option

Specifies the level of error detection to report when a variable is missing for an output data set during the processing of a DROP=, KEEP=, or RENAME= data set option.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Categories: Files: SAS Files
Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING
SASFILES

Default: The shipped default is WARN.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

DKROCOND=ERROR | WARN | WARNING | NOWARN | NOWARNING

Syntax Description

ERROR

sets the error flag and writes an error message to the SAS log when a variable is missing for an output data set during the processing of a DROP=, KEEP=, or RENAME= data set option.

WARN | WARNING

writes a warning message to the SAS log when a variable is missing for an output data set during the processing of a DROP=, KEEP=, or RENAME= data set option.

NOWARN | NOWARNING

does not write a warning message to the SAS log when a variable is missing for an output data set during the processing of a DROP=, KEEP=, or RENAME= data set option.

Example

In the following statements if the variable X is not in data set A and DKROCOND=ERROR, SAS sets the error flag to 1 and displays error messages:

```
data a;
  drop x;
run;
```

See Also

System Options:

- [“DKRICOND= System Option” on page 69](#)

DLCREATEDIR System Option

Specifies to create a directory for the SAS library that is named in a LIBNAME statement if the directory does not exist.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Files: SAS Files

PROC OPTIONS GROUP= SASFILES

Default: The shipped default is NODLCREATEDIR.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

DLCREATEDIR | NODLCREATEDIR

Syntax Description

DLCREATEDIR

specifies to create a directory for a SAS library that is named in a LIBNAME statement if the directory does not exist.

Restriction If the path specified in the LIBNAME statement contains multiple components, SAS creates only the final component in the path. If any intermediate components of the path do not exist, SAS does not assign the specified path. For example, when the code `libname mytestdir '/u/mydir/mysasprograms/test'` executes, and /mysasprograms exists, SAS creates the test directory. If /mysasprograms does not exist, SAS does not create the test directory.

NODLCREATEDIR

specifies not to create a directory for a SAS library that is named in a LIBNAME statement.

Details

SAS issues a note to the log when a directory for a SAS library is created.

See Also

Statements:

- “LIBNAME Statement” in *SAS Viya Statements: Reference*

DLDMGACTION= System Option

Specifies the type of action to take when a SAS data set or a SAS catalog is detected as damaged.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Files: SAS Files

PROC OPTIONS GROUP= SASFILES

Default: The shipped default is REPAIR for interactive mode and FAIL for batch mode.

Note: This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

`DLDMGACTION=FAIL | ABORT | REPAIR | NOINDEX | PROMPT`

Syntax Description

FAIL

stops the step and issues an error message to the log immediately.

ABORT

terminates the step and issues an error message to the log, and ends the SAS session.

REPAIR

For data files, automatically repairs and rebuilds indexes and integrity constraints, unless the data file is truncated. You use the REPAIR statement to restore the truncated data file. It issues a warning message to the log. For catalogs, REPAIR automatically deletes catalog entries for which an error occurs during the repair process.

NOINDEX

For data files, automatically repairs the data file without the indexes and integrity constraints, deletes the index file, updates the data file to reflect the disabled indexes and integrity constraints, and limits the data file to be opened only in INPUT mode.

Restriction NOINDEX does not apply to damaged catalogs or libraries, only data files.

PROMPT

For data sets, displays a dialog box where you can specify either FAIL, ABORT, REPAIR, or NOINDEX. For a damaged catalog or library, PROMPT displays a dialog box where you can specify either FAIL, ABORT, or REPAIR.

DSACCEL= System Option

Specifies whether the DATA step is enabled for parallel processing in CAS.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Language Control

PROC OPTIONS GROUP= LANGUAGECONTROL

Default: The shipped default is ANY.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

DSACCEL=ANY | NONE

Syntax Description**ANY**

enables the DATA step to execute in CAS.

NONE

disables the DATA step from executing in CAS.

Details

You can use the MSGLEVEL= system option to control the message detail that appears in CAS jobs:

- Specify MSGLEVEL=N to see only notes, warnings, and error messages.
- Specify MSGLEVEL=I to view additional CAS messages.

See Also

System options:

- [“MSGLEVEL= System Option”](#) on page 136

DSNFERR System Option

When a SAS data set cannot be found, specifies whether SAS issues an error message.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Environment Control: Error Handling
PROC OPTIONS GROUP=	ERRORHANDLING
Default:	The shipped default is DSNFERR.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

DSNFERR | **NODSNFERR**

Syntax Description

DSNFERR

specifies that SAS issue an error message and stop processing if a reference is made to a SAS data set that does not exist.

NODSNFERR

specifies that SAS ignore the error message and continue processing if a reference is made to a SAS data set that does not exist. The data set reference is treated as if `_NULL_` had been specified.

Details

- DSNFERR is similar to the BYERR system option, which issues an error message and stops processing if the SORT procedure attempts to sort a `_NULL_` data set.
- DSNFERR is similar to the VNFERR system option, which sets the error flag for a missing variable when a `_NULL_` data set is used.

See Also

System Options:

- [“BYERR System Option”](#) on page 48
- [“VNFERR System Option”](#) on page 224

DTRESET System Option

Specifies whether to update the date and time in the SAS log and in the procedure output file.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Log and Procedure Output Control: SAS Log and Procedure Output Log and Procedure Output Control: SAS Log Log and Procedure Output Control: Procedure Output
PROC OPTIONS GROUP=	LOG_LISTCONTROL LISTCONTROL LOGCONTROL
Default:	The shipped default is NODTRESET.
Interaction:	SAS Studio sets this option to DTRESET before each code submission. For more information, see “System Options in SAS Studio” on page 7 .
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

DTRESET | **NODTRESET**

Syntax Description

DTRESET

specifies that SAS update the date and time in the titles of the SAS log and the procedure output file.

NODTRESET

specifies that SAS not update the date and time in the titles of the SAS log and the procedure output file.

Details

The DTRESET system option updates the date and time in the titles of the SAS log and the procedure output file. This update occurs when the page is being written. The smallest time increment that is reflected is minutes.

The DTRESET option is especially helpful in obtaining a more accurate date and time stamp when you run long SAS jobs.

When you use NODTRESET, SAS displays the date and time that the job originally started.

ECHO System Option

Specifies a message to be echoed to stdout.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: SAS Log

**PROC OPTIONS
GROUP=** LOGCONTROL

Default: none

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

ECHO *'message'* | NOECHO

Required Arguments

ECHO *'message'*

specifies the text of the message to be echoed to stdout. The text must be enclosed in single or double quotation marks if the message is more than one word. Otherwise, the quotation marks are not needed.

NOECHO

specifies that no messages are to be echoed to stdout.

Details

You can specify multiple ECHO options. The strings are displayed in the order in which SAS encounters them. For more information, see [“Order of Precedence for Processing SAS System Options” on page 17](#).

For example, you can specify the following code:

```
-echo 'SAS is initializing.'
```

The message appears in the stdout location as SAS initializes.

See Also

System Options:

- [“ECHOAUTO System Option” on page 76](#)

ECHOAUTO System Option

Specifies whether the statements in the AUTOEXEC= file are written to the SAS log as they are executed.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: SAS Log

**PROC OPTIONS
GROUP=** LOGCONTROL

Default: The shipped default is NOECHOAUTO.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[ECHOAUTO](#) | [NOECHOAUTO](#)

Syntax Description

ECHOAUTO

specifies that the SAS statements in the AUTOEXEC= file are written to the SAS log as they are executed.

Requirement To print autoexec file statements in the SAS log, the SOURCE system option must be set.

NOECHOAUTO

specifies that SAS statements in the AUTOEXEC= file are not written in the SAS log, even though they are executed.

Details

Regardless of the setting of this option, messages that result from errors in the AUTOEXEC= files are printed in the SAS log.

See Also

System Options:

- [“AUTOEXEC System Option”](#) on page 43
- [“SOURCE System Option”](#) on page 186

ERRORABEND System Option

Specifies whether SAS responds to errors by terminating.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Alias: ERRABEND | NOERRABEND

Default: The shipped default is NOERRORABEND.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options”](#) on page 8.

Syntax

[ERRORABEND](#) | [NOERRORABEND](#)

Syntax Description**ERRORABEND**

specifies that SAS terminates for most errors (including syntax errors and file not found errors) that would normally cause it to issue an error message, set OBS=0, and go into syntax-check mode (if syntax checking is enabled). SAS also terminates if an error occurs in any global statement other than the LIBNAME and FILENAME statements.

Tip Use the ERRORABEND system option with SAS production programs, which presumably should not encounter any errors. If errors are encountered and ERRORABEND is in effect, SAS brings the errors to your attention immediately by terminating. ERRORABEND does not affect how SAS handles notes such as invalid data messages.

NOERRORABEND

specifies that SAS handle errors normally, that is, issue an error message, set OBS=0, and go into syntax-check mode (if syntax checking is enabled).

See Also

- “Global Statements” in *SAS Viya Statements: Reference*

System Options:

- “ERRORBYABEND System Option” on page 78
- “ERRORCHECK= System Option” on page 79

ERRORBYABEND System Option

Specifies whether SAS ends a program when an error occurs in BY-group processing.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Default: The shipped default is NOERRORBYABEND.

Note: This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

ERRORBYABEND | **NOERRORBYABEND**

Syntax Description**ERRORBYABEND**

specifies that SAS ends a program for BY-group error conditions that would normally cause it to issue an error message.

NOERRORBYABEND

specifies that SAS handle BY-group errors normally, that is, by issuing an error message and continuing processing.

Details

If SAS encounters one or more BY-group errors while ERRORBYABEND is in effect, SAS brings the errors to your attention immediately by ending your program. ERRORBYABEND does not affect how SAS handles notes that are written to the SAS log.

Note: Use the ERRORBYABEND system option with SAS production programs that should be error free.

See Also**System Options:**

- [“ERRORABEND System Option” on page 77](#)

ERRORCHECK= System Option

Specifies whether SAS enters syntax-check mode when errors are found in the LIBNAME, FILENAME, and %INCLUDE statements.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Default: The shipped default is NORMAL.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

ERRORCHECK=[NORMAL](#) | [STRICT](#)

Syntax Description**NORMAL**

specifies not to place the SAS program into syntax-check mode when an error occurs in a LIBNAME or FILENAME statement. In addition, the program or session does not terminate when a %INCLUDE statement fails due to a nonexistent file.

STRICT

specifies to place the SAS program into syntax-check mode when an error occurs in a LIBNAME or FILENAME statement. If the ERRORABEND system option is set and an error occurs in either a LIBNAME or FILENAME statement, SAS terminates. In addition, SAS terminates when a %INCLUDE statement fails due to a nonexistent file.

See Also

System Options:

- [“ERRORABEND System Option” on page 77](#)

ERRORS= System Option

Specifies the maximum number of observations for which SAS issues complete error messages.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Environment Control: Error Handling Log and Procedure Output Control: SAS Log
PROC OPTIONS GROUP=	ERRORHANDLING LOGCONTROL
Default:	The shipped default is 20.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

ERRORS=*n* | *nK* | *nM* | *nG* | *nT* | **MIN** | **MAX** | *hexX*

Syntax Description

n* | *nK* | *nM* | *nG* | *nT

specifies the number of observations for which SAS issues error messages in terms of 1 (*n*); 1,024 (*nK*); 1,048,576 (*nM*); 1,073,741,824 (*nG*); or 1,099,511,627,776 (*nT*). For example, a value of **8** specifies eight observations, and a value of **3M** specifies 3,145,728 observations.

MIN

sets the number of observations for which SAS issues error messages to 0.

MAX

sets the maximum number of observations for which SAS issues error messages to the largest signed, 4-byte integer representable in your operating environment.

hexX

specifies the maximum number of observations for which SAS issues error messages as a hexadecimal number. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **2dX** sets the maximum number of observations for which SAS issues error messages to 45 observations.

Details

If data errors are detected in more than *n* observations, processing continues, but SAS does not issue error messages for the additional errors.

Note: If you set **ERRORS=0** and an error occurs, or if the maximum number of errors has been reached, a warning message is displayed in the log, which states that the limit set by the **ERRORS** option has been reached.

FILELOCKS System Option

Specifies whether file locking is turned on or off and what action should be taken if a file cannot be locked.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Files: External Files Files: SAS Files
PROC OPTIONS GROUP=	EXTFILES ENVFILES SASFILES
Default:	The shipped default is FAIL
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

FILELOCKS=(*setting path* | *path setting*)

FILELOCKS=NONE | FAIL | CONTINUE | RESET

Required Arguments

setting

specifies the operating environment locking value for the specified path. The following values are valid:

- NONE
- FAIL
- CONTINUE
- RESET

path

specifies a path to a Linux directory. Enclose the path in single or double quotation marks.

Tip The *path* argument can contain an environment variable.

NONE

turns file locking off. NONE specifies that SAS attempts to open the file without checking for an existing lock on the file. NONE does not place an operating system lock on the file. These files are not protected from shared Update access.

Tip NONE does not suppress internal locking.

FAIL

turns file locking on. FAIL specifies that SAS attempts to place an operating system lock on the file. Access to the file is denied if the file is already locked, or if it cannot be locked. FAIL is the default value for FILELOCKS.

CONTINUE

turns file locking on. CONTINUE specifies that SAS attempts to place an operating system lock on the file. If a file is already locked by someone else, an attempt to open it fails. If the file cannot be locked for some other reason, the file is opened and a warning message is sent to the log. For example, a file cannot be locked if the file system does not support locking.

Tip CONTINUE does not suppress internal locking.

RESET

specifies that all previous FILELOCKS settings are deleted, and resets the global setting to the default value of FAIL. If you use the

FILELOCKS=(*setting path*|*path setting*) syntax, then RESET resets only those files that are in *path*.

Details***The Basics of File Locking***

The FILELOCKS system option enables you to lock both external files and SAS files based on global settings that you set in the FILELOCKS system option. External file locking applies to all files that are opened.

You can use multiple instances of the FILELOCKS option to establish different settings for different paths. One path can be a subdirectory of another path. In this case, the most specific matching path currently in effect governs operating system file locking. The following example shows how you can specify multiple instances of the FILELOCKS option in a configuration file:

```
filelocks=('/u/myuserid/temp' NONE)
filelocks=('/tmp' CONTINUE)
```

When the value of the FILELOCKS option is a set of *path* and *setting*, the path must be enclosed in quotation marks. If you use FILELOCKS on the command line, then quotation marks are not needed.

Note: To prevent data corruption, setting FILELOCKS to NONE or CONTINUE is not recommended.

Resetting Paths By Using the path and setting Arguments

The *path* and *setting* arguments enable you to apply a setting to a particular directory and its subtrees. If you set the value of *setting* to RESET, then the *path* and *setting* values are deleted.

For example, in the following case, **filelocks=('/' reset)**, the current values for *path* and *setting* are deleted, and FILELOCKS resets the values to the following default: (**/' fail**).

When FILELOCKS Is Set to FAIL

When FILELOCKS is set to FAIL (the default value), the following actions occur:

- SAS prevents two sessions from simultaneously opening the same SAS file for update or output.
- SAS prevents one session from reading a SAS file that another SAS session has open for update or output.
- SAS prevents one session from writing to a file that another SAS session has open in Read mode.

See Also

System Options:

- [“FILELOCKWAIT= System Option” on page 83](#)
- [“FILELOCKWAITMAX= System Option” on page 84](#)
- [“WORKINIT System Option” on page 230](#)

FILELOCKWAIT= System Option

Sets the number of seconds that SAS waits for a locked file to become available.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is 0.
Interaction:	The maximum value for FILELOCKWAIT= is based on the value of the FILELOCKWAITMAX= system option.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

FILELOCKWAIT=*wait-time*

Required Argument

wait-time

specifies the amount of time, in seconds, that SAS waits for a locked file to become available.

Details

Normally, SAS returns an error if the file that it attempts to access is locked. With the FILELOCKWAIT= system option, you can limit the amount of time SAS waits for a locked SAS file to become available. When you set FILELOCKWAIT= to a value of *wait-time*, SAS waits the specified amount of time for the file to become available before failing. When the time limit is reached, SAS returns a locked-file error, and the DATA step fails. The maximum time that you can set to wait for a locked file is 600 seconds (10 minutes). When you set FILELOCKWAIT= to 0, SAS immediately fails.

The FILELOCKWAIT= option is used primarily by a system administrator, who can change the maximum value of FILELOCKWAIT= using the FILELOCKWAITMAX= system option. FILELOCKWAITMAX= sets a maximum value for the FILELOCKWAIT= option. The default maximum value is 600 seconds (10 minutes), but a system administrator can set the value to 300 seconds (5 minutes) or to any other value that is less than or equal to 600. Changing the value of FILELOCKWAITMAX= does not affect the value of FILELOCKWAIT=, it affects only the maximum value of FILELOCKWAIT=. The FILELOCKWAIT= option can be restricted by a system administrator.

FILELOCKWAIT= is both a system option and a LIBNAME option. The system option applies to all SAS I/O files. The LIBNAME option applies to the members in a library only. The LIBNAME option overrides the system option.

See Also

System Options:

- “FILELOCKS System Option” on page 81
- “FILELOCKWAITMAX= System Option” on page 84

FILELOCKWAITMAX= System Option

Sets an upper limit on the time SAS waits for a locked file.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is 600.
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

FILELOCKWAITMAX=*wait-time*

Required Argument

wait-time

specifies the amount of time, in seconds, that SAS waits for a locked file to become available.

Default 600

Range 0–600

Details

The FILELOCKWAITMAX= system option, if used with the FILELOCKWAIT= system option, enables you to set the maximum amount of time SAS waits for a locked file to become available before failing. If you do not use the FILELOCKWAIT= system option, then the value of FILELOCKWAITMAX= does not affect the wait time.

Normally, SAS returns an error if the file that it attempts to access is locked. If you use the FILELOCKWAIT= system option, SAS waits the specified number of seconds for the file to become available before failing. By default, the maximum value of FILELOCKWAIT= is 600 seconds.

A system administrator can change the maximum value using the FILELOCKWAITMAX= system option. Setting FILELOCKWAITMAX=0 effectively turns off the FILELOCKWAIT= option.

FILELOCKWAIT= is both a system option and a LIBNAME statement option. The system option applies to all SAS I/O files. The LIBNAME option applies to the members in a library only. The LIBNAME option overrides the system option. FILELOCKWAITMAX= is a system option only.

See Also

System Options:

- [“FILELOCKS System Option” on page 81](#)
- [“FILELOCKWAIT= System Option” on page 83](#)

FILESYNC= System Option

Specifies when operating system buffers that contain contents of permanent SAS files are written to disk.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is HOST.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

FILESYNC= SAS | CLOSE | HOST | SAVE

Syntax Description

SAS

specifies that SAS requests the operating system to force buffered data to be written to disk when it is best for the integrity of the SAS file.

CLOSE

specifies that SAS requests the operating system to force buffered data to be written to disk when the SAS file is closed.

HOST

specifies that the operating system schedules when the buffered data for a SAS file is written to disk.

SAVE

specifies that the buffers are written to disk when the SAS file is saved.

Details

By using the FILESYNC= system option, SAS can tell the operating system when to force data that is temporarily stored in operating system buffers to be written to disk. Only SAS files in a permanent SAS library are affected; files in a temporary library are not affected.

If you specify a value other than the default value of HOST or CLOSE, the following occurs:

- the length of time it takes to run a SAS job increases
- the small chance of losing data in the event of a system failure is further reduced

Consult with your system administrator before you change the value of the FILESYNC= system option to a value other than the default value.

FIRSTOBS= System Option

Specifies the observation number or external file record that SAS processes first.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is 1.
Interaction:	SAS Studio sets FIRSTOBS=1 before each code submission. For more information, see “System Options in SAS Studio” on page 7 .
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

FIRSTOBS=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Syntax Description

n* | *nK* | *nM* | *nG* | *nT

specifies the number of the first observation or external file record to process, with *n* being an integer. Using one of the letter notations results in multiplying the integer by a specific value. That is, specifying K (kilo) multiplies the integer by 1,024; M (mega) multiplies by 1,048,576; G (giga) multiplies by 1,073,741,824; or T (tera) multiplies by 1,099,511,627,776. For example, a value of **8** specifies the eighth observations or records, and a value of **3m** specifies observation or record 3,145,728.

hexX

specifies the number of the first observation or the external file record to process as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **2dx** specifies the 45th observation.

MIN

sets the number of the first observation or external file record to process to 1. This is the default.

MAX

sets the number of the first observation to process to the maximum number of observations in the data sets or records in the external file, up to the largest eight-byte, signed integer, which is $2^{63}-1$, or approximately 9.2 quintillion observations.

Details

The FIRSTOBS= system option is valid for all steps for the duration of your current SAS session or until you change the setting. To affect any single SAS data set, use the FIRSTOBS= data set option.

You can apply FIRSTOBS= processing to WHERE processing.

Comparisons

- You can override the FIRSTOBS= system option by using the FIRSTOBS= data set option and by using the FIRSTOBS= option as a part of the INFILE statement.
- As the FIRSTOBS= system option specifies a starting point for processing, the OBS= system option specifies an ending point. The two options are often used together to define a range of observations or records to be processed.

Example

If you specify FIRSTOBS=50, SAS processes the 50th observation of the data set first.

This option applies to every input data set that is used in a program or a SAS process. In this example, SAS begins reading at the 11th observation in the data sets OLD, A, and B:

```
options firstobs=11;
data a;
    set old; /* 100 observations */
run;
data b;
    set a;
run;
data c;
    set b;
run;
```

Data set OLD has 100 observations, data set A has 90, B has 80, and C has 70. To avoid decreasing the number of observations in successive data sets, use the FIRSTOBS= data set option in the SET statement. You can also reset FIRSTOBS=1 between a DATA step and a PROC step.

See Also

Data Set Options:

- [“FIRSTOBS= Data Set Option”](#) in *SAS Viya Data Set Options: Reference*

Statements:

- [“INFILE Statement”](#) in *SAS Viya Statements: Reference*

System Options:

- [“OBS= System Option”](#) on page 142

FMTERR System Option

When a variable format cannot be found, specifies whether SAS generates an error or continues processing.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Environment Control: Error Handling
PROC OPTIONS GROUP=	ERRORHANDLING
Default:	The shipped default is FMTERR.
Restriction:	This option does not apply to CAS formats.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

FMTERR | **NOFMTERR**

Syntax Description

FMTERR

specifies that when SAS cannot find a specified variable format, it generates an error message and does not allow default substitution to occur.

NOFMTERR

replaces missing formats with the *w.* or *\$w.* default format, issues a note, and continues processing.

See Also

System Options:

- [“FMTSEARCH= System Option” on page 88](#)

FMTSEARCH= System Option

Specifies the order in which format catalogs are searched.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Environment Control: Files
PROC OPTIONS GROUP=	ENVFILES
Default:	The shipped default is (Work Library).

Restriction: This option does not apply to formats used in the CAS server. To manage user formats in the CAS server, see the [FMTSEARCH= CAS statement option](#) or the [LISTFMTSEARCH CAS statement option](#).

Requirement: Catalog specifications must be separated by a space.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Tip: You can use the APPEND or INSERT system options to add additional *catalog-specification*.

Syntax

FMTSEARCH=(*catalog-specification(s)*)

Syntax Description

catalog-specifications

searches format catalogs in the order listed, until the desired member is found.

The value of *catalog-specification* can be one of the following:

libref </LOCALE>

specifies to search the FORMATS catalog in the storage location that is specified by *libref*. When a *libref* is specified without a catalog, SAS uses FORMATS as the default catalog name.

If you specify /LOCALE, SAS searches for a catalog that is associated with the current SAS locale before it searches the FORMATS catalog. The locale catalog name is based on the POSIX locale name for the current locale. Two catalogs might exist for each POSIX locale name: one catalog for the language and one catalog for the language_country. If your current SAS locale is English_India, the POSIX locale name is en_IN. The two possible locale catalogs are *libref.FORMATS_en* and *libref.FORMATS_en_IN*. SAS searches, in order, these catalogs in *libref*:

1. *libref.FORMATS_language_country*
2. *libref.FORMATS_language*
3. *libref.FORMATS*

Tip You can obtain a POSIX locale value by using the [GETPXLOCALE function](#). You can use the [GETLOCENV function](#) to obtain the current SAS locale. For more information, see [SAS Viya National Language Support: Reference Guide](#).

See For a list of POSIX locale values and their corresponding SAS locale names, see [“LOCALE= Values for PAPERSIZE and DFLANG Options” in SAS Viya National Language Support: Reference Guide](#).

libref.catalog </LOCALE>

specifies to search for a specific library and catalog.

If you specify /LOCALE, SAS searches *libref.catalog* for a catalog that is associated with the current SAS locale. The locale catalog name is based on the POSIX locale name for the current locale. Two catalogs might exist for each POSIX locale name: one catalog for the language and one catalog for the language_country. If your current SAS locale is English_India, the POSIX locale

name is en_IN. The two possible locale catalogs are *libref.catalog_en* and *libref.catalog_en_IN*.

SAS searches, in order, these catalogs in *libref*, if you specify /LOCALE:

1. *libref.catalog_language_country*
2. *libref.catalog_language*
3. *libref.catalog*

Tip You can obtain a POSIX locale value by using the [GETPXLOCALE function](#). You can use the [GETLOCENV function](#) to obtain the current SAS locale. For more information, see *SAS Viya National Language Support: Reference Guide*.

See For a list of POSIX locale values and their corresponding SAS locale names, see “LOCALE= Values for PAPERSIZE and DFLANG Options” in *SAS Viya National Language Support: Reference Guide*.

Details

The default value for FMTSEARCH is (WORK LIBRARY). The catalogs Work.Formats and Library.Formats are always searched, regardless of whether they appear in the FMTSEARCH option. The Work.Formats catalog is always searched first unless it appears in the FMTSEARCH option. The Library.Formats catalog is searched second unless it appears in the FMTSEARCH option.

For example, FMTSEARCH=(MYLIB LIBRARY) results in searching these catalogs, in the order Work.Formats, Mylib.Formats, and Library.Formats.

If a catalog appears in the FMTSEARCH= list, the catalog is searched in the order in which it appears in the list. If a catalog in the list does not exist, that particular catalog is ignored and searching continues with no error or warning message given.

You can use the INSERT and APPEND system options to add additional format catalogs to search.

Examples

Example 1: Format Catalog Search Order with Default Libraries Searched First

If you specify FMTSEARCH=(ABC DEF.XYZ GHI), SAS searches for requested formats or informats in this order:

1. Work.Formats
2. Library.Formats
3. Abc.Formats
4. Def.Xyz
5. Ghi.Formats

Example 2: Format Catalog Search Order with Default Libraries Searched Last

If you specify FMTSEARCH=(ABC WORK LIBRARY), SAS searches in this order:

1. Abc.Formats

2. Work.Formats
3. Library.Formats

Because WORK appears in the FMTSEARCH list, Work.Formats is not automatically searched first.

Example 3: Format Catalog Search Order with POSIX Locale Values

If you specify FMTSEARCH=(ABC/LOCALE) and the current locale is German_Germany, SAS searches in this order:

1. Work.Formats
2. Library.Formats
3. Abc.Formats_de_DE
4. Abc.Formats_de
5. Abc.Formats

See Also

User-Defined Formats in CAS

- [“User-Defined Format Basics”](#) in *SAS Cloud Analytic Services: User-Defined Formats*
- [CAS Statement](#) user-defined format options

Procedures:

- [“FORMAT Procedure”](#) in *SAS Viya Visual Data Management and Utility Procedures Guide*

System Options:

- [“APPEND= System Option”](#) on page 39
- [“INSERT= System Option”](#) on page 99
- [“FMTERR System Option”](#) on page 88

FORMCHAR= System Option

Specifies the default output formatting characters.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: Procedure Output

PROC OPTIONS GROUP= LISTCONTROL

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options”](#) on page 8.

Syntax

FORMCHAR= *'formatting-characters'*

Syntax Description

'formatting-characters'

specifies any string or list of strings of characters up to 64 bytes long. If fewer than 64 bytes are specified, the string is padded with blanks on the right.

Tip For consistent results when you move your document to different computers, issue the following OPTIONS statement before using ODS destinations: :

```
options formchar="|----|+|----+|-\<>*";
```

Details

If you omit formatting characters as an option in a procedure that uses formatting characters, the default specifications given in the FORMCHAR= system option are used. Note that you can also specify a hexadecimal character constant as a formatting character. When you use a hexadecimal constant with this option, SAS interprets the value of the hexadecimal constant as appropriate for your operating system.

Note: To ensure that row and column separators and boxed tabular reports are printed legibly when using the standard forms characters, you must use these resources:

- either the SAS Monospace or the SAS Monospace Bold font
- a printer that supports TrueType fonts

See Also

[SAS Viya Universal Printing](#)

FULLSTIMER System Option

Specifies whether to write all available system performance statistics and the datetime stamp to the SAS log.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: SAS Log

PROC OPTIONS GROUP= LOGCONTROL

Default: The shipped default is NOFULLSTIMER.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

-FULLSTIMER | -NOFULLSTIMER

FULLSTIMER | NOFULLSTIMER

Required Arguments

FULLSTIMER

writes to the SAS log a list of the host-dependent resources that were used for each step and for the entire SAS session. A datetime stamp is included in the output.

NOFULLSTIMER

does not write to the SAS log a complete list of resources or a datetime stamp.

Details

SAS uses Linux system calls for your operating environment to get the statistical information from FULLSTIMER. The datetime stamp is listed in the output. You can change the behavior and format of the statistical information by using the STIMFMT system option.

Here is an example of FULLSTIMER output:

Log 2.1 FULLSTIMER Output

```
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      user cpu time       0.00 seconds
      system cpu time     0.00 seconds
      memory              769.75k
      OS Memory          24988.00k
      Timestamp           03/10/2016 12:49:24 PM
      Step Count                   19  Switch Count   32
      Page Faults                   0
      Page Reclaims                 264
      Page Swaps                     0
      Voluntary Context Switches     90
      Involuntary Context Switches   90
      Block Input Operations          0
      Block Output Operations         0
```

Note: If both FULLSTIMER and STIMER system options are set, the FULLSTIMER statistics are written to the log.

FULLSTIMER displays the following statistics:

Table 2.1 Description of FULLSTIMER Statistics

Statistic	Description
Real Time	the amount of real time (clock time) that is spent to process the SAS job. Real time is also referred to as elapsed time.
User CPU Time	the CPU time that is spent in the user program.
System CPU Time	the CPU time that is spent to perform operating system tasks (system overhead tasks) that support the execution of your SAS code.
Memory	the amount of memory required to run a step.
OS Memory	the largest amount of operating system memory that is available to SAS during the step.

Statistic	Description
Timestamp	the date and time that a step was executed.
Step Count	the number of PROC and DATA steps in a SAS job.
Switch Count	for SAS use only.
Page Faults	the number of pages that SAS tried to access but were not in main memory and required I/O activity.
Page Reclaims	the number of pages that were accessed without I/O activity.
Page Swaps	the number of times a process was swapped out of main memory.
Voluntary Context Switches	the number of times that the SAS process had to pause because of a resource constraint such as a disk drive.
Involuntary Context Switches	the number of times that the operating system forced the SAS session to pause processing to allow other process to run.
Block Input Operations	the number of I/O operations that are performed to read the data into memory.
Block Output Operations	the number of I/O operations that are performed to write the data to a file.

For more information about these statistics, see the man pages for the `getrusage()` and `times()` Linux system calls.

Note: Some procedures use multiple threads. On computers with multiple CPUs, the operating system can run more than one thread simultaneously. Consequently, CPU time might exceed real time in your FULLSTIMER output. For example, a SAS procedure could use two threads that run on two separate CPUs simultaneously. The value of CPU time would be calculated as the following:

```

CPU1 time + CPU2 time = total CPU time
1 second + 1 second = 2 seconds

```

Because CPU1 can run a thread at the same time that CPU2 runs a separate thread for the same SAS process, you can theoretically consume 2 CPU seconds in 1 second of real time.

See Also

System Options:

- “STIMEFMT System Option” on page 193
- “STIMER System Option” on page 197

HOSTINFOLONG System Option

Specifies to print additional operating environment information in the SAS log when SAS starts.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: SAS Log

PROC OPTIONS GROUP= LOGCONTROL

Default: The shipped default is HOSTINFOLONG.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

HOSTINFOLONG | **NOHOSTINFOLONG**

Syntax Description

HOSTINFOLONG

specifies to print additional operating environment information in the SAS log when SAS starts.

NOHOSTINFOLONG

specifies to omit additional operating environment information in the SAS log when SAS starts.

Details

When HOSTINFOLONG is specified, SAS writes additional information about the operating environment to the SAS log. Here is an example:

```
NOTE: Copyright (c) 2016 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software V.03.02 (TS1M0 MBCS3280)
      Licensed to SAS Institute Inc., Site 1.
NOTE: This session is executing on the Linux 2.6.32-573.el6.x86_64 (LIN X64)
      platform.

NOTE: Additional host information:

Linux LIN X64 2.6.32-573.1.el6.x86_64 #1 SMP Wed Feb 1 18:23:37 EDT 2015
      x86_64 Red Hat Enterprise Linux Server release 6.7 (Santiago)
```

See Also

System Options:

- [“CUID System Option” on page 63](#)

IBUFNO= System Option

Specifies an optional number of extra buffers to be allocated for navigating an index file.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is 0.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

IBUFNO=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Syntax Description

n | *nK* | *nM* | *nG* | *nT*

specifies the number of extra index buffers to be allocated in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). For example, a value of **8** specifies eight buffers, and a value of **3k** specifies 3,072 buffers.

Restriction Maximum value is 10,000.

hexX

specifies the number of extra index buffers as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **2dx** specifies 45 buffers.

MIN

sets the number of extra index buffers to 0.

MAX

sets the maximum number of extra index buffers to 10,000.

Details

An index is an optional SAS file that you can create for a SAS data file in order to provide direct access to specific observations. The index file consists of entries that are organized into hierarchical levels, such as a tree structure, and connected by pointers. When an index is used to process a request, such as for WHERE processing, SAS does a binary search on the index file and positions the index to the first entry that contains a qualified value. SAS uses the value's identifier to directly access the observation that contains the value. SAS requires memory for buffers when an index is actually used. The buffers are not required unless SAS uses the index, but they must be allocated in preparation for the index that is being used.

SAS automatically allocates a minimal number of buffers in order to navigate the index file. Typically, you do not need to specify extra buffers. However, using **IBUFNO**= to specify extra buffers could improve execution time by limiting the number of input/output operations that are required for a particular index file. However, the improvement in execution time comes at the expense of increased memory consumption.

The optimal value for buffer system options is dependent on your operating environment. Experiment with various buffers sizes to determine the optimal value for these system options.

Note: Whereas too few buffers allocated to the index file decrease performance, over allocation of index buffers creates performance problems as well. Experimentation is the best way to determine the optimal number of index buffers. For example, experiment with `ibufno=3`, then `ibufno=4`, and so on, until you find the least number of buffers that produces satisfactory performance results.

See Also

System Options:

- [“IBUFSIZE= System Option” on page 97](#)

IBUFSIZE= System Option

Specifies the buffer size for an index file.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is 0.
Restriction:	Specify a page size before the index file is created. After it is created, you cannot change the page size.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

IBUFSIZE=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | **MAX**

Note: You can also use the KB, MB, GB, and TB syntax notations.

Syntax Description

n* | *nK* | *nM* | *nG* | *nT

specifies the buffer size to process in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). For example, a value of **8** specifies 8 bytes, and a value of **3k** specifies 3,072 bytes. A value of 0 causes SAS to use the minimum optimal buffer size for the operating environment.

hexX

specifies the buffer size as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **2dx** sets the page size to 45 bytes.

MAX

sets the buffer size for an index file to the maximum possible number. For **IBUFSIZE=**, the value is 32,767 bytes.

Details

An index is an optional SAS file that you can create for a SAS data file in order to provide direct access to specific observations. The index file consists of entries that are organized into hierarchical levels, such as a tree structure, and connected by pointers. When an index is used to process a request, such as for WHERE processing, SAS does a search on the index file in order to rapidly locate the requested records.

Typically, you do not need to specify an index buffer size. However, the following situations could require a different buffer size:

- The buffer size affects the number of levels in the index. The more buffers there are, the more levels in the index. The more levels, the longer the index search takes. Increasing the buffer size allows more index values to be stored in each buffer, thus reducing the number of buffers (and the number of levels). The number of buffers required for the index varies with the buffer size, the length of the index value, and the values themselves. The main resource that is saved when reducing levels in the index is I/O. If your application is experiencing a lot of I/O in the index file, increasing the buffer size might help. However, you must re-create the index file after increasing the buffer size.
- The index file structure requires a minimum of three index values to be stored on a page. If the length of an index value is very large, you might get an error message that the index could not be created because the buffer size is too small to hold three index values. Increasing the buffer size should eliminate the error.

The optimal value for buffer system options is dependent on your operating environment. Experiment with various buffers sizes to determine the optimal value for these system options.

See Also

System Options:

- [“IBUFNO= System Option” on page 95](#)

INITSTMT= System Option

Specifies a SAS statement to execute after any statements in the AUTOEXEC= file and before any statements from the SYSIN= file.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Initialization and Operation
PROC OPTIONS GROUP=	EXECMODES
Alias:	IS=
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

INITSTMT=*'statement'*

Syntax Description

'statement'

specifies any SAS statement or statements.

Requirement *statement* must be able to run on a step boundary.

Comparisons

INITSTMT= specifies the SAS statements to be executed at SAS initialization, and the TERMSTMT= system option specifies the SAS statements to be executed at SAS termination.

Example

Here is an example of using this option on UNIX:

```
sas -initstmt '%put you have used the initstmt; data x; x=1;
run;'
```

See Also

System Options:

- [“TERMSTMT= System Option” on page 203](#)

INSERT= System Option

Inserts the specified value as the first value of the specified system option.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Files

PROC OPTIONS GROUP= ENVFILES

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

INSERT=(*system-option-1=argument-1* <*system-option-2=argument-2* ...>)

Syntax Description

system-option

can be AUTOEXEC, FMTSEARCH, MSG, SASAUTOS, SASHELP, SASSCRIPT, or SET.

Note Some of these options are available only when SAS starts. These options can be specified in the INSERT= option only when the INSERT= option is specified in a configuration file or a SAS command.

argument

specifies a new value that you want as the first value of *system-option*.

argument can be any value that could be specified for *system-option* if *system-option* is set using the OPTIONS statement.

Restriction The arguments for the FMTSEARCH system option apply only to format catalogs. They do not apply to CAS format libraries. For information about CAS format library search order, see the [CAS statement FMTSEARCH= option](#).

Details

If you specify a new value for the AUTOEXEC, FMTSEARCH, MSG, SASAUTOS, SASHELP, SASSCRIPT, or SET system options, the new value replaces the value of the option. Instead of replacing the value, you can use the INSERT= system option to add an additional value to the option as the first value of the option.

For a list of system options that the INSERT= system option and the APPEND= system option support, including the system options that can be used when SAS starts, submit the following OPTIONS procedure:

```
proc options listinsertappend;
run;
```

For example, if you enter the following SAS command, the only location in which SAS looks for help files is **/startup**. The output of PROC OPTIONS shows only **/startup**.

```
sas -autoexec /startup -config /myprogs/prog1.sas
```

If you want SAS to look in the current path for help files and **/startup/myautoexec**, and if you want SAS to look first in **/startup**, then you must use the INSERT option.

```
sas -insert autoexec /startup/myautoexec
```

Comparisons

The INSERT= system option adds a new value to the beginning of the current value of the AUTOEXEC, FMTSEARCH, MSG, SASAUTOS, SASHELP, SASSCRIPT, or SET system options. The APPEND= system option adds a new value to the end of one of these system options.

Example

The following table shows the results of adding a value to the beginning of the FMTSEARCH= option value:

Current FMTSEARCH= Value	Value of INSERT= System Option	New FMTSEARCH= Value
(WORK LIBRARY)	(fmtsearch=(abc def))	(ABC DEF WORK LIBRARY)

See Also

- [“Changing an Option Value By Using the INSERT and APPEND System Options” on page 13](#)

- [CAS Statement FMTSEARCH= option](#)
- [CAS Statement LISTFMTSEARCH= option](#)

System Options:

- [“APPEND= System Option” on page 39](#)

INTERVALDS= System Option

Specifies one or more interval name-value pairs, where the value is a SAS data set that contains user-defined intervals. The intervals can be used as arguments to the INTNX and INTCK functions.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Input Control: Data Processing

PROC OPTIONS GROUP= INPUTCONTROL

Requirement: The set of interval-value pairs must be enclosed in parentheses.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

```
INTERVALDS=(interval-1=libref.dataset-name-1
<interval-2=libref.dataset-name-2 ...> )
```

Syntax Description

interval

specifies the name of an interval. The value of *interval* is used to represent the set of intervals that is specified in *libref.dataset-name*.

Restriction *interval* cannot be a reserved SAS name.

Requirement When you specify multiple intervals, the interval name must not be the same as another interval.

libref.dataset-name

specifies the libref and the data set name of the file that contains the user-defined intervals.

Details

The INTCK and INTNX functions specify *interval* as the interval name in the function argument list to reference a data set that names user-defined intervals.

The same *libref.dataset-name* can be assigned to different intervals. An error occurs when more than one *interval* of the same name is defined for the INTERVALDS system option.

Example

This example assigns a single data set to an interval on the SAS command line or in a configuration file.

```
-intervals (mycompany=mycompany.storeHours)
```

The next example assigns multiple intervals using the OPTIONS statement. The intervals subsid1 and subsid2 are assigned the same libref and data set name.

```
options intervals="mycompany=mycompany.storeHours
               subsid1=subsid.storeHours subsid2=subsid.storeHours";
```

See Also

Functions:

- “INTCK Function” in *SAS Viya Functions and CALL Routines: Reference*
- “INTNX Function” in *SAS Viya Functions and CALL Routines: Reference*

INVALIDDATA= System Option

Specifies the value that SAS assigns to a variable when invalid numeric data is encountered.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Input Control: Data Processing
PROC OPTIONS GROUP=	INPUTCONTROL
Default:	The shipped default is a period (.).
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

```
INVALIDDATA=character'
```

Syntax Description

character'

specifies the value to be assigned, which can be a letter (A through Z, a through z), a period (.), or an underscore (_).

Details

The INVALIDDATA= system option specifies the value that SAS is to assign to a variable when invalid numeric data is read with an INPUT statement or the INPUT function.

JREOPTIONS System Option

Identifies the Java Runtime Environment (JRE) options for SAS.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Initialization and Operation
PROC OPTIONS GROUP=	EXECMODES
Default:	none

CAUTION: **Changing Java options that affect SAS could cause SAS to fail.** Before you change the settings for the JREOPTIONS option, contact SAS Technical Support to make sure that the Java setting that you want to change will not cause SAS to fail. A best practice is to change only the Java properties for your own Java code.

Syntax

-JREOPTIONS (*-JRE-option-1* <*-JRE-option-n*>)

Required Argument

-JRE-option

specifies one or more JRE options.

JRE options must begin with a hyphen (-). Use a space to separate multiple JRE options. Valid values for *JRE-option* depend on your installation's JRE. For information about JRE options, see your installation's Java documentation.

Details

JRE options must be enclosed in parentheses. On the command line, you must put a backslash (\) before the open parenthesis and close parenthesis, as shown in the examples below. If you specify multiple JREOPTIONS options, then SAS appends JRE options to JRE options that are currently defined. Incorrect JRE options are ignored.

Example: Using JRE Options

```
-jreoptions \(-Dmy.java.property\)
-jreoptions \(-Xmx512m -Xms256m\)
```

LABEL System Option

Specifies whether SAS procedures can use labels with variables.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: Procedure Output
PROC OPTIONS GROUP=	LISTCONTROL

Default: The shipped default is LABEL.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

LABEL | **NOLABEL**

Syntax Description

LABEL

specifies that SAS procedures can use labels with variables. The LABEL system option must be in effect before the LABEL option of any procedure can be used.

NOLABEL

specifies that SAS procedures cannot use labels with variables. If NOLABEL is specified, the LABEL option of a procedure is ignored.

Details

A *label* is a string of up to 256 characters that can be written by certain procedures in place of the variable's name.

See Also

Data Set Options:

- [“LABEL= Data Set Option” in SAS Viya Data Set Options: Reference](#)

LABELCHKPT System Option

Specifies whether checkpoint-restart data for labeled code sections is to be recorded for batch programs.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

**PROC OPTIONS
GROUP=** ERRORHANDLING

Default: The shipped default is NOLABELCHKPT.

Restriction: The LABELCHKPT system option can be specified only if the STEPCHKPT system option is not specified when SAS starts.

Requirement: This option can be used only in batch mode.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

LABELCHKPT | **NOLABELCHKPT**

Syntax Description

LABELCHKPT

enables checkpoint mode for labeled code sections, which specifies to record checkpoint-restart data.

NOLABELCHKPT

disables checkpoint mode for labeled code sections, which specifies not to record checkpoint-restart data.

Details

Using the LABELCHKPT system option puts SAS in checkpoint mode for SAS programs that run in batch. Each time a label is encountered, SAS records data in a checkpoint-restart library. If a program terminates without completing, the program can be resubmitted, beginning at the labeled code section that was executing when the program terminated.

To ensure that the checkpoint-restart data is accurate, specify the ERRORCHECK STRICT option and set the ERRORABEND option. By setting these options, SAS terminates for most errors.

SAS can run in checkpoint-restart mode either for labeled code sections or for DATA and PROC steps, but not both.

Comparisons

The LABELCHKPT system option enables checkpoint mode for labeled code sections in batch programs that terminate before completing. Execution resumes at the labeled code section that was executing when the failure occurred.

The STEPCHKPT system option enables checkpoint mode for DATA and PROC steps in batch programs that terminate before completing. Execution resumes with the DATA or PROC step that was executing when the failure occurred.

See Also

Statements:

- [“CHECKPOINT EXECUTE_ALWAYS Statement” in SAS Viya Statements: Reference](#)

System Options:

- [“CHKPTCLEAN System Option” on page 58](#)
- [“LABELCHKPTLIB= System Option” on page 105](#)
- [“LABELRESTART System Option” on page 107](#)
- [“STEPCHKPT System Option” on page 189](#)

LABELCHKPTLIB= System Option

Specifies the libref of the library where the checkpoint-restart data is saved for labeled code sections.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

**PROC OPTIONS
GROUP=**

ERRORHANDLING

Default: The shipped default is Work.**Restriction:** The LABELCHKPTLIB= system option can be specified only if the STEPCHKPT system option is not specified when SAS starts.**Requirement:** This option can be used only in batch mode.**Note:** This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

LABELCHKPTLIB=*libref*

Syntax Description

libref

specifies the libref that identifies the library where the checkpoint-restart data is saved.

Requirement The LIBNAME statement that identifies the checkpoint-restart library must use the BASE engine and be the first statement in the batch program.

Details

When the LABELCHKPT system option is specified, checkpoint-restart data for labeled code sections in batch programs is saved in the libref that is specified in the LABELCHKPTLIB= system option. If no libref is specified, SAS uses the Work library to save checkpoint data. The LIBNAME statement that defines the libref must be the first statement in the batch program.

If the Work library is used to save checkpoint data, the NOWORKTERM and NOWORKINIT system options must be specified. When you set these options, the checkpoint-restart data is available when the batch program is resubmitted. These two options ensure that the Work library is saved when SAS ends and is restored when SAS starts. If the NOWORKTERM option is not specified, the Work library is deleted at the end of the SAS session and the checkpoint-restart data is lost. If the NOWORKINIT option is not specified, a new Work library is created when SAS starts, and again the checkpoint-restart data is lost.

The LABELCHKPTLIB= option must be specified for any SAS session that accesses checkpoint-restart data that is collected at label points and that is not saved to the Work library.

Comparisons

When the LABELCHKPT system option is set, the library that is specified by the LABELCHKPTLIB system option names the library where the checkpoint-restart data is saved for labeled code sections. When the LABELRESTART system option is set, the library that is specified by the LABELCHKPTLIB system option names the library where the checkpoint-restart data is used to resume execution of labeled code sections.

When the STEPCHKPT system option is set, the library that is specified by the STEPCHKPTLIB system option names the library where the checkpoint-restart data is saved for DATA and PROC steps. When the STEPSTART system option is set, the

library that is specified by the STEPCHKPTLIB system option names the library where the checkpoint-restart data is used to resume execution of DATA and PROC steps.

See Also

Statements:

- [“CHECKPOINT EXECUTE_ALWAYS Statement”](#) in *SAS Viya Statements: Reference*

System Options:

- [“LABELCHKPT System Option”](#) on page 104
- [“LABELRESTART System Option”](#) on page 107
- [“STEPCHKPT System Option”](#) on page 189
- [“WORKINIT System Option”](#) on page 230
- [“WORKTERM System Option”](#) on page 231

LABELRESTART System Option

Specifies whether to execute a batch program by using checkpoint-restart data for data collected at labeled code sections.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Error Handling
PROC OPTIONS GROUP=	ERRORHANDLING
Default:	The shipped default is NOLABELRESTART.
Restriction:	The LABELRESTART system option can be specified only if the STEPCHKPT system option is not specified when SAS starts.
Requirement:	This option can be used only in batch mode.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

LABELRESTART | **NOLABELRESTART**

Syntax Description

LABELRESTART

enables restart mode, which specifies to execute the batch program by using the checkpoint-restart data.

NOLABELRESTART

disables restart mode, which specifies not to execute the batch program by using checkpoint-restart data.

Details

You specify the LABELRESTART option when you want to resubmit a batch program that ran in checkpoint mode for labeled code sections and terminated before the batch program completed. When you resubmit the batch program, SAS determines from the checkpoint data the label that was executing when the program terminated. The program resumes executing the batch program at that label.

Comparisons

When you specify the LABELRESTART option, SAS uses the checkpoint-restart data for labeled code sections to resume execution of batch programs.

When you specify the STEPSTART option, SAS uses the checkpoint-restart data for DATA and PROC steps to resume execution of batch programs.

See Also

Statements:

- [“CHECKPOINT EXECUTE_ALWAYS Statement”](#) in *SAS Viya Statements: Reference*

System Options:

- [“CHKPTCLEAN System Option”](#) on page 58
- [“LABELCHKPT System Option”](#) on page 104
- [“LABELCHKPTLIB= System Option”](#) on page 105
- [“STEPCHKPT System Option”](#) on page 189
- [“STEPRESTART System Option”](#) on page 192

LAST = System Option

Specifies the most recently created data set.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is <code>_NULL_</code> .
Restriction:	<code>_LAST_ =</code> is not allowed with data set options.
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

`_LAST_ = SAS-data-set`

Syntax Description

SAS-data-set

specifies a SAS data set name.

Restriction No data set options are allowed.

Tip Use *libref.membername* or *membername* syntax, not a string that is enclosed in quotation marks, to specify a SAS data set name.

Details

By default, SAS automatically keeps track of the most recently created SAS data set. Use the `_LAST_` system option to override the default.

LINESIZE= System Option

Specifies the line size for the SAS log and for SAS procedure output.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Categories: Log and Procedure Output Control: SAS Log and Procedure Output
Log and Procedure Output Control: SAS Log
Log and Procedure Output Control: Procedure Output

PROC OPTIONS GROUP= LOG_LISTCONTROL
LISTCONTROL
LOGCONTROL

Alias: LS=

Default: The display width setting for interactive modes; 132 for batch mode

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

`LINESIZE=n | MIN | MAX | hexX`

Syntax Description

n
specifies the number of characters in a line.

MIN
sets the number of characters in a line to 64.

MAX
sets the number of characters in a line to 256.

hexX
specifies the number of characters in a line as a hexadecimal number. You must specify the value beginning with a number (0–9), followed by an X. For example, the value `0FAx` sets the line size of the SAS procedure output to 250.

Details

The `LINESIZE=` system option specifies the line size (printer line width) in characters for the SAS log and the SAS output that is produced for an ODS markup destination by a DATA step where the FILE statement destination is PRINT (the FILE PRINT ODS statement is not affected by the `LINESIZE=` system option)

LOCKDOWN System Option

Enables the ability to limit access to files and to specific SAS features for a SAS session executing in a batch or the workspace server.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Initialization and Operation
PROC OPTIONS GROUP=	EXECMODES
Default:	NOLOCKDOWN
Requirement:	XCMD must be disabled to use the LOCKDOWN option. See “XCMD System Option” on page 232 .
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .
See:	“SAS Workspace Server and SAS Object Spawner: How To” in SAS Viya Administration: Workspace Server and Object Spawner “LOCKDOWN Statement” in SAS Viya Statements: Reference

Syntax

`LOCKDOWN | NOLOCKDOWN`

Syntax Description

LOCKDOWN

enables the ability to limit access to files and to specific SAS features for a SAS session executing in a batch or server processing mode.

NOLOCKDOWN

disables the ability to limit access to files and to specific SAS features for a SAS session executing in a batch or server processing mode.

Details

In addition to LOCKDOWN, security administrators also rely on the NOXCMD system option. For more information, see [“XCMD System Option” on page 232](#).

When the LOCKDOWN option is specified for a SAS session, SAS enters a *locked-down state* at a lockdown point. A SAS session in the locked-down state has following restrictions:

- limited file system access

All access to local files and directories are validated through the lockdown path list.

The lockdown path list specifies which host file resources are available when a SAS session is in the locked-down state. This list includes default system directories and

user directories and files. For details about the lockdown path list, see “[LOCKDOWN Statement](#)” in *SAS Viya Statements: Reference*.

- limited SAS language features

The following SAS language features are disabled:

- DATA step Java Object "javaobj"
- PROC JAVAINFO
- FUNCTIONS: ADDR, ADDRLONG, PEEK, PEEKLONG, PEEKC, PEEKCLONG, POKE, POKELONG, and MODULE

LOCKDOWN does not take effect in a SAS session until *after* the lockdown point.

The *lockdown point* is the point during SAS execution when the following tasks have completed to establish a user’s SAS environment:

- SAS session initialization
- AUTOEXEC execution
- INITSTMT execution

During initialization of the user’s SAS environment, all paths and files are available and work as designed (for example, SASHELP, WORK, LOG, and so on). AUTOEXEC pre-assigned libraries also work as designed. When initialization is complete, SAS is put in the locked-down state with limited file system access.

LOG System Option

Specifies a destination for the SAS log when running in batch mode.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Files
PROC OPTIONS GROUP=	ENVFILES, LOGCONTROL
Default:	A file in the current directory with the same filename as the SAS source file and an extension of .log.
Note:	This option cannot be restricted by a site administrator. For more information, see “ Restricted Options ” on page 8.

Syntax

LOG *file-specification* | -NOLOG

Required Arguments

LOG *file-specification*

specifies the destination for the SAS log. The *file-specification* can be any valid Linux path to a directory, a filename, or an environment variable that is associated with a path. If you specify only the path to a directory, the log file is created in the specified directory. The default name for this file is *filename.log*, where *filename* is the name of your SAS job.

NOLOG

suppresses the creation of the SAS log. Do not use this value unless your SAS program is thoroughly debugged.

Details

The LOG system option specifies a destination for the SAS log when running in batch mode. The LOG system option is valid in batch mode; it is ignored in interactive modes.

Using directives in the value of the LOG system option enables you to control when logs are open and closed and how they are named, based on real-time events such as time, month, day of week, and so on. For a valid list of directives, see “LOGPARM= System Option” on page 112.

See Also**System Options:**

- “LOGPARM= System Option” on page 112

LOGPARM= System Option

Specifies when SAS log files are opened, closed, and, in conjunction with the LOG= system option, how they are named.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: SAS Log

PROC OPTIONS GROUP= LOGCONTROL

Default: The shipped default is WRITE=BUFFERED ROLLOVER=NONE OPEN=REPLACE.

Note: This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

```
LOGPARM="<OPEN= APPEND | REPLACE | REPLACEOLD>
< ROLLOVER= AUTO | NONE | SESSION | n | nK | nM | nG>
< WRITE= BUFFERED | IMMEDIATE> "
```

Note: You can also use the KB, MB, and GB syntax notations.

Syntax Description**OPEN=APPEND | REPLACE | REPLACEOLD**

when a log file already exists, specifies how the contents of the existing file are treated.

APPEND

appends the log when opening an existing file. If the file does not already exist, a new file is created.

REPLACE

overwrites the current contents when opening an existing file. If the file does not already exist, a new file is created.

REPLACEOLD

replaces files that are more than one day old. If the file does not already exist, a new file is created.

Default REPLACE

ROLLOVER=AUTO|NONE|SESSION | n | nK | nM | nG

specifies when or if the SAS log “rolls over”. That is, when the current log is closed and a new one is opened.

AUTO

causes an automatic “rollover” of the log when the directives in the value of the LOG= option change, that is, the current log is closed and a new log file is opened.

Restriction Rollover does not occur more often than once a minute.

Interactions Rollover is triggered by a change in the value of the LOG= option.

The name of the new log file is determined by the value of the LOG= system option. If LOG= does not contain a directive, however, the name would never change, so the log would never roll over, even when ROLLOVER=AUTO.

NONE

specifies that rollover does not occur, even when a change occurs in the name that is specified with the LOG= option.

Interaction If the LOG= value contains any directives, they do not resolve. For example, if Log="#b.log" is specified, the directive “#” does not resolve, and the name of the log file remains "#b.log".

SESSION

at the beginning of each SAS session, opens the log file, resolves directives that are specified in the LOG= system option, and uses its resolved value to name the new log file. During the course of the session, no rollover is performed.

n |nK |nM |nG

causes the log to rollover when the log reaches a specific size, stated in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); or 1,073,741,824 (gigabytes). When the log reaches the specified size, it is closed and renamed by appending “old” to the log filename, and if it exists, the lock file for a server log. For example, a filename of mylog.log would be renamed mylogold.log. A new log file is opened using the name specified in the LOG= option.

Restriction The minimum log file size is 10K.

Interaction When a rollover occurs because of size and the LOG= value contains directives, the directives do not resolve. For example, if Log="#b.log" is specified, the directive “#” does not resolve, and the name of the log file remains "#b.log".

Note If you use ROLLOVER=n to roll over your files, the OPEN= parameter is ignored, and the initial log file is opened with OPEN=APPEND.

CAUTION **Old log files can be overwritten.** SAS maintains only one old log file with the same name as the open log file. If rollover occurs more than once, the old log file is overwritten.

Default NONE

See [“LOG System Option” on page 111](#)

WRITE=BUFFERED | IMMEDIATE

specifies when content is written to the SAS log.

BUFFERED

writes content to the SAS log only when a buffer is full in order to increase efficiency.

IMMEDIATE

writes to the SAS log each time that statements are submitted that produce content for the SAS log. SAS does no buffering of log messages.

Interaction This argument is valid in the windowing environment when SAS is started with the ALTLOG= system option. The contents are written to the file that is specified by the ALTLOG= option and to the SAS log.

Default BUFFERED

Windows specifics The buffered log contents are written periodically, using an interval that is specified by SAS.

Details

The LOGPARM= system option controls the opening and closing of SAS log files when SAS is operating in batch mode or in line mode. This option also controls the naming of new log files, in conjunction with the LOG= system option and the use of directives in the value of LOG=.

Using directives in the value of the LOG= system option enables you to control when logs are open and closed and how they are named, based on actual time events, such as time, month, and day of week.

You can begin directives with either the % symbol or the # symbol, and use both symbols in the same directive. For example, this log specification uses both symbols: -log=mylog%b#C.log.

The following table contains a list of directives that are valid in LOG= values:

Table 2.2 Directives for Controlling the Name of SAS Log Files

Directive	Description	Range
%a or #a	Locale's abbreviated day of week	Sun–Sat
%A or #A	Locale's full day of week	Sunday–Saturday

Directive	Description	Range
%b or #b	Local's abbreviated month	Jan–Dec
%B or #B	Locale's full month	January–December
%C or #C	Century number	00–99
%d or #d	Day of the month	01–31
%H or #H	Hour	00–23
%j or #j	Julian day	001–366
%l or #l *	User name	alphanumeric string that is the name of the user that started SAS
%M or #M	Minutes	00–59
%m or #m	Month number	01–12
%n or #n	Current system node name (without domain name)	none
%p or #p *	Process ID	alphanumeric string that is the SAS session process ID
%P or #P	Sysin filename (without the .sas extension)	none
%s or #s	Seconds	00–59
%u or #u	Day of week	1= Monday–7=Sunday
%v or #v *	Unique identifier	alphanumeric string that creates a log filename that does not currently exist
%w or #w	Day of week	0=Sunday–6=Saturday
%W or #W	Week number (Monday as first day; all days in new year preceding first Monday are in week 00)	00–53
%y or #y	Year without century	00–99
%Y or #Y	Full year	1970–9999
%%	Percent escape writes a single percent sign in the log filename.	%

Directive	Description	Range
##	Pound escape writes a single number sign in the log filename.	#

* Because %v, %l, and %p are not a time-based format, the log filename never changes after it has been generated. Therefore, the log never rolls over. In these situations, specifying ROLLOVER=AUTO is equivalent to specifying ROLLOVER=SESSION.

Note: Directives that you specify in the LOG= system option are not the same as the conversion characters that you specify to format logging facility logs. Directives specify a format for a log name. Conversion characters specify a format for log messages. Directives and conversion characters that use the same characters might function differently.

Example

Rolling over the log at a certain time and using directives to name the log according to the time:

If this command is submitted at 9:43 AM, this example creates a log file called test0943.log, and the log rolls over each time the log filename changes. In this example, at 9:44 AM, the test0943.log file is closed, and the test0944.log file is opened.

```
sas -sysin myprog.sas -log "test%H%M.log" -logparm "rollover=auto"
```

Preventing log rollover but using directives to name the log:

For a SAS session that begins at 9:34 AM, this example creates a log file named test0934.log, and prevents the log file from rolling over:

```
sas -sysin myprog.sas -log "test%H%M.log" -logparm "rollover=session"
```

Preventing log rollover and preventing the resolution of directives:

This example creates a log file named test%H%M.log, ignores the directives, and prevents the log file from rolling over during the session:

```
sas -sysin myprog.sas -log "test%H%M.log" -logparm "rollover=none"
```

Creating log files with unique identifiers:

This example uses a unique identifier to create a log file with a unique name:

```
sas -sysin myprog.sas -log "test%v.log" -logparm "rollover=session"
```

SAS replaces the directive %v with *process_IDvn*, where *process_ID* is a numeric process identifier that is determined by the operating system and *n* is an integer number, starting with 1. The letter v that is between *process_ID* and *n* is always a lowercase letter.

For this example, *process_ID* is 3755. If the file does not already exist, SAS creates a log file with the name test3755v1.log. If test3755v1.log does exist, SAS attempts to create a log file by incrementing *n* by 1, and this process continues until SAS can generate a log file. For example, if the file test3755v1.log exists, SAS attempts to create the file test3755v2.log.

Naming a log file by the user that started SAS:

This example creates a log filename that contains the user name that started the SAS session:

```
sas -sysin myprog.sas -log "%l.log" -logparm "rollover=session";
```

LRECL= System Option

Specifies the default logical record length to use for reading and writing external files.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: External Files
PROC OPTIONS GROUP=	EXTFILES
Defaults:	The shipped default for variable length records (RECFM=V) and print records (RECFM=P) is 32767. The shipped default for fixed length records (RECFM=F) and binary records (RECFM=N) is 256.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

LRECL=*n* | *n*K | *hexX* | MIN | MAX

Note: You can also use the KB syntax notation.

Syntax Description

n

specifies the logical record length in multiples of 1 (bytes) or 1,024 (kilobytes). For example, a value of 32 specifies 32 bytes, and a value of 16k specifies 16,384 bytes.

Range	1–32767
--------------	---------

Restriction	A value of 32K is not valid. It is 1 over the maximum allowed.
--------------------	--

hexX

specifies the logical record length as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by an X. For example, the value 2dx sets the logical record length to 45 characters.

MIN

specifies a logical record length of 1.

MAX

specifies a logical record length of 32,767.

Details

The logical record length for reading or writing external files is first determined by the LRECL= option in the access method statement, function, or command that is used to read or write an individual file. If the logical record length is not specified by any of these means, SAS uses the value that is specified by the LRECL= system option.

Use a value for the LRECL= system option that is not an arbitrary large value. Large values for this option can result in excessive use of memory, which can degrade performance.

LPTYPE System Option

Specifies which Linux command and option settings are used to route files to the printer.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: Procedure Output

PROC OPTIONS GROUP= LISTCONTROL

Default: No default

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

LPTYPE=BSD | SYSV

Required Arguments

BSD

causes SAS to use the **lpr** command to send files to the printer.

SYSV

causes SAS to use the **lp** command to send files to the printer.

Details

The LPTYPE option determines whether SAS is to use the **lpr** or the **lp** Linux command to print files.

If you do not know whether to specify BSD or SYSV, check with your system administrator.

By default, SAS uses the **lpr** command if your operating system is derived from Berkeley's version. Otherwise, it uses the **lp** command.

See Also

System Options:

- [“PRINTCMD System Option” on page 154](#)

MAXMEMQUERY System Option

Specifies the maximum amount of memory that can be allocated per request for certain procedures.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: System Administration: Memory

PROC OPTIONS GROUP= MEMORY

Default: The shipped default is 256M.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

MAXMEMQUERY=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Required Arguments

n* | *nK* | *nM* | *nG* | *nT

specifies the limit in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes), or 1,099,511,627,776 (terabytes). You can specify decimal values for the number of kilobytes, megabytes, gigabytes, or terabytes. For example, a value of **8** specifies 8 bytes, a value of **.782k** specifies 801 bytes, and a value of **3m** specifies 3,145,728 bytes.

hexX

specifies the amount of memory as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by hexadecimal characters (0–9, A–F), and then followed by an X. For example, **2dx** sets the amount of memory to 45 bytes.

MIN

specifies 0 bytes, which indicates that there is no limit on the total amount of memory that can be allocated per request by each SAS procedure. These memory allocations are limited by the value of MEMSIZE.

MAX

specifies a limit to the amount of memory that is allocated. Memory allocations (9,007,199,254,740,992 byte limit on 64-bit machines) are limited by the value of MEMSIZE.

Details

Some SAS procedures use the MAXMEMQUERY option to specify the largest block of virtual memory that a procedure can request at one time. By contrast, the MEMSIZE option places a limit on the total amount of virtual memory that SAS dynamically allocates at any time. This virtual memory is supported by a combination of real memory and paging space. The operating environment begins paging when the amount of virtual memory that is required exceeds the real memory that is available. To prevent paging and the associated performance problems, the MAXMEMQUERY and MEMSIZE system options should be set to a subset of real memory.

MEMSIZE System Option

Specifies the limit on the total amount of virtual memory that can be used by a SAS session.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: System Administration: Memory

PROC OPTIONS MEMORY, PERFORMANCE
GROUP=

Default: The shipped default is 2G.

Restriction: The MEMSIZE option is not valid for the CAS server.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

MEMSIZE *n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Required Arguments

n | *nK* | *nM* | *nG* | *nT*

specifies the limit in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). You can specify decimal values for the number of kilobytes, megabytes, or gigabytes (for example, a value of .25G specifies 268,435,456 bytes).

hexX

specifies the amount of memory as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by hexadecimal characters (0–9, A–F), and then followed by an X. For example, `0F00000x` sets the value of the MEMSIZE option to 15,728,640 bytes. A value of `0x` is equivalent to using the MAX value.

MAX

specifies to set the memory size to the largest reasonable value depending on the amounts of physical memory and paging space that are available when SAS is started.

Details

The Basics

The MEMSIZE system option limits the total amount of memory that is available to each SAS session. It places an enforced limit on the amount of virtual memory that SAS can dynamically allocate at execution. If MEMSIZE is set too low, your jobs can fail, and errors appear in the SAS log indicating that insufficient memory was available. By contrast, the REALMEMSIZE and MAXMEMQUERY system options, the SORTSIZE= option in the SORT procedure, and the SUMSIZE= option in the SUMMARY procedure all provide for procedure tuning.

When you start a SAS session, if the value of MEMSIZE is larger than the amount of virtual memory that is available for a process, then you are notified in the SAS log. If this occurs, adjust the value of MEMSIZE so that it is smaller than the amount of virtual memory or use the MAX value. The MAX value automatically considers both page size and virtual memory limit and adjusts the MEMSIZE value accordingly. You can use the `limit`, `ulimit -a`, or `ulimit -aS` command to see the amount of virtual memory that is available for your user ID.

If you specify an unreasonably small value for MEMSIZE (for example, 6K), then the MEMSIZE value automatically increases to the minimum value that enables SAS to start.

Numeric values in excess of 9,223,372,036,854,775,807 bytes are rejected as invalid and prevent SAS from starting.

SAS does not automatically reserve or allocate the amount of virtual memory that you specify in the MEMSIZE system option. SAS uses only as much memory as it needs to complete a process. For example, a DATA step might require only 20 MB of memory, so even though MEMSIZE is set to 500 MB, SAS uses only 20 MB of memory. While your SAS jobs are running, you can monitor the effects of larger memory settings by using system monitoring tools, such as VMSTAT and the top tool. With some tools, address space might be allocated to memory, but pages might not be assigned to that memory. These tools report a higher value than real memory actually used. When a user invokes third-party software, such as database vendor code that SAS loads, the memory allocations for that third-party software are not controlled by MEMSIZE. Third-party software memory usage can be reported by the top tool.

Setting the Size of MEMSIZE

Setting MEMSIZE=MAX sets MEMSIZE to 80% of physical memory. Setting MEMSIZE to MAX is the same as setting MEMSIZE to 0. Setting MEMSIZE to MAX is reasonable only if no processes that consume large amounts of memory are likely to become active after SAS has started. For example, if multiple instances of SAS are running concurrently, and all of the sessions were started with a MEMSIZE value of MAX, then one or more of these sessions can encounter out-of-memory conditions, or the operating system can run out of available paging space. MEMSIZE=MAX calculates a value that would help prevent the system from paging if all of the memory were allocated.

The optimal setting for this option depends on the other applications that are running and the system resources available at your site. The amount of memory available to SAS processes can also be limited by your system administrator.

If you set MEMSIZE to the maximum amount of memory that is reasonably attainable, some procedures scale themselves to the available memory. To determine the limit on the total amount of memory to be used by SAS, you can issue a PROC OPTIONS statement:

```
proc options option=memsize;
run;
```

Setting MEMSIZE to 0 is used as a test that can determine a good value to set for MEMSIZE.

To determine the optimal setting of MEMSIZE, execute a SAS procedure or DATA step with the FULLSTIMER option and MEMSIZE set to 0. Note the amount of memory that is used by the process, and then set MEMSIZE to a larger amount.

Comparisons

Some SAS procedures use the REALMEMSIZE system option to specify how much real memory the procedure can allocate and use without inducing excessive page swapping. By contrast, the MEMSIZE system option places a limit on the total amount of virtual memory that SAS dynamically allocates at any time. This virtual memory is supported by a combination of real memory and paging space.

The operating environment begins paging when the amount of virtual memory that is required exceeds the real memory that is available. To prevent paging and the associated performance problems, the REALMEMSIZE and MEMSIZE system options should be set to a subset of real memory.

See Also

System Options:

- [“REALMEMSIZE System Option” on page 157](#)

Procedures:

- [“SORT Procedure” in *SAS Viya Visual Data Management and Utility Procedures Guide*](#)

MERGENOBY System Option

Specifies the type of message that is issued when MERGE processing occurs without an associated BY statement.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Files: SAS Files

PROC OPTIONS GROUP= SASFILES

Default: The shipped default is NOWARN.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

MERGENOBY= [NOWARN](#) | [WARN](#) | [ERROR](#)

Syntax Description

NOWARN

specifies that no warning message is issued.

WARN

specifies that a warning message is issued.

ERROR

specifies that an error message is issued.

METACONNECT= System Option

Identifies one named profile from the metadata connection profiles for connecting to a SAS 9 Metadata Server.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Communications: Metadata

PROC OPTIONS GROUP= META

- Default:** Blank-value
- Requirement:** Valid only when the METAPROFILE= option is set
- Notes:** You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see [SAS 9.4 Guide to Metadata-Bound Libraries](#).
This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).
-

Syntax

METACONNECT= "blank-value" | *named-connection* | NONE

Syntax Description

“blank-value”

specifies the first connection profile in the XML document specified in METAPROFILE. This is the default value when the METAPROFILE option is specified and METACONNECT= is not used.

named-connection

is the name of a connection profile in the XML document specified in METAPROFILE. The maximum length is 256 characters. Quotation marks are optional.

NONE

is a special option that disables METAPROFILE processing.

Details

This system option is one of a category of system options that define a connection to the SAS 9 Metadata Server.

METACONNECT= specifies the name of a connection profile in the XML document specified in METAPROFILE. Most XML documents contain only one named connection profile, so METACONNECT= is not required.

METACONNECT=NONE is provided to turn off the re-routing of connections that occurs when the default server connection profile is active in clustered SAS 9 Metadata Server configurations. This option ensures that administrative requests, like stopping one server from the cluster, fail if they cannot be executed on the intended server node.

Example

Here is an example that invokes a user-defined connection profile named Mike's profile:

```
-METAPROFILE "sas-config-path/userprofiles.xml"
-METACONNECT "Mike's profile"
```

See Also

System Options

- [“METAPROTOCOL= System Option” on page 129](#)

METAENCRYPTALG System Option

Specifies the type of encryption to use when communicating with the SAS 9 Metadata Server.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Communications: Metadata
PROC OPTIONS GROUP=	META
Alias:	METAENCRYPTALGORITHM
Default:	SASPROPRIETARY
Notes:	You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see SAS 9.4 Guide to Metadata-Bound Libraries . This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

METAENCRYPTALG *algorithm* | NONE

Syntax Description

algorithm

specifies the algorithm that SAS clients use to communicate with the SAS 9 Metadata Server. The following algorithms can be used:

- RC2
- RC4
- DES
- TripleDES
- SAS Proprietary (alias SAS)
- AES

NONE

Does not specify an encryption algorithm.

Details

Use the METAENCRYPTALG and METAENCRYPTLEVEL system options to define the type and level of encryption that SAS clients use when they communicate with the SAS 9 Metadata Server.

If you specify an encryption algorithm other than SAS Proprietary (alias SAS), you must have SAS/SECURE software. In SAS 9.4, SAS/SECURE is included with Base SAS software. In prior releases of SAS, SAS/SECURE was an add-on product that was licensed separately. This change makes strong encryption available in all deployments (except where prohibited by import restrictions).

For more information about the encryption algorithms, see the [Encryption in SAS Viya: Data in Motion](#).

See Also

System Options

- [“METAENCRYPTLEVEL System Option” on page 125](#)

METAENCRYPTLEVEL System Option

Specifies the level of encryption when communicating with the SAS 9 Metadata Server.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Communications: Metadata

**PROC OPTIONS
GROUP=** META

Default: CREDENTIALS

Notes: You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see [SAS 9.4 Guide to Metadata-Bound Libraries](#).

This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

METAENCRYPTLEVEL [EVERYTHING](#) | [CREDENTIALS](#)

Syntax Description

EVERYTHING

specifies to encrypt all communication with the metadata server.

CREDENTIALS

specifies to encrypt only login credentials. This is the default.

Details

The SAS IOM supports encrypted communication with the SAS 9 Metadata Server. Use the METAENCRYPTLEVEL and METAENCRYPTALG system options to define the level and type of encryption that SAS clients use when they communicate with the metadata server.

See Also

System Options

- [“METAENCRYPTALG System Option” on page 124](#)

METAPASS= System Option

Specifies the password for the SAS 9 Metadata Server.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Communications: Metadata

PROC OPTIONS GROUP= META

Notes: You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see [SAS 9.4 Guide to Metadata-Bound Libraries](#).
The OPTIONS procedure displays passwords in the SAS log as eight Xs, regardless of the actual password length.
This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

METAPASS= *"password"*

Syntax Description

"password"

is the password for the user ID on the SAS 9 Metadata Server. The maximum length is 512 characters. The quotation marks are optional.

Note To specify an encoded password, use the PWENCODE procedure to disguise the text string, and specify the encoded password for METAPASS=. The metadata server decodes the encoded password. For more information, see the PWENCODE procedure in the *SAS Viya Visual Data Management and Utility Procedures Guide*.

Details

This system option is one of a category of system options that define a connection to the metadata server.

When you are running interactively, you can be prompted for connection properties. Prompting occurs when either METASERVER= or METAPORT= are not specified. Prompting also occurs when METAUSER= or METAPASS= are not specified, and a trusted peer or IWA connection is rejected. For information about trusted peer and IWA, see [SAS 9.4 Intelligence Platform: Application Server Administration Guide](#).

Example

To set the default metadata server to use the password sasuser1, port 8561, repository Foundation, metadata server a123.us.company.com, and user ID myuserid, you would use the following options:

```
-METAPASS "sasuser1"
-METAPORT 8561
-METAREPOSITORY "Foundation"
-METASERVER "a123.us.company.com"
-METAUSER "myuserid"
```

See Also

System Options

- “METAPORT= System Option” on page 127
- “METASERVER= System Option” on page 131
- “METAUSER= System Option” on page 133

METAPORT= System Option

Specifies the TCP port for the SAS 9 Metadata Server.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Communications: Metadata
PROC OPTIONS GROUP=	META
Range:	1–65535
Notes:	You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see SAS 9.4 Guide to Metadata-Bound Libraries . This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

METAPORT=*number*

Syntax Description

number

is the TCP port that the SAS 9 Metadata Server is listening to for connections. The default port number that is configured for the metadata server at installation is 8561. Installers are not required to use this value, so you must specify METAPORT= to connect to the metadata server. Do not quote this value.

Details

This system option is one of a category of system options that define a connection to the SAS 9 Metadata Server.

When you are running interactively, you can be prompted for connection values. Prompting occurs when either METASERVER= or METAPORT= are not specified. Prompting also occurs when METAUSER= or METAPASS= are not specified, and a trusted peer or IWA connection is rejected. For information about trusted peer and IWA, see [SAS 9.4 Intelligence Platform: Application Server Administration Guide](#).

Example

Use one of these options to set the default SAS 9 Metadata Server to use the password sasuser1, port 8561, repository Foundation, metadata server a123.us.company.com, and user ID myuserid:

```
-METAPASS "sasuser1"
-METAPORT 8561
-METAREPOSITORY "Foundation"
-METASERVER "a123.us.company.com"
-METAUSER "myuserid"
```

See Also

System Options

- “[METAPASS= System Option](#)” on page 125
- “[METASERVER= System Option](#)” on page 131
- “[METASPN= System Option](#)” on page 132
- “[METAUSER= System Option](#)” on page 133

METAPROFILE System Option

Identifies the XML document that contains connection profiles for the SAS 9 Metadata Server.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Communications: Metadata

**PROC OPTIONS
GROUP=** META

Notes: You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see [SAS 9.4 Guide to Metadata-Bound Libraries](#).

This option can be restricted by a site administrator. For more information, see “[Restricted Options](#)” on page 8.

Syntax

METAPROFILE "*XML-document*"

Syntax Description

"XML-document"

is the pathname of the XML document that contains connection profiles for connecting to the SAS 9 Metadata Server. The pathname is the physical location that is recognized by the operating environment. Quotation marks are required.

Beginning in SAS 9.4, all configurations that include a SAS 9 Metadata Server define a shared file named metadataConfig.xml file in the top-level configuration directory and in every configuration directory that contains a SAS server. This metadataConfig.xml file contains a single connection profile that lists the host and port addresses of the metadata server or metadata server nodes at the time of configuration. The connection profile in the metadataConfig.xml file is referred to as

the “default server connection profile”. Use of the metadataConfig.xml file is recommended for clustered SAS 9 Metadata Server configurations, but it is not required.

Details

This system option is one of a category of system options that define a connection to the SAS 9 Metadata Server. Instead of specifying individual connection options for the metadata server, you can use the METAPROFILE option.

METAPROFILE must be configured before SAS is started or specified at SAS invocation. It specifies the pathname of the XML document that contains one or more connection profiles. A connection profile contains metadata server connection properties, such as the name of the host computer on which the metadata server is invoked and the TCP port. Optional metadata server connection properties are the user ID and password of the requesting user.

SAS uses the first connection profile in the specified XML document to establish the metadata server connection by default. The METACONNECT= system option might be needed if there is more than one connection profile in the XML document. METACONNECT=NONE is provided to disable METAPROFILE processing. For more information, see [METACONNECT=](#).

The default server connection profile in metadataConfig.xml reflects the initial metadata server configuration. When a metadata server node is added to or removed from the cluster, an administrator must run the sas-upgrade-metadata-profile batch utility to update the default server connection profile. For information about the utility, see [SAS 9.4 Intelligent Platform: System Administration Guide](#).

The default server connection profile does not contain user ID and password information. You can provide this information with the METAUSER= and METAPASS= options.

Example

To configure access to the default server connection profile, you could use the following option:

```
-METAPROFILE "C:\sas9-config-path\metadataConfig.xml"
```

See Also

System Options

- [“METACONNECT= System Option” on page 122](#)

METAPROTOCOL= System Option

Specifies the network protocol for connecting to the SAS 9 Metadata Server.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Communications: Metadata
PROC OPTIONS GROUP=	META

Default: BRIDGE

Notes: You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see [SAS 9.4 Guide to Metadata-Bound Libraries](#).
This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

METAPROTOCOL=BRIDGE

Syntax Description

BRIDGE

specifies that the connection to the SAS 9 Metadata Server uses the SAS Bridge protocol. In this release, it is the only supported value and the default value, so there is no need to specify this system option.

Details

This system option is one of a category of system options that define a connection to the SAS 9 Metadata Server.

METAREPOSITORY= System Option

Specifies the SAS 9 Metadata Repository to use with the SAS 9 Metadata Server.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Communications: Metadata

PROC OPTIONS GROUP= META

Default: Foundation

Notes: You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see [SAS 9.4 Guide to Metadata-Bound Libraries](#).
This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

METAREPOSITORY= "name"

Syntax Description

"name"

is the name of the repository to use. The maximum length is 32,000 characters. The quotation marks are optional.

Details

This system option is one of a category of system options that define a connection to the SAS 9 Metadata Server.

See Also

System Options

- “METAPASS= System Option” on page 125
- “METAPORT= System Option” on page 127
- “METASERVER= System Option” on page 131
- “METAUSER= System Option” on page 133

METASERVER= System Option

Specifies the host name or address of the SAS 9 Metadata Server.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Communications: Metadata

PROC OPTIONS GROUP= META

Notes: You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see [SAS 9.4 Guide to Metadata-Bound Libraries](#). This option cannot be restricted by a site administrator. For more information, see “[Restricted Options](#)” on page 8.

Syntax

METASERVER= "*address*"

Syntax Description

"*address*"

is the host name or network IP address of the computer that hosts the SAS 9 Metadata Server. An example is `metaserver="a123.us.company.com"`. The value `localhost` can be used when connecting to a metadata server on the same computer. The maximum length is 256 characters. The quotation marks are optional, unless there are special characters in the address.

Details

This system option is one of a category of system options that define a connection to the SAS 9 Metadata Server.

When you are running interactively, you can be prompted for connection properties. Prompting occurs when either METASERVER= or METAPORT= are not specified. Prompting also occurs when METAUSER= or METAPASS= are not specified, and a trusted peer or IWA connection is rejected. For information about trusted peer and IWA, see [SAS Intelligence Platform: Security Administration Guide](#).

Example

Use these options to set the default metadata server to use the password sasuser1, port 8561, repository Foundation, metadata server a123.us.company.com, and user ID myuserid:

```
-METAPASS "sasuser1"
-METAPORT 8561
-METAREPOSITORY "Foundation"
-METASERVER "a123.us.company.com"
-METAUSER "myuserid"
```

See Also

System Options

- “METAPASS= System Option” on page 125
- “METAPORT= System Option” on page 127
- “METASPN= System Option” on page 132
- “METAUSER= System Option” on page 133

METASPN= System Option

Specifies the service principal name (SPN) for the SAS 9 Metadata Server.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Communications: Metadata
PROC OPTIONS GROUP=	META
Default:	Generated in the form SAS/ <i>machine-name</i>
Notes:	You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see SAS 9.4 Guide to Metadata-Bound Libraries . This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

METASPN=*SPN-name*

Syntax Description

SPN-name

is the SPN for the principal that runs the SAS 9 Metadata Server. The maximum length is 256 characters. The following formats are supported for *SPN-name*: **SAS/***machine-name* or **SAS/***machine-name.company.com*. **SAS** is the name of the service and represents the service type.

Details

Integrated Windows Authentication (IWA) is enabled for a SAS 9 Metadata Server session by specifying the SAS option `-SSPI` in a configuration file or at SAS 9 invocation. When the SAS session is invoked with the `-SSPI` option and a user attempts to connect to the SAS 9 Metadata Server without specifying `METAUSER=` and `METAPASS=`, SAS creates a default SPN and makes an IWA connection.

For more information about the default SPN and special requirements for using IWA on Linux, see [SAS Intelligence Platform: Security Administration Guide](#). If the IWA connection fails, the server attempts to make a trusted peer connection.

The `METASPN` system option is provided for the rare circumstance in which the default SPN cannot be used by SAS. This can occur on Windows systems when SAS servers are not being run under the local system account. The local system account can register the default SPN—a domain user account cannot. For more information about SPNs and SAS servers, see [SAS Intelligence Platform: Security Administration Guide](#).

`METASPN=` is used with `METASERVER=` and `METAPORT=`. If you specify `METAUSER=` and `METAPASS=`, then the `METASPN=` value is *not* used.

Example: Default Form

Here is an example that shows a `METASPN=` value in the default form:

```
-METASERVER "a123.company.com"
-METAPORT 9999
-METASPN "SAS/a123.company.com"
```

See Also

System Options

- [“METASERVER= System Option” on page 131](#)
- [“METAPORT= System Option” on page 127](#)

METAUSER= System Option

Specifies the user ID for connecting to the SAS 9 Metadata Server.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Communications: Metadata
PROC OPTIONS GROUP=	META
Notes:	You use this option to access SAS 9 metadata-bound libraries from SAS Viya. For more information, see SAS 9.4 Guide to Metadata-Bound Libraries . This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

`METAUSER= "userid"`

Syntax Description

"userid"

is the user ID for connecting to the SAS 9 Metadata Server. The maximum length is 256 characters. The quotation marks are optional, unless the user ID includes a special character, such as "sasadm@saspw".

Details

This system option is one of a category of system options that define a connection to the SAS 9 Metadata Server.

When you are running interactively, you can be prompted for connection properties. Prompting occurs when either METASERVER= or METAPORT= are not specified. Prompting also occurs when METAUSER= or METAPASS= are not specified, and a trusted peer or IWA connection is rejected. For information about trusted peer and IWA, see [SAS Intelligence Platform: Security Administration Guide](#).

In a network environment, METAUSER= must specify a fully qualified user ID in the form of SERVERNAME\USERID. For information about user definitions, see [SAS Intelligence Platform: Security Administration Guide](#).

Example

Use these options to set the default metadata server to use the password sasuser1, port 8561, repository Foundation, metadata server a123.us.company.com, and user ID myuserid:

```
-METAPASS "sasuser1"
-METAPORT 8561
-METAREPOSITORY "Foundation"
-METASERVER "a123.us.company.com"
-METAUSER "myuserid"
```

See Also

System Options

- “METAPASS= System Option” on page 125
- “METAPORT= System Option” on page 127
- “METASERVER= System Option” on page 131

MISSING= System Option

Specifies the character to print for missing numeric values.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Log and Procedure Output Control: SAS Log and Procedure Output Log and Procedure Output Control: SAS Log Log and Procedure Output Control: Procedure Output
PROC OPTIONS GROUP=	LOG_LISTCONTROL LISTCONTROL

LOGCONTROL

Default: The shipped default is a period (.).

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

MISSING=<'>*character*<>

Syntax Description

character

specifies the value to be printed. The value can be any character. Single or double quotation marks are optional.

Restriction '00'x is not a valid value for *character*.

Details

The MISSING= system option does not apply to special missing values such as .A and .Z.

MSG System Option

Specifies the library that contains the SAS error messages.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Files

**PROC OPTIONS
GROUP=** ENVFILES

Alias: SASMSG

Default: The shipped default is !SASROOT/sasmsg (set in the installed !SASROOT/sasv9.cfg file).

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

-MSG *pathname*

-MSG ('*pathname*' '*pathname*' ...)

Required Argument

pathname

must resolve to a valid Linux pathname. You can use an environment variable that resolves to a valid pathname.

Details

The MSG system option specifies the library that contains the SAS error messages. This option is set during the installation process and is not normally changed after installation. To add additional pathnames, use the APPEND and INSERT system options.

See Also

System Options:

- [“APPEND= System Option” on page 39](#)
- [“INSERT= System Option” on page 99](#)

MSGCASE System Option

Specifies whether notes, warnings, and error messages that are generated by SAS are displayed in uppercase characters.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: SAS Log
PROC OPTIONS GROUP=	LOGCONTROL
Default:	The shipped default is NOMSGCASE.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

MSGCASE | NOMSGCASE

Required Arguments

MSGCASE

displays notes, warnings, and error messages in uppercase characters.

NOMSGCASE

displays notes, warnings, and error messages in uppercase and lowercase characters.

Details

The MSGCASE system option specifies whether notes, warnings, and error messages that are generated by SAS are displayed in uppercase characters. User-generated messages and source lines are not affected by the MSGCASE system option.

MSGCASE is supported in NL formats. For information about NL formats, see [SAS Viya National Language Support: Reference Guide](#).

MSGLEVEL= System Option

Specifies the level of detail in messages that are written to the SAS log.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: SAS Log
PROC OPTIONS GROUP=	LOGCONTROL
Default:	The shipped default is N.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

MSGLEVEL=N | I

Syntax Description

N

specifies to print notes, warnings, CEDA message, and error messages only.

I

specifies to print additional notes pertaining to index usage, merge processing, sort utilities, data type conversion, execution in CAS, and Hadoop MapReduce jobs along with standard notes, warnings, CEDA messages, and error messages.

Details

Some of the conditions under which the MSGLEVEL= system option applies are as follows:

- If MSGLEVEL=I, SAS writes informative messages to the SAS log about index processing. In general, when a WHERE expression is executed for a data set with indexes, the following information appears in the SAS log:
 - if an index is used, a message appears that indicates the name of the index.
 - if an index is not used but one exists that could optimize at least one condition in the WHERE expression, messages provide suggestions that describe what you can do to influence SAS to use the index. For example, a message could suggest sorting the data set into index order or to specify more buffers.
 - a message displays the IDXWHERE= or IDXNAME= data set option value if the setting can affect index processing.
- If MSGLEVEL=I, SAS writes a warning to the SAS log when a MERGE statement would cause variables to be overwritten.
- If MSGLEVEL=I, SAS writes a message that indicates which sorting product was used.
- If MSGLEVEL=I and a data set is copied to a SAS library whose engine does not support the VARCHAR data type, SAS writes a message to the SAS log for each variable whose data type was converted from VARCHAR to CHAR.

If MSGLEVEL=N, one message is written to the SAS log to indicate that the data type for multiple variables was changed from VARCHAR to CHAR.
- If MSGLEVEL=I, SAS writes messages to the log about why a DATA step did not run in CAS.

MSYMTABMAX System Option

Specifies the maximum amount of memory available to the macro variable symbol tables.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Macro: SAS Macro
PROC OPTIONS GROUP=	MACRO
Default:	4M (set in the installed !SASROOT/sasv9.cfg file)
See:	MSYMTABMAX= System Option in SAS Viya Macro Language: Reference

Syntax

MSYMTABMAX=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Required Arguments

n* | *nK* | *nM* | *nG* | *nT

specifies the maximum amount of memory that is available in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). You can specify decimal values for the number of kilobytes, megabytes, or gigabytes. For example, a value of **8** specifies 8 bytes, a value of **.782k** specifies 801 bytes, and a value of **3m** specifies 3,145,728 bytes.

hexX

specifies the maximum amount of memory that is available as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by hexadecimal characters (0–9, A–F), and then followed by an X. For example, **2dx** sets the maximum amount of memory to 45 bytes.

MIN

sets the amount of memory that is available to the minimum setting, which is 0 bytes. Setting the amount of memory to the minimum setting causes all macro symbol tables to be written to disk.

MAX

sets the amount of memory that is available to the maximum setting. On 64-bit computers, this value is 9,007,199,254,740,992 bytes.

MVARSIZE System Option

Specifies the maximum size for in-memory macro variables.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Macro: SAS Macro
PROC OPTIONS GROUP=	MACRO

Default: The shipped default is 65534.

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

See: MVARSIZE System Option in [SAS Viya Macro Language: Reference](#)

Syntax

MVARSIZE=*n* | *n*K | *n*M | *n*G | *hexX* | MIN | MAX

Note: You can also use the KB, MB, and GB syntax notations.

Required Arguments

***n* | *n*K | *n*M | *n*G**

specifies the maximum macro variable size in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); or 1,073,741,824 (gigabytes). You can specify decimal values for the number of kilobytes, megabytes, or gigabytes. For example, a value of **8** specifies 8 bytes, a value of **.782k** specifies 801 bytes, and a value of **3m** specifies 3,145,728 bytes.

hexX

specifies the maximum macro variable size as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by hexadecimal characters (0–9, A–F), and then followed by an X. For example, **2dx** sets the maximum macro variable size to 45 bytes.

MIN

sets the macro variable size to the minimum setting, which is 0 bytes. Setting the macro variable size to the minimum setting causes all macro variable values to be written to disk.

MAX

sets the macro variable size to the maximum setting, which is 65,534 bytes.

NEWS= System Option

Specifies an external file that contains messages to be written to the SAS log, immediately after the header.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Categories: Environment Control: Files
Log and Procedure Output Control: SAS Log

**PROC OPTIONS
GROUP=** ENVFILES
LOGCONTROL

Default: !SASROOT/misc/base/news (set in the installed !SASROOT/sasv9/cfg file)

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

NEWS=*external-file* | -NONews

Syntax Description**NEWS file-specification**

specifies an external file. This file contains the messages for the SAS log.

NONEWS

specifies that the contents of the NEWS file are not displayed in the SAS log, even if the file exists. This option causes any previous NEWS specifications to be ignored.

Details

The NEWS file can contain information for uses, including news items about SAS.

The contents of the NEWS file are written to the SAS log immediately after the SAS header.

NOTES System Option

Specifies whether notes are written to the SAS log.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: SAS Log

PROC OPTIONS GROUP= LOGCONTROL

Default: The shipped default is NOTES.

Interaction: SAS Studio sets this option to NOTES before and after each code submission. For more information, see [“System Options in SAS Studio” on page 7](#).

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[NOTES](#) | [NONOTES](#)

Syntax Description**NOTES**

specifies that SAS write notes to the SAS log.

NONOTES

specifies that SAS does not write notes to the SAS log. NONOTES does not suppress error and warning messages.

Details

You must specify NOTES for SAS programs that you send to SAS for problem determination and resolution.

NUMBER System Option

Specifies whether to print the page number in the title line of each page of SAS PDF and RTF output.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Log and Procedure Output Control: SAS Log and Procedure Output Log and Procedure Output Control: SAS Log Log and Procedure Output Control: Procedure Output
PROC OPTIONS GROUP=	LOG_LISTCONTROL LISTCONTROL LOGCONTROL
Default:	The shipped default is NUMBER.
Interaction:	SAS Studio sets this option to NUMBER before each code submission. For more information, see “System Options in SAS Studio” on page 7 .
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

NUMBER | **NONNUMBER**

Syntax Description

NUMBER

specifies that SAS print the page number on the first title line of each page of SAS PDF and RTF output.

NONNUMBER

specifies that SAS not print the page number on the first title line of each page of SAS PDF and RTF output.

OBJECTSERVER System Option

Specifies whether SAS is to run as an Integrated Object Model (IOM) server.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Initialization and Operation
PROC OPTIONS GROUP=	EXECMODES
Default:	NOOBJECTSERVER
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

OBJECTSERVER | **NOOBJECTSERVER**

Syntax Description

OBJECTSERVER

specifies to run SAS as an IOM server.

NOOBJECTSERVER

specifies not to run SAS as on IOM server.

Details

An IOM server is a noninteractive SAS session that is run with the OBJECTSERVER system option. The spawner sets OBJECTSERVER for the SAS IOM servers that it invokes. In order to make it easy to specify the command, the server can be started by using a simple command with an option to connect back to the metadata server to obtain additional IOM-specific options.

You can specify the server start-up command in several locations:

- system command line
- script

Here is the general form of the server start-up command:

```
SAS-exec -objectserver other-system-options
```

- *SAS-exec*
is the absolute path to the SAS executable (for example, `/opt/sas/viya/home/SASFoundation/sas`).
- **-objectserver**
launches this SAS session as an IOM server.
- *other-system-options*
are optional, additional system options. System options that are typically used for servers include options such as LOGCONFIGLOC, NOTERMINAL, and NOLOGO.

OBS= System Option

Specifies the observation that is used to determine the last observation to process, or specifies the last record to process.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	9,223,372,036,854,775,807
Interaction:	SAS Studio sets OBS=MAX before each code submission. For more information, see “System Options in SAS Studio” on page 7 .
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

OBS=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Syntax Description

n | nK | nM | nG | nT

specifies a number to indicate when to stop processing, with n being an integer. Using one of the letter notations results in multiplying the integer by a specific value. That is, specifying K (kilo) multiplies the integer by 1,024; M (mega) multiplies by 1,048,576; G (giga) multiplies by 1,073,741,824; or T (tera) multiplies by 1,099,511,627,776. For example, a value of **20** specifies 20 observations or records, whereas a value of **3m** specifies 3,145,728 observations or records.

hexX

specifies a number to indicate when to stop processing as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by an X. For example, the hexadecimal value F8 must be specified as **0F8x** in order to specify the decimal equivalent of 248. The value **2dx** specifies the decimal equivalent of 45.

MIN

sets the number to 0 to indicate when to stop processing.

Interaction If OBS=0 and the NOREPLACE option is in effect, then SAS can still take certain actions because it actually executes each DATA and PROC step in the program, using no observations. For example, SAS executes procedures, such as CONTENTS and DATASETS, that process libraries or SAS data sets. External files are also opened and closed. Therefore, even if you specify OBS=0, when your program writes to an external file with a PUT statement, an end-of-file mark is written, and any existing data in the file is deleted.

MAX

sets the number to indicate when to stop processing to the maximum number of observations or records, up to the largest eight-byte, signed integer, which is $2^{63}-1$, or approximately 9.2 quintillion.

Details

OBS= tells SAS when to stop processing observations or records. To determine when to stop processing, SAS uses the value for OBS= in a formula that includes the value for OBS= and the value for FIRSTOBS=. The formula is

$$(\text{obs} - \text{firstobs}) + 1 = \text{results}$$

For example, if OBS=10 and FIRSTOBS=1 (which is the default for FIRSTOBS=), the result is 10 observations or records, that is $(10 - 1) + 1 = 10$. If OBS=10 and FIRSTOBS=2, the result is nine observations or records, that is, $(10 - 2) + 1 = 9$.

OBS= is valid for all steps during your current SAS session or until you change the setting.

You can also use OBS= to control analysis of SAS data sets in PROC steps.

If SAS is processing a raw data file, OBS= specifies the last line of data to read. SAS counts a line of input data as one observation, even if the raw data for several SAS data set observations is on a single line.

Comparisons

- An OBS= specification from either a data set option or an INFILE statement option takes precedence over the OBS= system option.

- As the OBS= system option specifies an ending point for processing, the FIRSTOBS= system option specifies a starting point. The two options are often used together to define a range of observations to be processed.

Examples

Example 1: Using OBS= to Specify When to Stop Processing Observations

This example illustrates the result of using OBS= to tell SAS when to stop processing observations. This example creates a SAS data set, executes the OPTIONS statement by specifying FIRSTOBS=2 and OBS=12, and executes the PRINT procedure. The result is 11 observations, that is, $(12 - 2) + 1 = 11$. The result of OBS= in this situation appears to be the observation number that SAS processes last, because the output starts with observation 2, and ends with observation 12, but this result is only a coincidence.

```
data Ages;
  input Name $ Age;
  datalines;
Miguel 53
Brad 27
Willie 69
Marc 50
Sylvia 40
Arun 25
Gary 40
Becky 51
Alma 39
Tom 62
Kris 66
Paul 60
Randy 43
Barbara 52
Virginia 72
run;

options firstobs=2 obs=12;
proc print data=Ages;
run;
```

Output 2.1 PROC PRINT Output Using OBS= and FIRSTOBS=

Obs	Name	Age
2	Brad	27
3	Willie	69
4	Marc	50
5	Sylvia	40
6	Arun	25
7	Gary	40
8	Becky	51
9	Alma	39
10	Tom	62
11	Kris	66
12	Paul	60

Example 2: Using OBS= with WHERE Processing

This example illustrates the result of using OBS= along with WHERE processing. The example uses the data set that was created in Example 1, which contains 15 observations, and the example assumes a new SAS session with the defaults FIRSTOBS=1 and OBS=MAX.

First, here is the PRINT procedure with a WHERE statement. The subset of the data results in 12 observations:

```
proc print data=Ages;
  where Age LT 65;
run;
```

Output 2.2 PROC PRINT Output Using a WHERE Statement

Obs	Name	Age
1	Miguel	53
2	Brad	27
4	Marc	50
5	Sylvia	40
6	Arun	25
7	Gary	40
8	Becky	51
9	Alma	39
10	Tom	62
12	Paul	60
13	Randy	43
14	Barbara	52

Executing the OPTIONS statement with OBS=10 and the PRINT procedure with the WHERE statement results in 10 observations, that is, $(10 - 1) + 1 = 10$. Note that with WHERE processing, SAS first subsets the data and then SAS applies OBS= to the subset.

```
options obs=10;
proc print data=Ages;
  where Age LT 65;
run;
```

Output 2.3 PROC PRINT Output Using a WHERE Statement and OBS=

Obs	Name	Age
1	Miguel	53
2	Brad	27
4	Marc	50
5	Sylvia	40
6	Arun	25
7	Gary	40
8	Becky	51
9	Alma	39
10	Tom	62
12	Paul	60

The result of OBS= appears to be how many observations to process, because the output consists of 10 observations, ending with the observation number 12. However, the result is only a coincidence. If you apply FIRSTOBS=2 and OBS=10 to the subset, the result is nine observations, that is, $(10 - 2) + 1 = 9$. OBS= in this situation is neither the observation number to end with nor how many observations to process; the value is used in the formula to determine when to stop processing.

```
options firstobs=2 obs=10;
proc print data=Ages;
  where Age LT 65;
run;
```


Output 2.4 PROC PRINT Output Using WHERE Statement, OBS=, and FIRSTOBS=

Obs	Name	Age
2	Brad	27
4	Marc	50
5	Sylvia	40
6	Arun	25
7	Gary	40
8	Becky	51
9	Alma	39
10	Tom	62
12	Paul	60

Example 3: Using OBS= When Observations Are Deleted

This example illustrates the result of using OBS= for a data set that has deleted observations. The example uses the data set that was created in Example 1, with observation 6 deleted. The example also assumes a new SAS session with the defaults FIRSTOBS=1 and OBS=MAX.

First, here is PROC PRINT output of the modified file:

```
options firstobs=1 obs=max nodate pageno=1;

proc print data=Ages;
run;
```

Output 2.5 PROC PRINT Output Showing Observation 6 Deleted

Obs	Name	Age
1	Miguel	53
2	Brad	27
3	Willie	69
4	Marc	50
5	Sylvia	40
7	Gary	40
8	Becky	51
9	Alma	39
10	Tom	62
11	Kris	66
12	Paul	60
13	Randy	43
14	Barbara	52
15	Virginia	72

Executing the OPTIONS statement with OBS=12, then the PRINT procedure, results in 12 observations, that is, $(12 - 1) + 1 = 12$:

```
options obs=12;
proc print data=Ages;
run;
```

Output 2.6 PROC PRINT Output Using OBS=

Obs	Name	Age
1	Miguel	53
2	Brad	27
3	Willie	69
4	Marc	50
5	Sylvia	40
7	Gary	40
8	Becky	51
9	Alma	39
10	Tom	62
11	Kris	66
12	Paul	60
13	Randy	43

The result of OBS= appears to be how many observations to process, because the output consists of 12 observations, ending with the observation number 13. However, if you apply FIRSTOBS=2 and OBS=12, the result is 11 observations, that is $(12 - 2) + 1 = 11$. OBS= in this situation is neither the observation number to end with nor how many observations to process; the value is used in the formula to determine when to stop processing.

```
options firstobs=2 obs=12;
proc print data=Ages;
run;
```

Output 2.7 PROC PRINT Output Using OBS= and FIRSTOBS=

Obs	Name	Age
2	Brad	27
3	Willie	69
4	Marc	50
5	Sylvia	40
7	Gary	40
8	Becky	51
9	Alma	39
10	Tom	62
11	Kris	66
12	Paul	60
13	Randy	43

See Also

Data Set Options:

- “FIRSTOBS= Data Set Option” in *SAS Viya Data Set Options: Reference*
- “OBS= Data Set Option” in *SAS Viya Data Set Options: Reference*
- “REPLACE= Data Set Option” in *SAS Viya Data Set Options: Reference*

System Options:

- “FIRSTOBS= System Option” on page 86

OPLIST System Option

Specifies whether the settings of the SAS system options are written to the SAS log.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: SAS Log

PROC OPTIONS GROUP= LOGCONTROL

Default: The shipped default is NOOPLIST.

Note: This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

OPLIST | NOOPLIST

Syntax Description

OPLIST

specifies to write SAS system option settings to the SAS log.

NOOPTLIST

specifies not to write SAS system options to the SAS log.

Details

The OPLIST system option echoes only the system options specified on the command line. It does not echo any system options specified in the configuration file or in the SASV9_OPTIONS environment variable. (If you want to echo the contents of the configuration file, use the VERBOSE option.) For example, invoke SAS with the following command:

```
sas -fullstimer -nonews -oplist
```

SAS writes this line to the SAS log:

```
NOTE: SAS command line: -fullstimer -nonews -oplist
```

Password values that are provided for system options (such as PDFOPENPW) are automatically masked in the SAS log. For example, suppose that you invoke SAS with the following command:

```
sas -oplist -pdfopenpw foobar -stimer
```

The command is displayed in the log as follows:

```
NOTE: SAS command line:
```

```
    /opt/sas/viya/home/SASFoundation/sas -oplist -pdfopenpw XXXXXXXX -stimer
```

See Also

System Options:

- [“VERBOSE System Option” on page 223](#)

OVP System Option

Specifies whether overprinting of error messages to make them bold, is enabled.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: SAS Log
PROC OPTIONS GROUP=	LOGCONTROL
Default:	The shipped default is NOOVP.
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

OVP | **NOOVP**

Syntax Description

OVP

specifies that overprinting of error messages is enabled.

NOOVP

specifies that overprinting of error messages is disabled.

Details

When OVP is specified, error messages are emphasized when SAS overprints the error message two additional times with overprint characters.

PAGENO= System Option

Resets the SAS output page number.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: Procedure Output

PROC OPTIONS GROUP= LISTCONTROL

Default: The shipped default is 1.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

PAGENO=*n* | *nK* | *nM* | *nG* | *hexX* | **MIN** | **MAX**

Syntax Description

n | *nK* | *nM* | *nG*

specifies the page number in multiples of 1: *n*; 1,024 (kilo); 1,048,576 (mega); 1,073,741,824 (giga). For example, a value of **8** sets the page number to 8 and a value of **3k** sets the page number to 3,072.

hexX

specifies the page number as a hexadecimal number. You must specify the value beginning with a number (0-9), followed by an X. For example, the value **2dx** sets the page number to 45.

MIN

sets the page number to the minimum number, 1.

MAX

specifies the maximum page number as the largest signed, four-byte integer that is representable in your operating environment.

Details

The PAGENO= system option specifies a beginning page number for the next page of output that SAS produces. Use PAGENO= to reset page numbering during a SAS session.

PAGESIZE= System Option

Specifies the number of lines that compose a page of the SAS log.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Log and Procedure Output Control: SAS Log and Procedure Output Log and Procedure Output Control: SAS Log Log and Procedure Output Control: Procedure Output
PROC OPTIONS GROUP=	LOG_LISTCONTROL LISTCONTROL LOGCONTROL
Alias:	PS=
Default:	60 for batch mode
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

PAGESIZE=*n* | *nK* | *hexX* | MIN | MAX

Syntax Description

n | *nK*

specifies the number of lines that compose a page in terms of lines (*n*) or units of 1,024 lines (*nK*). You can specify decimal values for the number of kilobytes. For example, a value of **800** specifies 800 lines, a value of **.782k** specifies 801 lines, and a value of **3k** specifies 3,072 lines.

hexX

specifies the number of lines that compose a page as a hexadecimal number. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **2dx** sets the number of lines that compose a page to 45 lines.

MIN

sets the number of lines that compose a page to the minimum setting, 15.

MAX

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

Details

The PAGESIZE= system option affects the SAS log in batch and non-interactive modes.

PARAM= System Option

Specifies a parameter string that is passed to an external program.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
------------------	--

Category: Environment Control: Files
PROC OPTIONS GROUP= ENVFILES

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

PARAM=<>*string*<>

Syntax Description

string

specifies a character string that contains a parameter.

Example

This statement passes the parameter X=2 to an external program:

```
options parm='x=2';
```

PARMCARDS= System Option

Specifies the file reference to open when SAS encounters the PARMCARDS statement in a procedure.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Files

PROC OPTIONS GROUP= ENVFILES

Default: The shipped default is FT15F001.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

PARMCARDS=*file-ref*

Syntax Description

file-ref

specifies the file reference to open.

Details

The PARMCARDS= system option specifies the file reference of a file that SAS opens when it encounters a PARMCARDS (or PARMCARDS4) statement in a procedure.

SAS writes all data lines after the PARMCARDS (or PARMCARDS4) statement to the file until it encounters a delimiter line of either one or four semicolons. The file is then

closed and made available to the procedure to read. There is no parsing or macro expansion of the data lines.

PRINT System Option

Specifies a destination for SAS output when running in batch mode.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Environment Control: Files
PROC OPTIONS GROUP=	ENVFILES
Default:	the SAS output from a batch SAS program is written to a file in the current directory with the same filename as the SAS source file, with an extension of .lst

Syntax

PRINT *file-specification* | **NOPRINT**

Required Arguments

PRINT *file-specification*

specifies the location for the SAS procedure output file. The *file-specification* can be any valid Linux path to a directory, a filename, or an environment variable that is associated with a path. If you specify only the path to a directory, the procedure output file is created in the specified directory. The default name for this file is *filename.lst*, where *filename* is the name of your SAS job.

NOPRINT

suppresses the creation of the SAS procedure output file.

Details

The PRINT system option specifies a destination for SAS output when running in batch mode. The PRINT system option is valid in batch mode; it is ignored in interactive modes.

PRINTCMD System Option

Specifies the print command that SAS uses.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: Procedure Output
PROC OPTIONS GROUP=	LISTCONTROL
Default:	None
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

PRINTCMD="*print-command*"

Required Argument

print-command

specifies the options that you can use with PRINTCMD.

Details

The syntax of the options passed to the print command is controlled by the LPTYPE system option. If LPTYPE is set to BSD, the command uses **lpr** command options. If LPTYPE is set to SYSV, the command uses **lp** command options.

If your site uses a print command (spooler) other than **lp** or **lpr**, *print-command* specifies its name. The PRINTCMD option overrides the LPTYPE setting.

When specified in an OPTIONS statement, the PRINTCMD option does not change the print commands assigned to previously defined filenames. For example, consider the following code:

```
filename pc1 printer;
proc printto print=pc1;
run;
proc print data=sales.week;
run;
```

```
options printcmd="netlp";
```

```
filename pc2 printer;
proc printto print=pc2;
run;
proc print data=sales.month;
run;
```

Output associated with PC2 uses the **netlp** command; output associated with PC1 uses the default print command.

See Also

System Options:

- “LPTYPE System Option” on page 118

PRINTMSGLIST System Option

Specifies whether to print all messages to the SAS log or to print only top-level messages to the SAS log.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: SAS Log
PROC OPTIONS GROUP=	LOGCONTROL
Default:	The shipped default is PRINTMSGLIST.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[PRINTMSGLIST](#) | [NOPRINTMSGLIST](#)

Syntax Description

PRINTMSGLIST

specifies to print the entire list of messages to the SAS log.

NOPRINTMSGLIST

specifies to print only the top-level message to the SAS log.

QUOTELENMAX System Option

If a quoted string exceeds the maximum length allowed, specifies whether SAS writes a note to the SAS log.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Default: The shipped default is QUOTELENMAX.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[QUOTELENMAX](#) | [NOQUOTELENMAX](#)

Syntax Description

QUOTELENMAX

specifies that SAS write a note to the SAS log about the maximum length for strings in quotation marks.

NOQUOTELENMAX

specifies that SAS does not write a note to the SAS log about the maximum length for strings in quotation marks.

Details

If a string in quotation marks is too long, SAS writes the following note to the SAS log:

```
Note: The quoted string currently being processed has become
      more than 262 characters long. You may have unbalanced
      quotation marks.
```

If you are running a program that has long strings in quotation marks, and you do not want to see this note, use the NOQUOTELENMAX system option.

REALMEMSIZE System Option

Specifies the amount of real (physical) memory SAS can expect to allocate.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	System Administration: Memory
PROC OPTIONS GROUP=	MEMORY
Default:	The shipped default is 0.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

REALMEMSIZE *n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Required Arguments

n* | *nK* | *nM* | *nG* | *nT

specifies the amount of memory to reserve in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). The value of *n* can be a decimal value. For example, a value of **8** specifies 8 bytes, a value of **.782k** specifies 801 bytes, and a value of **3m** specifies 3,145,728 bytes.

hexX

specifies the amount of memory as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by hexadecimal characters (0–9, A–F), and then followed by an X. For example, the value **2dx** sets the amount of memory to 45 bytes.

MIN

specifies a value of 0, which indicates that the memory usage is determined by SAS when SAS starts.

MAX

specifies to set the memory size to the largest permissible value. This value depends on the system limit.

Details

The Basics

The REALMEMSIZE system option sets a recommended upper limit on real memory for procedures that can use both real memory and utility disk space, such as PROC SORT. This upper limit helps avoid virtual memory thrashing.

The REALMEMSIZE option should never be set above the amount of real memory. If the amount of real memory is insufficient for a job to run, then setting the MEMSIZE option above the amount of real memory might enable the job to run using a combination of real and virtual memory.

Comparisons

Some SAS procedures use the REALMEMSIZE system option to specify how much real memory the procedure can allocate and use without inducing excessive page swapping. By contrast, the MEMSIZE system option places a limit on the total amount of virtual memory that SAS dynamically allocates at any time. This virtual memory is supported by a combination of real memory and paging space.

The operating environment begins paging when the amount of virtual memory that is required exceeds the real memory that is available. To prevent paging and the associated performance problems, the REALMEMSIZE and MEMSIZE system options should be set to a subset of real memory.

See Also

System Options:

- “MEMSIZE System Option” on page 119

Procedures:

- “SORT Procedure” in *SAS Viya Visual Data Management and Utility Procedures Guide*

REPLACE System Option

Specifies whether permanently stored SAS data sets can be replaced.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is REPLACE.
Notes:	This option can be restricted by a site administrator. For more information, see “ Restricted Options ” on page 8. When you use the OUT2= <i>PermanentLibrary</i> <code>._ALL_</code> option within PROC CONTENTS, you must also set the REPLACE system option or the REPLACE=YES data set option.

Syntax

REPLACE | **NOREPLACE**

Syntax Description

REPLACE

specifies that a permanently stored SAS data set can be replaced with another SAS data set of the same name.

NOREPLACE

specifies that a permanently stored SAS data set cannot be replaced with another SAS data set of the same name, which prevents the accidental replacement of existing SAS data sets.

Details

This option has no effect on data sets in the WORK library, even if you use the WORKTERM= system option to store the WORK library files permanently.

Comparisons

The REPLACE= data set option overrides the REPLACE system option.

See Also**System Options:**

- [“WORKTERM System Option” on page 231](#)

REUSE= System Option

Specifies whether SAS reuses space when observations are added to a compressed SAS data set.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Files: SAS Files

PROC OPTIONS GROUP= SASFILES

Default: The shipped default is NO.

Interaction: The REUSE= data set option overrides the REUSE= system option.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

REUSE=YES | NO

Syntax Description**YES**

specifies to track free space and reuses it whenever observations are added to an existing compressed data set.

Interactions REUSE=YES takes precedence over the POINTOBS=YES data set option setting.

When using COMPRESS=YES and REUSE=YES system options settings, observations cannot be addressed by observation number.

NO

specifies not to track free space.

Details

If space is reused, observations that are added to the SAS data set are inserted wherever enough free space exists, instead of at the end of the SAS data set.

Specifying REUSE=NO results in less efficient usage of space if you delete or update many observations in a SAS data set. However, the APPEND procedure and other procedures that add observations to the SAS data set continue to add observations to the end of the data set, as they do for uncompressed SAS data sets.

You cannot change the REUSE= attribute of a compressed SAS data set after it is created. Space is tracked and reused in the compressed SAS data set according to the REUSE= value that was specified when the SAS data set was created, not when you add and delete observations. Even with REUSE=YES, the APPEND procedure adds observations at the end.

See Also

Data Set Options:

- “COMPRESS= Data Set Option” in *SAS Viya Data Set Options: Reference*
- “REUSE= Data Set Option” in *SAS Viya Data Set Options: Reference*

System Options:

- “COMPRESS= System Option” on page 59

RSASUSER System Option

Specifies whether to open the Sasuser library for Read access or Read-Write access in Batch mode.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Files

PROC OPTIONS GROUP= ENVFILES

Default: The shipped default is NORSASUSER.

Note: This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

RSASUSER | NORSASUSER

Syntax Description

RSASUSER

opens the SASUSER library in Read-Only mode.

NORSASUSER

opens the SASUSER library in Read-Write mode.

Details

If the Sasuser library is being shared by multiple users or the same user is running SAS multiple times simultaneously, the Sasuser library is often shared. By default, if one user has a member of the Sasuser library open for update, all other users are denied access to that SAS library member. For example, if one user is writing to the Sasuser.Profile catalog, no other user can even read data from the Profile catalog.

Specifying RSASUSER enables a group of users to share Sasuser library members by allowing all users Read-Only access to members. In the Profile catalog example, if RSASUSER is in effect, all users can open the Profile catalog for Read-Only access, allowing other users to concurrently read from the Profile catalog. However, no user can write information out to the Profile catalog; you receive an error message if you try to do so.

Specifying RSASUSER from the command line affects only that session's access to files. To enable a group of users to share members in the Sasuser library, the system manager should set RSASUSER in a common SAS configuration file. This configuration file should be shared by all users that share the Sasuser library.

If you specify RSASUSER but no Profile catalog exists in the Sasuser library, the Profile catalog is created in the Work library.

Note: The RSASUSER option is extremely useful for sharing information (such as the Profile catalog) stored in the Sasuser library. It is less practical when used in conjunction with SAS modules that require Update access to the Sasuser library.

RTRACE System Option

Produces a list of resources that are read or loaded during a SAS session.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: SAS Log
PROC OPTIONS GROUP=	LOGCONTROL
Default:	None

Syntax

RTRACE ALL | NONE | VER

Required Arguments

ALL

produces a list of resources that are read or loaded during a SAS session.

NONE

turns off RTRACE on all files.

VER

writes the version number and other trace information for each module that SAS loads.

Details

The RTRACE system option produces a list of resources that are read or loaded during the execution of SAS. If you specify -RTRACE ALL but do not specify the RTRACELOC system option, the output is written to the SAS log.

See Also

System Options:

- [“RTRACELOC System Option” on page 162](#)

RTRACELOC System Option

Specifies the pathname of the file to which the list of resources that are read or loaded during a SAS session is written.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Environment Control: Files
PROC OPTIONS GROUP=	ENVFILES
Default:	None
Tip:	You can expand the RTRACELOC filename when %p (PID), %d (date), or %t (time) are specified.

Syntax

RTRACELOC=*pathname*

Required Argument

pathname

specifies the file to which RTRACE information is written. The *pathname* must include the path and the filename for the RTRACE output.

You can expand the output filename to include the process ID, date, or time in the filename by specifying %p, %d, or %t, respectively. The system date is included in the filename in the YYYYMMDD (year, month, day) format. The time is included in the filename formatted as HHMMSSmmm (hours, minutes, seconds, milliseconds).

For example, to include the date, time, and process ID in the filename, you can specify the following options:

```
-rtrace all -rtraceloc mytrace.%d.%t.%p
```

This results in a filename similar to mytrace.20140306.125510942.3808.

Details

The RTRACELOC system option specifies the pathname of the file to which RTRACE information is written. If the *pathname* does not include a filename, the output is directed to standard output. If you specify -RTRACE ALL, but do not specify the RTRACELOC system option, the output is written to the SAS log.

See Also

System Options:

- [“RTRACE System Option” on page 161](#)

S= System Option

Specifies the length of statements on each line of a source statement and the length of data on lines that follow a DATALINES statement.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Input Control: Data Processing
PROC OPTIONS GROUP=	INPUTCONTROL
Default:	The shipped default is 0.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

S=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Syntax Description

n* | *nK* | *nM* | *nG* | *nT

specifies the length of statements and data in terms of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). For example, a value of **8** specifies 8 bytes, and a value of **3m** specifies 3,145,728 bytes.

hexX

specifies the length of statements and data as a hexadecimal number. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **2dx** sets the length of statements and data to 45.

MIN

sets the length of statements and data to 0.

MAX

sets the length of statements and data to 9,007,199,254,740,992.

Details

Input can be from either fixed-length or variable-length records. Both fixed-length and variable-length records can be sequenced or unsequenced. The location of the sequence numbers is determined by whether the file record format is fixed-length or variable-length.

SAS uses the value of S to determine whether to look for sequence numbers in the input, and to determine how to read the input:

Record Type	Value of S	SAS Looks for Sequence Numbers	How SAS Reads The Input
Fixed-length	S>0 or S=MAX	No	The value of S is used as the length of the source or data to be scanned and ignores everything beyond that length on each line.
Fixed-length	S=0 or S=MIN	Yes, at the end of the line of input.	<p>SAS inspects the last <i>n</i> columns (where <i>n</i> is the value of the SEQ= system option) of the first sequence field.</p> <p>If those columns contain numbers, they are assumed to be sequence numbers and SAS ignores the last eight columns of each line.</p> <p>If the <i>n</i> columns contain non-digit characters, SAS reads the last eight columns as data columns.</p>
Variable-length	S>0 or S=MAX	No	The value of S is used as the starting column of the source or data to be scanned and ignores everything before that length on each line.

Record Type	Value of S	SAS Looks for Sequence Numbers	How SAS Reads The Input
Variable-length	S=0 or S=MIN	Yes, at the beginning of each line of input.	<p>SAS inspects the last n columns (where n is the value of the SEQ= system option) of the first sequence field.</p> <p>If those columns contain numbers, they are assumed to be sequence numbers and SAS ignores the first eight columns of each line.</p> <p>If the n columns contain non-digit characters, SAS reads the first eight columns as data columns.</p>

Comparisons

The S= system option operates exactly like the S2= system option except that S2= controls input only from a %INCLUDE statement, an autoexec file, or an autocall macro file.

See Also

System Options:

- [“S2= System Option” on page 165](#)
- [“S2V= System Option” on page 168](#)
- [“SEQ= System Option” on page 173](#)

S2= System Option

Specifies the length of statements on each line of a source statement from a %INCLUDE statement, an AUTOEXEC= file, or an autocall macro file.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Input Control: Data Processing

PROC OPTIONS GROUP= INPUTCONTROL

Default: The shipped default is 0.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

S2=S | n | nK | nM | nG | nT | hexX | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Syntax Description

S

uses the current value of the S= system option to compute the record length of text that comes from a %INCLUDE statement, an AUTOEXEC= file, or an autocall macro file.

n | nK | nM | nG | nT

specifies the length of the statements in a file that is specified in a %INCLUDE statement, an autoexec file, or an autocall macro file, in terms of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). For example, a value of **8** specifies 8 bytes, and a value of **3m** specifies 3,145,728 bytes.

hexX

specifies the length of statements as a hexadecimal number. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **2dx** sets the length of statements to 45.

MIN

sets the length of statements and data to 0.

MAX

sets the length of statements and data to 9,007,199,254,740,992.

Details

Input can be from either fixed-length or variable-length records. Both fixed-length and variable-length records can be sequenced or unsequenced. The location of the sequence numbers is determined by whether the file record format is fixed-length or variable-length.

SAS uses the value of S2 to determine whether to look for sequence numbers in the input, and to determine how to read the input:

Record Type	Value of S2	SAS Looks for Sequence Numbers	How SAS Reads The Input
Fixed-length	S2>0 or S2=MAX	No	The value of S2 is used as the length of the source or data to be scanned and ignores everything beyond that length on each line.

Record Type	Value of S2	SAS Looks for Sequence Numbers	How SAS Reads The Input
Fixed-length	S2=0 or S2=MIN	Yes, at the end of the line of input.	<p>SAS inspects the last n columns (where n is the value of the SEQ= system option) of the first sequence field.</p> <p>If those columns contain numbers, they are assumed to be sequence numbers and SAS ignores the last eight columns of each line.</p> <p>If the n columns contain non-digit characters, SAS reads the last eight columns as data columns.</p>
Variable-length	S2>0 or S2=MAX	No	<p>The value of S2 is used as the starting column of the source or data to be scanned and ignores everything before that length on each line.</p>
Variable-length	S2=0 or S2=MIN	Yes, at the beginning of each line of input.	<p>SAS inspects the last n columns (where n is the value of the SEQ= system option) of the first sequence field.</p> <p>If those columns contain numbers, they are assumed to be sequence numbers and SAS ignores the first eight columns of each line.</p> <p>If the n columns contain non-digit characters, SAS reads the first eight columns as data columns.</p>

Comparisons

The S2= system option operates exactly like the S= system option except that the S2= option controls input from a %INCLUDE statement, an autoexec file, or an autocall macro file.

The S2= system option reads both fixed-length and variable-length record formats from a file specified in a %INCLUDE statement, an autoexec file, or an autocall macro file. The S2V= system option reads only a variable-length record format from a file specified in a %INCLUDE statement, an autoexec file, or an autocall macro file.

See Also

System Options:

- [“S= System Option” on page 163](#)
- [“S2V= System Option” on page 168](#)
- [“SEQ= System Option” on page 173](#)

S2V= System Option

Specifies the starting position to begin reading a file that is specified in a %INCLUDE statement, an autoexec file, or an autocall macro file with a variable length record format.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Input Control: Data Processing

PROC OPTIONS GROUP= INPUTCONTROL

Default: The shipped default is 0.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

S2V=S2 | S | n | nK | nM | nG | nT | MIN | MAX | hexX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Syntax Description

S2

specifies to use the current value of the S2= system option to compute the starting position of the variable-sized record to read from a %INCLUDE statement, an autoexec file, or an autocall macro file. This is the default. The S2= option has a default value of 0.

S

specifies to use the current value of the S= system option to compute the starting position of the variable-sized record to read from a %INCLUDE statement, an autoexec file, or an autocall macro file.

n* | *nK* | *nM* | *nG* | *nT

specifies the starting position of the variable-length record to read that comes from a %INCLUDE statement, an autoexec file, or an autocall macro file, in terms of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). For example, a value of **8** specifies 8 bytes, and a value of **3m** specifies 3,145,728 bytes.

MIN

sets the starting position of the variable-length record to read that comes from a %INCLUDE statement, an autoexec file, or an autocall macro, to 0.

MAX

sets the starting position of the variable-length record to read that comes from a %INCLUDE statement, an autoexec file, or an autocall macro, to 9,007,199,254,740,992.

hexX

specifies the starting position of the variable-length record to read that comes from a %INCLUDE statement, an autoexec file, or an autocall macro, as a hexadecimal number. You must specify the value beginning with a number (0–9), followed by an X.

Details

Both the S2V= system option and the S2= system option specify the starting position for reading variable-sized record input from a %INCLUDE statement, an autoexec file, or an autocall macro file. When values for both options are specified, the value of the S2V= system option takes precedence over the value specified for the S2= system option.

Comparisons

The S2= system option specifies the starting position for reading both fixed-length and variable-length record formats for input from a %INCLUDE statement, an autoexec file, or an autocall macro file. The S2V= system option specifies the starting position for reading only variable-length record formats for input from a %INCLUDE statement, an autoexec file, or an autocall macro file.

See Also**System Options:**

- [“S= System Option” on page 163](#)
- [“S2= System Option” on page 165](#)
- [“SEQ= System Option” on page 173](#)

SASHELP= System Option

Specifies the location of the Sashelp library.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Files

PROC OPTIONS GROUP= ENVFILES

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Tip: You can use the APPEND or INSERT system options to add additional *library-specifications*.

Syntax

SASHELP=*library-specification*

Syntax Description

library-specification

identifies an external library. *library-specification* can be an environment variable.

Details

To add additional directory specifications, use the INSERT or APPEND system option.

See Also

System Options:

- [“APPEND= System Option” on page 39](#)
- [“INSERT= System Option” on page 99](#)

SASAUTOS System Option

Specifies the autocall library.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Categories: Environment Control: Files
Macro: SAS Macro

PROC OPTIONS GROUP= ENVFILES
MACRO

Default: The shipped default is the SASAUTOS fileref

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

See: [“SASAUTOS= System Option” in SAS Viya Macro Language: Reference](#)

Syntax

SASAUTOS=*'directory-specification'* | *fileref*

SASAUTOS =(*'directory-specification-1'* | *fileref-1*, ..., *'directory-specification-n'* | *fileref-n*)

NOSASAUTOS

Required Arguments

directory-specification

specifies a pathname to an autocall macro library.

fileref

specifies a name (shorthand reference) that has been assigned to an autocall macro library.

Note that the SASAUTOS option uses filerefs, not librefs.

Details

Each autocall macro library consists of files in a Linux directory. The *directory-specification* can be the pathname of a Linux directory, a fileref, or an environment variable.

If you specify the pathname of a directory, you must enclose the name in quotation marks. You can omit the quotation marks only if you are specifying the option in the configuration file, in the SAS command, or in the SASV9_OPTIONS environment variable, and if the name cannot be taken to be a fileref.

How you specify multiple directory names, filerefs, or environment variables depends on where you specify the SASAUTOS option:

- If you specify the SASAUTOS option in the configuration file or in the SASV9_OPTIONS environment variable, use either multiple SASAUTOS options, or enclose the directory names in parentheses. Separate the names with a comma or a blank space.
- If you specify the SASAUTOS option in the SAS command, use the APPEND or INSERT system options to append to the end or insert at the beginning of the current SASAUTOS value. For example, the following code adds `/users/userid/also` to the end of the current SASAUTOS value, `/users/userid/here`:

```
sas -sasautos /users/userid/here -append sasautos /users/userid/also
```

For more information, see [“APPEND= System Option” on page 39](#), and [“INSERT= System Option” on page 99](#).

- If you specify the SASAUTOS option in the OPTIONS statement, you must enclose the directory names in parentheses. Separate the names with a comma or a blank space.

At configuration time, SAS concatenates all directories specified for SASAUTOS. However, after the session starts, any new directories that you specify override any current autocall libraries.

The NOSASAUTOS option causes SAS to ignore all previous SASAUTOS specifications (whether specified in the SAS command, in the configuration file, or in the SASV9_OPTIONS environment variable).

The default value of the SASAUTOS option is the SASAUTOS fileref. There is no Linux directory assigned to the fileref, so you must define the SASAUTOS fileref if you want to use it as your autocall library.

Examples

Example 1: Specifying Multiple Environment Variables in the *OPTIONS* Statement

The following example shows the syntax to use if you are specifying multiple environment variables in the *OPTIONS* statement:

```
options sasautos=(AUTODIR, SASAUTOS);
```

The environment variables that you specify must be defined. For example, you could define the *AUTODIR* environment variable at SAS invocation by using the following code:

```
-set AUTODIR /tmp/sasautos
```

For more information about how to define an environment variable, see [“SET System Option” on page 174](#).

Example 2: Specifying a Fileref in the *OPTIONS* Statement

The fileref that you specify must be defined. For example, you could define the *AUTODIR* fileref using a *FILENAME* statement:

```
filename AUTODIR '/tmp/sasautos';
```

Once the fileref is defined, you can use it in an *OPTIONS* statement to set the autocall library.

```
options sasautos=autodir;
```

See Also

System Options:

- [“APPEND= System Option” on page 39](#)
- [“INSERT= System Option” on page 99](#)
- [“MAUTOSOURCE System Option” in *SAS Viya Macro Language: Reference*](#)
- [“MRECALL System Option” in *SAS Viya Macro Language: Reference*](#)

SASUSER= System Option

Specifies the SAS library to use as the Sasuser library.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Files

**PROC OPTIONS
GROUP=** ENVFILES

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

SASUSER=*library-specification*

Syntax Description

library-specification

specifies the libref or the physical name that contains a user's Profile catalog. *library-specification* can be an environment variable.

Example Using an environment variable:

```
sas -sasuser $HOME
```

Details

The library and catalog are created automatically by SAS; you do not have to create them explicitly.

SERVICESBASEURL= System Option

Specifies the root URL for calls to SAS Viya services.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Communications: Networking and Encryption

PROC OPTIONS GROUP= COMMUNICATIONS

Default: The shipped default is None

Notes: This option is not used in programming-only deployments.

This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Example: "http://myservices.mycompany.com"

Syntax

SERVICESBASEURL="*your-site-services-URL*"

Syntax Description

"your-site-services-URL"

specifies the root URL that is used to connect to SAS Viya services.

Details

You use the SERVICBASEURL= option to connect to SAS Viya services from a batch or command line SAS program that does not use the Cloud Analytic Services (CAS) server. The CAS server is connected to SAS Viya services during deployment.

SEQ= System Option

Specifies the length of the numeric portion of the sequence field in input source lines or data lines.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Input Control: Data Processing

PROC OPTIONS INPUTCONTROL
GROUP=

Default: The shipped default is 8.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

SEQ=*n* | MIN | MAX | *hexX*

Syntax Description

n
specifies the length in terms of bytes.

MIN
sets the minimum length to 1.

MAX
sets the maximum length to 8.

Tip When SEQ=8, all eight characters in the sequence field are assumed to be numeric.

hexX
specifies the length as a hexadecimal. You must specify the value beginning with a number (0–9), followed by an X.

Details

Unless the S= or S2= system option specifies otherwise, SAS assumes an eight-character sequence field. However, some editors place some alphabetic information (for example, the filename) in the first several characters. The SEQ= value specifies the number of digits that are right-justified in the eight-character field. For example, if you specify SEQ=5 for the sequence field AAA00010, SAS looks at only the last five characters of the eight-character sequence field and, if the characters are numeric, treats the entire eight-character field as a sequence field.

See Also

System Options:

- [“S= System Option” on page 163](#)
- [“S2= System Option” on page 165](#)

SET System Option

Defines a SAS environment variable.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Files

PROC OPTIONS
GROUP=

ENVFILES

Default: The shipped default is None

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

SET=*variable-name value*

Syntax Description

variable-name

specifies the environment variable name.

value

specifies a pathname or other value for the environment variable.

Details

The SET option lets you define an environment variable that is valid within the SAS session and any shell started from within the SAS session. Using the SET option is similar to using the SAS `setenv` command. For information about executing system commands from within your SAS session, see [“Executing Operating System Commands from Your SAS Session” in *Batch and Line Mode Processing in SAS Viya*](#).

A special use for the SET option is to specify the name of the `!SASROOT` directory:

```
-set SASROOT pathname
```

The pathname specified can then be used to expand `!SASROOT` (as shown in [“Character Substitutions in Pathnames” in *Batch and Line Mode Processing in SAS Viya*](#)).

After exiting your SAS session, environment variables that are set with the SET option no longer exist.

SETINIT System Option

Specifies whether site license information can be altered.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: System Administration: Installation

PROC OPTIONS
GROUP=

INSTALL

Default: The shipped default is NOSETINIT.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

SETINIT | **NOSETINIT**

Syntax Description**SETINIT**

in a non-windowing environment, specifies that you can change license information by running the SETINIT procedure.

Requirement If the setinit has expired, you must use the SETINIT option when you run SAS to update the licensing information.

NOSETINIT

specifies not to enable you to alter site license information after installation.

Details

SETINIT is set in the installation process and is not normally changed after installation. The SETINIT option is valid only in a non-windowing SAS session.

SORTANOM System Option

Specifies options for the host sort utility.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Sort: Procedure Options

**PROC OPTIONS
GROUP=** SORT

Default: none

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

SORTANOM=*option(s)*

Required Argument***option***

can be any one or more of the following:

B

tells SyncSort to run in multi-call mode, instead of single-call mode. (See the documentation for **syncsort** for more information.)

Note This option is available for **syncsort** only.

T

writes to the SAS log statistics about the external sorting process.

V

writes to the SAS log all of the commands that are passed to the host sort utility.

SORTCUT System Option

Specifies the data size in number of observations above which SAS uses the host sort instead of the internal SAS sort.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Sort: Procedure Options
PROC OPTIONS GROUP=	SORT
Default:	The shipped default is 0.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

SORTCUT=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Required Arguments

n* | *nK* | *nM* | *nG* | *nT

specifies the number of observations in multiples of 1 (*n*); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). You can specify decimal values for the number of kilobytes, megabytes, or gigabytes. For example, a value of **800** specifies 800 observations, a value of **.782k** specifies 801 observations, and a value of **3m** specifies 3,145,728 observations.

hexX

specifies the number of observations as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by hexadecimal characters (0–9, A–F), and then followed by an X. For example, the value **2ffx** specifies 767 observations.

MIN

specifies 0 observations.

MAX

specifies 9,007,199,254,740,992 observations.

Details

When you specify SORTPGM=BEST, SAS uses the value of the SORTCUT and SORTCUTP options to determine whether to use the host sort or the SAS sort. If the number of observations in the data set is greater than the number that you specify with SORTCUT, the host sort is used. If both SORTCUT and SORTCUTP are either not defined or are set to 0, the SAS sort is used. If you specify both options and either condition is true, SAS chooses the host sort.

See Also

System Options:

- “SORTCUTP System Option” on page 178
- “SORTPGM System Option” on page 182

SORTCUTP System Option

Specifies the data size in bytes above which SAS uses the host sort instead of the internal SAS sort.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Sort: Procedure Options
PROC OPTIONS GROUP=	SORT
Default:	The shipped default is 0.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

SORTCUTP=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Required Arguments

n* | *nK* | *nM* | *nG* | *nT

specifies the number of bytes in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes) or 1,099,511,627,776 (terabytes). You can specify decimal values for the number of kilobytes, megabytes, or gigabytes. For example, a value of **8** specifies 8 bytes, a value of **.782k** specifies 801 bytes, and a value of **3m** specifies 3,145,728 bytes.

hexX

specifies the number of bytes as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by hexadecimal characters (0–9, A–F), and then followed by an X. For example, the value **2dx** specifies 45 bytes.

MIN

specifies 0 bytes.

MAX

specifies 9,007,199,254,740,992 bytes.

Details

When you specify SORTPGM=BEST, SAS uses the value of the SORTCUT and SORTCUTP options to determine whether to use the host sort or the SAS sort. If the data set to be sorted is larger than the number of bytes (or kilobytes or megabytes) that you specify with SORTCUTP, the host sort is used instead of the SAS sort. The value that you specify must be less than or equal to 2,147,483,647 bytes. If both SORTCUT and SORTCUTP are either not defined or are set to 0, the SAS sort is used. If you specify both options and either condition is true, SAS chooses the host sort.

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

$$\text{number-of-bytes} = ((\text{length-of-obs}) + (\text{length-of-all-keys})) * \text{number-of-obs}$$

See Also

System Options:

- “SORTANOM System Option” on page 176
- “SORTCUT System Option” on page 177
- “SORTPGM System Option” on page 182

SORTDEV System Option

Specifies the pathname used for temporary files created by the host sort utility.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Sort: Procedure Options
PROC OPTIONS GROUP=	SORT
Default:	Same location as -WORK (set in the !SASROOT/sasv9.cfg file)
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

SORTDEV=*'directory-specification'*

Details

The SORTDEV option specifies an alternative directory for temporary files created by the host sort program.

SORTDUP= System Option

Specifies whether the SORT procedure removes duplicate variables based on all variables in a data set or the variables that remain after the DROP or KEEP data set options have been applied.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Sort: Procedure Options
PROC OPTIONS GROUP=	SORT
Default:	The shipped default is PHYSICAL.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

`SORTDUP=PHYSICAL | LOGICAL`

Syntax Description

PHYSICAL

removes duplicates based on all the variables that are present in the data set.

LOGICAL

removes duplicates based on only the variables remaining after the `DROP=` and `KEEP=` data set options are processed.

Details

The `SORTDUP=` option specifies what variables to sort to remove duplicate observations when the `SORT` procedure `NODUPRECS` option is specified.

When `SORTDUP=` is set to `LOGICAL` and `NODUPRECS` is specified in the `SORT` procedure, duplicate observations are removed based on the variables that remain after a `DROP` or `KEEP` operation on the input data set. Setting `SORTDUP=LOGICAL` increases the number of duplicate observations that are removed because it eliminates variables before observations are compared. Setting `SORTDUP=LOGICAL` might improve performance.

When `SORTDUP=` is set to `PHYSICAL` and `NODUPRECS` is specified in the `SORT` procedure, duplicate observations are removed based on all of the variables in the input data set.

See Also

Procedures:

- “[SORT Procedure](#)” in *SAS Viya Visual Data Management and Utility Procedures Guide*

SORTEQUALS System Option

Specifies whether observations in the output data set with identical `BY` variable values are in a particular order.

Valid in: Configuration file, SAS command, `OPTIONS` statement, `SASV9_OPTIONS` environment variable

Category: Sort: Procedure Options

**PROC OPTIONS
GROUP=** SORT

Default: The shipped default is `SORTEQUALS`.

Note: This option can be restricted by a site administrator. For more information, see “[Restricted Options](#)” on page 8.

Syntax

`SORTEQUALS | NOSORTEQUALS`

Syntax Description

SORTEQUALS

specifies that observations with identical BY variable values are to retain the same relative positions in the output data set as in the input data set.

NOSORTEQUALS

specifies that no resources be used to control the order of observations with identical BY variable values in the output data set.

Interaction To achieve the best sorting performance when using the THREADS= system option, specify THREADS=YES and NOSORTEQUALS.

Tip To save resources, use NOSORTEQUALS when you do not need to maintain a specific order of observations with identical BY variable values.

Comparisons

The SORTEQUALS and NOSORTEQUALS system options set the sorting behavior of PROC SORT for your SAS session. The EQUAL or NOEQUAL option in the PROC SORT statement overrides the setting of the system option for an individual PROC step and specifies the sorting behavior for that PROC step only.

See Also

Procedure Statement Options:

- PROC SORT statement EQUALS option, “[SORT Procedure](#)” in *SAS Viya Visual Data Management and Utility Procedures Guide*

System Options:

- “[THREADS System Option](#)” on page 204

SORTNAME System Option

Specifies the name of the host sort utility.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Sort: Procedure Options

PROC OPTIONS GROUP= SORT

Default: None

Note: This option can be restricted by a site administrator. For more information, see “[Restricted Options](#)” on page 8.

Syntax

SORTNAME='host-sort-utility-name'

Details

The SORTNAME option specifies the name of the default host sort utility, **syncsort**.

See Also

System Options:

- [“SORTPGM System Option” on page 182](#)

SORTPARM System Option

Specifies parameters for the host sort utility.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Sort: Procedure Options
PROC OPTIONS GROUP=	SORT
Default:	None
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

SORTPARM=*'parameter(s)'*

Required Argument

parameter

specifies any parameter that you want to pass to the sort utility. For a description of these parameters, see the documentation for the sort that you are using.

SORTPGM System Option

Specifies whether to use the internal SAS sort utility or the host sort utility or to let SAS choose which sort utility to use.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Sort: Procedure Options
PROC OPTIONS GROUP=	SORT
Default:	BEST
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

SORTPGM=SAS | HOST | BEST

Required Arguments

SAS

tells SAS to use the SAS sort.

HOST

tells SAS to use the sort that is specified by the SORTNAME system option.

BEST

tells SAS to use the best routine to sort the data set: the SAS sort or the host sort that is specified by the SORTNAME system option. When set, the settings of the SORTCUT and SORTCUTP system options determine whether SAS chooses the SAS sort or the host sort. When SORTCUT and SORTCUTP are not set (or when they are both 0), SAS selects the sorting algorithm based on the following order of precedence:

- host sort utility
- SAS sort utility

See Also

System Options:

- [“SORTCUT System Option” on page 177](#)
- [“SORTCUTP System Option” on page 178](#)
- [“SORTNAME System Option” on page 181](#)
- [“SORTSIZE= System Option” on page 183](#)

SORTSIZE= System Option

Specifies the amount of memory that is available to the SORT procedure.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Categories: Sort: Procedure Options
System Administration: Memory
System Administration: Performance

PROC OPTIONS GROUP= MEMORY
PERFORMANCE
SORT

Default: The shipped default is MAX.

Syntax

SORTSIZE=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Syntax Description

n* | *nK* | *nM* | *nG* | *nT

specifies the amount of memory in terms of 1 (byte); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). For example, a value of **4000** specifies 4,000 bytes and a value of **2m** specifies 2,097,152 bytes. If *n*=0, the sort utility uses its default. Valid values for SORTSIZE range from 0 to 9,223,372,036,854,775,807.

hexX

specifies the amount of memory as a hexadecimal number. This number must begin with a number (0-9), followed by an X. For example, **0fffX** specifies 4095 bytes of memory.

MIN

specifies the minimum amount of memory available.

MAX

specifies the maximum amount of memory available.

Details

The SORT procedure uses the SORTSIZE system option to limit the amount of memory that it acquires or allocates for sorting. The amount of memory that SAS uses for the SORT procedure also depends on the values of the MEMSIZE and REALMEMSIZE system options. By contrast with the SORTSIZE option, the MEMSIZE system option places a limit on the total amount of virtual memory that SAS dynamically allocates at any time. This virtual memory is supported by a combination of real memory and paging space. The operating environment begins paging when the amount of virtual memory that is required exceeds the real memory that is available. To prevent paging and the associated performance problems, the SORTSIZE system option should be set to a subset of real memory. You can set SORTSIZE to MAX if MEMSIZE is set to a subset of real memory. In most cases, you can set SORTSIZE=MAX because this value limits the amount of memory that is used by the SORT procedure.

The proper specification of SORTSIZE= can improve sort performance by restricting the swapping of memory that is controlled by the operating environment.

See Also

Procedures:

- “SORT Procedure” in *SAS Viya Visual Data Management and Utility Procedures Guide*

SORTVALIDATE System Option

Specifies whether the SORT procedure verifies if a data set is sorted according to the variables in the BY statement when a user-specified sort order is denoted in the sort indicator.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Sort: Procedure Options

PROC OPTIONS GROUP= SORT

Default: The shipped default is NOSORTVALIDATE.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[SORTVALIDATE](#) | [NOSORTVALIDATE](#)

Syntax Description

SORTVALIDATE

specifies that the SORT procedure verifies if the observations in the data set are sorted by the variables specified in the BY statement.

NOSORTVALIDATE

specifies that the SORT procedure is not to verify if the observations in the data set are sorted.

Details

You can use the SORTVALIDATE system option to specify whether the SORT procedure validates that a data set is sorted correctly when the data set sort indicator shows a user-specified sort order. The user can specify a sort order by using the SORTEDBY= data set option in a DATA statement or by using the SORTEDBY= option in the DATASETS procedure MODIFY statement. When the sort indicator is set by a user, SAS cannot be absolutely certain that a data set is sorted according to the variables in the BY statement.

If the SORTVALIDATE system option is set and the data set sort indicator was set by a user, the SORT procedure performs a sequence check on each observation to ensure that the data set is sorted according to the variables in the BY statement. If the data set is not sorted correctly, SAS sorts the data set.

At the end of a successful sequence check or at the end of a sort, the SORT procedure sets the **Validated** sort information to Yes. If a sort is performed, the SORT procedure updates the **Sortedby** sort information to the variables that are specified in the BY statement.

If an output data set is specified, the **Validated** sort information in the output data set is set to Yes. If no sort is necessary, the data set is copied to the output data set.

See Also

Data Set Options:

- [“SORTEDBY= Data Set Option” in SAS Viya Data Set Options: Reference](#)

Procedures:

- [“DATASETS Procedure” in SAS Viya Visual Data Management and Utility Procedures Guide](#)
- [“SORT Procedure” in SAS Viya Visual Data Management and Utility Procedures Guide](#)

SOURCE System Option

Specifies whether SAS writes source statements to the SAS log.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: SAS Log
PROC OPTIONS GROUP=	LOGCONTROL
Default:	The shipped default is SOURCE.
Interaction:	SAS Studio sets this option to SOURCE after each code submission. For more information, see “System Options in SAS Studio” on page 7 .
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

[SOURCE](#) | [NOSOURCE](#)

Syntax Description

SOURCE

specifies to write SAS source statements to the SAS log.

NOSOURCE

specifies not to write SAS source statements to the SAS log.

Details

The SOURCE system option does not affect whether statements from a file read with %INCLUDE or from an autocall macro are printed in the SAS log.

Note: SOURCE must be in effect when you execute SAS programs that you want to send to SAS for problem determination and resolution.

SOURCE2 System Option

Specifies whether SAS writes secondary source statements from included files to the SAS log.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: SAS Log
PROC OPTIONS GROUP=	LOGCONTROL
Default:	The shipped default is NOSOURCE2.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

[SOURCE2](#) | [NOSOURCE2](#)

Syntax Description

SOURCE2

specifies to write to the SAS log secondary source statements from files that have been included by %INCLUDE statements.

NOSOURCE2

specifies not to write secondary source statements to the SAS log.

Details

Note: SOURCE2 must be in effect when you execute SAS programs that you want to send to SAS for problem determination and resolution.

SPOOL System Option

Specifies whether SAS statements are written to a utility data set in the Work library.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Input Control: Data Processing

PROC OPTIONS GROUP= INPUTCONTROL

Default: The shipped default is NOSPOOL.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[SPOOL](#) | [NOSPOOL](#)

Syntax Description

SPOOL

specifies that SAS write statements to a utility data set in the Work library for later use by a %INCLUDE or %LIST statement, or by the RECALL command, within a windowing environment.

NOSPOOL

specifies that SAS does not write statements to a utility data set. Specifying NOSPOOL accelerates execution time, but you cannot use the %INCLUDE and %LIST statements to resubmit SAS statements that were executed earlier in the session.

Example

Specifying SPOOL is especially helpful in interactive line mode because you can resubmit a line or lines of code by referring to the line numbers. Here is an example of code including line numbers:

```
00001 data test;
00002     input w x y z;
00003     datalines;
00004 411.365 101.945 323.782 512.398
00005 ;
```

If SPOOL is in effect, you can resubmit line number 1 by submitting this statement:

```
%inc 1;
```

You can also resubmit a range of lines by placing a colon (:) or hyphen (-) between the line numbers. For example, these statements resubmit lines 1 through 3 and 4 through 5 of the above example:

```
%inc 1:3;
%inc 4-5;
```

STARTLIB System Option

Specifies whether SAS assigns user-defined permanent librefs when SAS starts.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: External Files
PROC OPTIONS GROUP=	EXTFILES
Defaults:	The default is STARTLIB for the windowing environment. The default is NOSTARTLIB for batch, interactive-line, and noninteractive modes.
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

STARTLIB | **NOSTARTLIB**

Syntax Description

STARTLIB


specifies that when SAS starts, SAS assigns user-defined permanent librefs.

NOSTARTLIB

specifies that SAS is not to assign user-defined permanent librefs when SAS starts.

Details

In batch and interactive line-mode, SAS assigns permanent librefs only when you start SAS with the STARTLIB option specified either on the command line or in the configuration file.

In SAS Studio, you can assign a permanent libref by using the New Library window and selecting the **Re-create this library at startup** check box. To open the New Library window, click the **Libraries** section in the navigation pane and select .

STDIO System Option

Specifies whether SAS should use stdin, stdout, and stderr.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Input Control: Data Processing
PROC OPTIONS GROUP=	INPUTCONTROL
Default:	The shipped default NOSTDIO.

Syntax

[STDIO](#) | [NOSTDIO](#)

Syntax Description

STDIO

specifies that SAS uses stdin, stdout, and stderr.

NOSTDIO

specifies that SAS does not use stdin, stdout, and stderr.

Details

This option tells SAS to take its input from standard input (stdin), to write its log to standard error (stderr), and to write its output to standard output (stdout).

This option is designed for running SAS in batch mode or from a shell script. If you specify this option interactively, SAS starts a line mode session.

The STDIO option does not affect the assignment of the Stdio, Stdin, and Stderr filerefs.

For example, in the following SAS command, the file *myinput* is used as the source program, and files *myoutput* and *mylog* are used for the procedure output and log respectively.

```
sas -stdio <myinput> myoutput> mylog
```

If you are using the C shell, you should use parentheses:

```
(sas -stdio <myinput> myoutput)>& output_log
```

STEPCHKPT System Option

Specifies whether checkpoint-restart data for DATA and PROC steps is to be recorded for a batch program.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Default: The shipped default is NOSTEPCHKPT.

Restriction: The STEPCHKPT system option can be specified only if the LABELCHKPT system option is not specified when SAS starts.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[STEPCHKPT](#) | [NOSTEPCHKPT](#)

Syntax Description

STEPCHKPT

enables checkpoint mode, which specifies to record checkpoint-restart data for DATA and PROC steps.

NOSTEPCHKPT

disables checkpoint mode, which specifies not to record checkpoint-restart data. This is the default.

Details

Using the STEPCHKPT system option puts SAS in checkpoint mode for SAS programs that run in batch. Each time a DATA step or PROC step executes, SAS records data in a checkpoint-restart library. If a program terminates without completing, the program can be resubmitted. Execution begins with the step that was executing when the program terminated.

To ensure that the checkpoint-restart data is accurate, when you specify the STEPCHKPT option, also specify the ERRORCHECK STRICT option and set the ERRORABEND option so that SAS terminates for most errors.

Comparisons

The STEPCHKPT system option enables checkpoint mode for DATA and PROC steps in batch programs that terminate before completing. Execution resumes with the DATA or PROC step that was executing when the failure occurred.

The LABELCHKPT system option enables checkpoint mode for labeled code sections in batch programs that terminate before completing. Execution resumes at the labeled code section that was executing when the failure occurred.

See Also

Statements:

- [“CHECKPOINT EXECUTE_ALWAYS Statement” in SAS Viya Statements: Reference](#)

System Options:

- [“CHKPTCLEAN System Option” on page 58](#)
- [“ERRORABEND System Option” on page 77](#)

- “ERRORCHECK= System Option” on page 79
- “LABELCHKPT System Option” on page 104
- “STEPCHKPTLIB= System Option” on page 191
- “STEPRESTART System Option” on page 192

STEPCHKPTLIB= System Option

Specifies the libref of the library where checkpoint-restart data for DATA and PROC steps is saved.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Restriction: The STEPCHKPTLIB system option can be specified only if the LABELCHKPT system option is not specified when SAS starts.

Note: This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

STEPCHKPTLIB=*libref*

Syntax Description

libref

specifies the libref that identifies the library where the checkpoint-restart data for DATA and PROC steps is saved.

Default	Work
----------------	------

Requirement	The LIBNAME statement that identifies the checkpoint-restart library must use the BASE engine and be the first statement in the batch program.
--------------------	--

Details

When the STEPCHKPT system option is specified, checkpoint-restart data for batch programs is saved in the libref that is specified in the STEPCHKPTLIB= system option. If no libref is specified, SAS uses the Work library to save checkpoint data. The LIBNAME statement that defines the libref must be the first statement in the batch program.

If the Work library is used to save checkpoint data, the NOWORKTERM and NOWORKINIT system options must be specified so that the checkpoint-restart data is available when the batch program is resubmitted. These two options ensure that the Work library is saved when SAS ends and is restored when SAS starts. If the NOWORKTERM option is not specified, the Work library is deleted at the end of the SAS session and the checkpoint-restart data is lost. If the NOWORKINIT option is not specified, a new Work library is created when SAS starts, and again the checkpoint-restart data is lost.

The STEPCHKPTLIB= option must be specified for any SAS session that accesses checkpoint-restart data that is not saved to the Work library.

When you want to run the CLEANWORK utility, the Work library directory and its contents are deleted when the utility is run after the SAS session ends.

Comparisons

When the STEPCHKPT system option is set, the library specified by the STEPCHKPTLIB system option names the library where checkpoint-restart data is saved for DATA and PROC steps. When the STEPRESTART system option is set, the library specified by the STEPCHKPTLIB system option names the library where checkpoint-restart data is used to resume execution of DATA and PROC steps.

When the LABELCKPT system option is set, the library specified by the LABELCHKPTLIB system option names the library where checkpoint-restart data is saved for labeled code sections. When the LABELRESTART system option is set, the library specified by the LABELCHKPTLIB system option names the library where checkpoint-restart data is used to resume execution of labeled code sections.

See Also

Statements:

- [“CHECKPOINT EXECUTE_ALWAYS Statement” in SAS Viya Statements: Reference](#)

System Options:

- [“LABELCHKPT System Option” on page 104](#)
- [“LABELCHKPTLIB= System Option” on page 105](#)
- [“STEPCHKPT System Option” on page 189](#)
- [“STEPRESTART System Option” on page 192](#)
- [“WORKINIT System Option” on page 230](#)
- [“WORKTERM System Option” on page 231](#)

STEPRESTART System Option

Specifies whether to execute a batch program by using checkpoint-restart data for DATA and PROC steps.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Default: The shipped default is NOSTEPRESTART.

Restriction: The STEPRESTART system option can be specified only if the LABELCHKPT system option is not specified when SAS starts.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

STEPRESTART | NOSTEPRESTART

Syntax Description

STEPRESTART

enables restart mode, which specifies to execute the batch program by using the checkpoint-restart data for DATA and PROC steps.

NOSTEPRESTART

disables restart mode, which specifies not to execute the batch program using checkpoint-restart data.

Details

You specify the STEPRESTART option when you want to resubmit a batch program that ran in checkpoint mode and terminated before it completed. When you resubmit the batch program, SAS determines from the checkpoint data which DATA step or PROC step was executing when the program terminated, and resumes executing the batch program by using that DATA or PROC step.

Comparisons

When you specify the STEPRESTART option, SAS uses the checkpoint-restart data for DATA and PROC steps to resume execution of batch programs.

When you specify the LABELRESTART option, SAS uses the checkpoint-restart data for labeled code sections to resume execution of batch programs.

See Also

Statements:

- “CHECKPOINT EXECUTE_ALWAYS Statement” in *SAS Viya Statements: Reference*

System Options:

- “CHKPTCLEAN System Option” on page 58
- “LABELCHKPT System Option” on page 104
- “LABELRESTART System Option” on page 107
- “STEPCHKPT System Option” on page 189
- “STEPCHKPTLIB= System Option” on page 191

STIMEFMT System Option

Specifies the format that is used to display the time on FULLSTIMER and STIMER output.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: SAS Log

PROC OPTIONS LOGCONTROL
GROUP=

Default: The shipped default is 512M.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

STIMEFMT=*value(s)*

Required Arguments

value

specifies the options to use with STIMEFMT. The following options are available:

Datetime Stamp options

The datetime stamp options are described below:

- | | |
|--------|---|
| TS | specifies to always display the datetime stamp as part of STIMER and FULLSTIMER. |
| TSFULL | specifies to display the datetime stamp as part of FULLSTIMER. TSFULL is the default. |
| TSOFF | turns off the datetime stamp for STIMER and FULLSTIMER. |

Memory

is normally displayed as part of FULLSTIMER. The default memory output is displayed in kilobytes. The following options for memory are available:

- | | |
|---------|--|
| MEMFULL | writes memory statistics as part of FULLSTIMER, but not as part of STIMER. |
| MEM | writes memory statistics as part of FULLSTIMER and STIMER. |
| KB | writes memory in kilobytes. |
| MB | writes memory in megabytes. |
| GB | writes memory in gigabytes. |
| C | adds commas to the numbers in the memory display. |
| NC | does not add commas to the numbers in the memory display. |

Elapsed and CPU time

can be configured to display hours, minutes, seconds, or best fit in STIMER and FULLSTIMER.

- | | |
|---------------|--|
| Z H HOURS | writes the time as hours:minutes:seconds. |
| M MINUTES | writes the time as minutes:seconds. |
| S SECONDS | writes the time as seconds. |
| HMS | writes the format leaving out leading zeros for hours and minutes. |

Counters

specifies that additional counters can be displayed as part of FULLSTIMER.

- | | |
|------------|-------------------------|
| E ENABLE | enables extra counters. |
|------------|-------------------------|

D | DISABLE disables extra counters.

Help

provides two values that are used to access help for the STIMEFMT option:

FMT lists the available datetime stamp formats.

OPT lists other option values that are available.

Details

STIMEFMT Basics

The STIMEFMT system enables you to customize the format of output produced by the STIMER and FULLSTIMER system options. You can perform the following tasks using STIMEFMT:

- list the formats that are available:

```
options stimefmt=fmt;
```

- list other options that are available:

```
options stimefmt=opt;
```

- turn the datetime stamp on or off for STIMER:

```
options stimefmt=tson | tsoff | tsfull;
```

- combine options as needed:

```
options stimefmt=(tson YNNDDS);
```

- separate a memory value with commas:

```
options stimefmt=c;
```

- do not use commas when specifying values:

```
options stimefmt=nc;
```

- select a unit for memory:

```
options stimefmt=GB | MB | KB;
```

- turn on memory reporting for STIMER and FULLSTIMER:

```
options stimefmt=mem;
```

- set the time display in the datetime stamp:

```
options stimefmt=TOD | TIME | TIMEAMP;
(TOD and TIME specify military time.)
```

- control the display of CPU or real time by using hours or minutes

Formats for Displaying the Datetime Stamp

The format of the datetime stamp can be set to standard formats that are supported by SAS. These formats include the following:

ABS. (Absolute seconds since Jan. 1, 1970)

DATE. DATE9.

DDMMYY.	DDMMYY10.	DDMMYYB.
DDMMYYB10.	DDMMYYC.	DDMMYYC10.
DDMMYYD.	DDMMYYD10.	DDMMYYN.
DDMMYYN10.	DDMMYYP.	DDMMYYP10.
DDMMYYYS.	DDMMYYYS10.	

ISO. (ISO Standard Time)

MMDDYY.	MMDDYY10.	MMDDYY.
MMDDYYB10.	MMDDYYC.	MMDDYYC10.
MMDDYYD.	MMDDYYD10.	MMDDYYN.
MMDDYYN8.	MMDDYYP.	MMDDYYP10.
MMDDYYYS.	MMDDYYYS10.	

NLDATM. NLDATMAP.

YYMMDD.	YYMMDD10.	YYMMDDB.
YYMMDDDB10.	YYMMDDC.	YYMMDDC10.
YYMMDDD.	YYMMDDD10.	YYMMDDN.
YYMMDDN8.	YYMMDDP.	YYMMDDP10.
YYMMDDS.	YYMMDDS10.	

TOD. (Writes time as military time.)

TIME. (Writes time as military time.)

TIMEAMPM. (Writes time as AM and PM.)

The syntax for the OPTIONS statement is listed below:

```
options stimefmt=fmt;
```

where *fmt* is a valid SAS format.

Using Multiple Values for the STIMEFMT Option

The STIMEFMT option can specify multiple values at the same time to enable you to set multiple settings. Multiple values must be enclosed in parentheses. For example:

```
options stimefmt=(h YYMMDD. gb c);
```

Displaying the Settings for the STIMEFMT Option

PROC OPTIONS always displays the current state of all settings for STIMEFMT. The following example shows log output when you execute PROC OPTIONS:

```
proc options option=stimefmt;
run;
```

Log 2.2 Log Output from PROC OPTIONS

```
SAS (r) Proprietary Software Release V.03.02 TS1M0

STIMEFMT=(NLDATM2. HMS TIMEAMPM KB MEMFULL TSFULL NC)
          Specifies the format that is used to display the FULLSTIMER
and STIMER output for timestamp, memory, CPU and
          elapsed time statistics.
NOTE: PROCEDURE OPTIONS used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds
```

Resetting STIMEFMT to the Default Values

You can reset the settings for STIMEFMT to its default values by executing the following OPTIONS statement:

```
options stimefmt=normal;
```

See Also

System Options:

- “FULLSTIMER System Option” on page 92
- “STIMER System Option” on page 197

STIMER System Option

Specifies whether to write a subset of system performance statistics to the SAS log.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Log and Procedure Output Control: SAS Log

PROC OPTIONS GROUP= LOGCONTROL

Default: The shipped default is STIMER.

Syntax

STIMER | NOSTIMER

Required Arguments

STIMER

writes only real time and CPU time to the SAS log.

NOSTIMER

does not write any statistics to the SAS log.

Details

The STIMER system option specifies whether a subset of all the performance statistics of your system that are available to SAS are written to the SAS log. (Using STIMEFMT can affect the output.) Here is an example of STIMER output:

Log 2.3 STIMER Output

real time	1.34 seconds
cpu time	0.04 seconds

STIMER displays the following statistics:

Table 2.3 Description of STIMER Statistics

Statistic	Description
real time	the amount of time spent to process the SAS job. Real time is also referred to as elapsed time.
CPU time	the total time spent to execute your SAS code and to perform system overhead tasks on behalf of the SAS process. This value is the combination of the user CPU and system CPU statistics from FULLSTIMER.

If both STIMER and FULLSTIMER are set, the FULLSTIMER statistics are written to the SAS log.

Note: Some procedures use multiple threads. On computers with multiple CPUs, the operating system can run more than one thread simultaneously. Consequently, CPU time might exceed real time in your STIMER output. For example, a SAS procedure could use two threads that run on two separate CPUs simultaneously. The value of CPU time would be calculated as the following:

```
CPU1 time + CPU2 time = total CPU time
1 second + 1 second = 2 seconds
```

Because CPU1 can run a thread at the same time that CPU2 runs a separate thread, you can theoretically consume 2 CPU seconds in 1 second of real time.

See Also

System Options:

- [“FULLSTIMER System Option” on page 92](#)
- [“STIMEFMT System Option” on page 193](#)

STRIPESIZE= System Option

Specifies one or more directory and size argument pairs that set the size of SAS data sets and utility files in that directory to the size of an I/O device stripe.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Categories: Files: SAS Files
System Administration: Performance
System Administration: TK

PROC OPTIONS GROUP= SASFILES
PERFORMANCE
TK

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

STRIPESIZE=(*directory-path-1 size-1 <directory-path-2 size-2 ...>*)

STRIPESIZE=(*directory-path* RESET)

Syntax Description

directory-path

specifies an existing directory where the I/O page size for SAS data sets and utility files that are created in this directory is set to *size*. Subdirectories of *directory-path* inherit the data set or utility file size unless the STRIPESIZE= option has been set for the subdirectory.

Requirements If *directory-path* contains spaces, enclose the directory in quotation marks.

directory-path must be a valid Linux path and the casing must match the casing for the Linux directory.

size

specifies the number of bytes in a RAID (Redundant Array of Independent Disks) stripe, in multiples of 1: 1,024 (kilo); 1,048,576 (mega); 1,073,741,824 (giga). For example, a value of 1024 specifies 1,024 bytes, and a value of 3m specifies 3,145,728 bytes.

Range 1024–2147483648

RESET

specifies to remove the directory from the value of the STRIPESIZE= option.

Details

You can use the STRIPESIZE= option to set the SAS I/O buffer size for a directory to be the size of a RAID stripe. SAS data sets or utility files that are created in the directory have a page size that matches the RAID stripe size. Using this option can improve the performance of individual disks and improve the performance of the SORT procedure.

Setting the STRIPESIZE= option supersedes buffer size option specifications that are set by data set options and procedure options, as well as the BUFSIZE= system option. If the value of *size* is determined to be in error, the page size is determined in this order:

1. data set options or procedure options
2. the BUFSIZE= system option

If the value of *size* is determined to be in error, the size of utility files is determined by the value of the UBUFSIZE= system option.

If SAS determines that the value of *size* might cause inefficient processing, SAS continues to use the value of *size* to create data sets and utility files, and writes a message to the SAS log.

Each time you set the STRIPESIZE= option, the directory and the size are appended to the current value of the option.

To update the stripe size for a directory that is currently in the STRIPESIZE= value list, set the STRIPESIZE= option by using the directory path and a new stripe size value.

See Also

Data Set Options:

- “BUFSIZE= Data Set Option” in *SAS Viya Data Set Options: Reference*

Procedures:

- “SORT Procedure” in *SAS Viya Visual Data Management and Utility Procedures Guide*

System Options:

- “BUFSIZE= System Option” on page 47

SYNTAXCHECK System Option

Specifies whether to enable syntax check mode for multiple steps.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Default: The shipped default is SYNTAXCHECK.

Interaction: SAS Studio sets this option to NOSYNTAXCHECK before each code submission and to SYNTAXCHECK after each code submission. For more information, see [“System Options in SAS Studio” on page 7](#).

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[SYNTAXCHECK](#) | [NOSYNTAXCHECK](#)

Syntax Description

SYNTAXCHECK
enables syntax check mode.

NOSYNTAXCHECK
does not enable syntax check and enables continuous processing of statements regardless of syntax error conditions.

CAUTION **Setting NOSYNTAXCHECK might cause a loss of data.**
Manipulating and deleting data by using untested code might result in a loss of data if your code contains invalid syntax. Be sure to test code completely before placing it in a production environment.

Details

If a syntax or semantic error occurs in a DATA step after the SYNTAXCHECK option is set, then SAS enters syntax check mode, which remains in effect from the point where SAS encountered the error to the end of the code that was submitted. After SAS enters syntax mode, all subsequent DATA step statements and PROC step statements are validated.

SAS can enter syntax check mode only if your program creates a data set. If you use the DATA_NULL_ statement, then SAS cannot enter syntax check mode because no data set is created.

In syntax check mode, SAS internally sets the OBS= option to 0 and the REPLACE/NOREPLACE option to NOREPLACE. When these options are in effect, SAS acts as follows:

- reads the remaining statements in the DATA step or PROC step
- checks that statements are valid SAS statements
- executes global statements
- writes error to the SAS log
- creates the descriptor portion of any output data sets that are specified in program statements
- does not write any observations to new data sets that SAS creates
- does not execute most of the subsequent DATA steps or procedures in the program (exceptions include PROC DATASETS and PROC CONTENTS)

Any data sets that are created after SAS has entered syntax check mode do not replace existing data sets with the same name.

Place the OPTIONS statement that enables SYNTAXCHECK before the step for which you want it to take effect. If you place the OPTIONS statement inside a step, then SYNTAXCHECK does not take effect until the beginning of the next step.

Comparisons

The ERRORCHECK= option can be set to enable or disable syntax check mode for the LIBNAME statement, the FILENAME statement, and the %INCLUDE statement. If you specify the NOSYNTAXCHECK option and the ERRORCHECK=STRICT option, SAS does not enter syntax check mode when an error occurs.

See Also

System Options:

- [“ERRORCHECK= System Option” on page 79](#)

SYSIN System Option

Specifies the default location of SAS source code when running in batch mode.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Files

**PROC OPTIONS
GROUP=** ENVFILES

Default: None

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

SYSIN *filename* | **NOSYSIN**

Required Arguments

SYSIN *filename*

specifies an external file. The value for *filename* must be a valid Linux filename.

NOSYSIN

invokes SAS, processes the autoexec file, and then terminates SAS, returning you to the command prompt.

Details

This option applies only when you are using batch mode. It is not necessary to precede the filename with the SYSIN option if the filename immediately follows the keyword **SAS**. For example, the following two SAS commands are equivalent:

```
sas saspgms/report1.sas
sas -sysin saspgms/report1.sas
```

The syntax of the SYSIN system option also enables you to specify NOSYSIN. If you specify NOSYSIN, SAS is invoked, the autoexec file is processed, and then SAS terminates, returning you to the command prompt. The following example shows the syntax:

```
sas -nosysin -autoexec mysas.sas
```

This option is useful if you want to test an autoexec file without actually running a complete SAS session.

See Also

[“Running SAS in Batch Mode” in *Batch and Line Mode Processing in SAS Viya*](#)

TERMINAL System Option

Specifies whether to associate a terminal with a SAS session.

Valid in: Configuration file, SAS invocation

Category: Environment Control: Initialization and Operation

**PROC OPTIONS
GROUP=** EXECMODES

Default: The shipped default is TERMINAL.

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

[TERMINAL](#) | [NOTERMINAL](#)

Syntax Description

TERMINAL

specifies that a physical display is available for the execution environment.

NOTERMINAL

specifies that a physical display is not available for the execution environment. Dialog boxes are not displayed.

Details

SAS defaults to TERMINAL, but might set the option to NOTERMINAL if SAS determines that the session should run in the background.

The TERMINAL option is typically used with the following execution modes:

- SAS windowing environment mode
- interactive line mode
- noninteractive mode

The NOTERMINAL option is typically used with server execution modes.

TERMSTMT= System Option

Specifies the SAS statements to execute when SAS terminates.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Initialization and Operation

PROC OPTIONS GROUP= EXECMODES

Operating environment: In some operating system environments there is a limit to the size of the value for TERMSTMT=. To circumvent this limitation, you can use the %INCLUDE statement.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

TERMSTMT=*'statement(s)'*

Syntax Description

'statement(s)'

is one or more SAS statements.

Length maximum length is 2,048 characters

Details

TERMSTMT= is fully supported in batch mode. In interactive modes, TERMSTMT= is executed only when you submit the ENDSAS statement from an editor window to terminate the SAS session. Terminating SAS by any other means in interactive mode results in TERMSTMT= not being executed.

An alternate method for specifying TERMSTMT= is to put a %INCLUDE statement at the end of a batch file or to submit a %INCLUDE statement before terminating the SAS session in interactive mode.

Comparisons

TERMSTMT= specifies the SAS statements to be executed at SAS termination, and INITSTMT= specifies the SAS statements to be executed at SAS initialization.

See Also

Statements:

- “%INCLUDE Statement” in *SAS Viya Statements: Reference*

System Options:

- “INITSTMT= System Option” on page 98

THREADS System Option

Specifies that SAS uses threaded processing if available.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: System Administration: Performance

PROC OPTIONS GROUP= PERFORMANCE

Default: The shipped default is THREADS.

Note: This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

THREADS | **NOTHREADS**

Syntax Description

THREADS

specifies to use threaded processing for SAS applications that support it.

Interaction If THREADS is specified either as a SAS system option or in PROC SORT, and another program has the input SAS data set open for reading or writing, then the procedure might fail and write a subsequent message to the SAS log.

NOTHEADS

specifies not to use threaded processing for running SAS applications that support it.

Interaction When you specify NOTHEADS, CPUCOUNT= is ignored unless you specify a procedure option that overrides the NOTHEADS system option.

Details

The THREADS system option enables some legacy SAS processes that are thread-enabled to take advantage of multiple CPUs by threading the processing and I/O operations. Threading the processing and I/O operations achieves a degree of parallelism that generally reduces the real time to completion for a given operation at the possible cost of additional CPU resources. The thread-enabled processes include PROC SORT and PROC SQL.

In some cases, SAS might determine to use a single-threaded operation. For example, small data sets might not be processed using threading.

The SAS Object Spawner uses threaded technology without the consideration of the NOTHEADS option.

Comparisons

The system option THREADS determines when threaded processing is in effect. The SAS system option CPUCOUNT= suggests how many system CPUs are available for use by thread-enabled SAS procedures.

See Also**System Options:**

- [“CPUCOUNT= System Option” on page 62](#)
- [“UTILLOC= System Option” on page 210](#)

TIMEZONE= System Option

Specifies the user local time zone.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Language Control

PROC OPTIONS GROUP= LANGUAGECONTROL

Alias: TZ=, except in the restricted options configuration file where TIMEZONE= must be used

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

TIMEZONE=*time-zone-name* | *time-zone-ID*

Syntax Description

time-zone-name

specifies a three- or four-character time zone name. For example, EST is a time zone name for Eastern Time.

Default BLANK, indicating that the SAS server time zone and the client time zone are the same

See For a list of time zone names, see [Appendix 1, “Time Zone IDs and Time Zone Names,”](#) on page 267.

time-zone-ID

specifies a *region/area* value that is defined by SAS. When you specify a time zone ID, the time zone that SAS uses is determined by time zone name and daylight saving time rules.

Note Time zone IDs are compatible with Java time zone names.

See For a list of time zone IDs, see [Appendix 1, “Time Zone IDs and Time Zone Names,”](#) on page 267.

Details

You set the TIMEZONE= option to a time zone ID or a time zone name in order for SAS to use a particular time zone. The time zone setting affects the following SAS components:

- times that are recorded by events and logs
- time of data set creation or modification
- DATE() function
- DATETIME() function
- TIME() function
- TODAY() function
- time zone functions TZONEOFF(), TZONEID(), TZONENAME(), TZONES2U(), and TZONEU2S(),
- time zone formats B8601DXw., E8601DXw., B8601LXw., E8601LXw., B8601TXw., E8601TXw., NLDATMZw., NLDATMTZw., and NLDATMWZw.

You set a time zone by specifying a time zone ID or a time zone name. A time zone ID is a region and an area separated by a forward slash (/). For example, America/New_York and Asia/Osaka are time zone IDs.

A time zone name is a three- or four-character name for a time zone. For example, EST is Eastern Time and JST is Japan Time. SAS determines the time by using time zone rules, including daylight saving time rules, before using a time value.

Some time zones names are valid for different locales. For example, CST is Central Daylight Time, Cuba Daylight Time, and China Daylight Time. SAS uses the value of the LOCALE= system option to determine the region and area to use. If TIMEZONE='CST' and LOCALE='zh_CN', SAS uses the Asia/Beijing time zone. If the time zone name does not exist for the locale, SAS searches all time zones and sets the time zone to the first match that it finds.

When this option is restricted and the value of TIMEZONE= is the default value of BLANK, time zone behavior does not use time zone information.

See Also

“Specifying Time Zones in SAS” in *SAS Viya National Language Support: Reference Guide*

UBUFNO= System Option

Specifies the number of buffers to use for utility files.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Files: SAS Files System Administration: Performance
PROC OPTIONS GROUP=	SASFILES PERFORMANCE
Default:	The shipped default is 0.

Syntax

UBUFNO=*n* | *nK* | *nM* | *nG* | *hexX* | MIN | MAX

Syntax Description

n | *nK* | *nM* | *nG*

specifies the number of utility file buffers to be allocated in multiples of 1: 1,024 (kilo); 1,048,576 (mega); 1,073,741,824 (giga). For example, a value of **8** specifies 8 buffers, and a value of **.003k** specifies 3 buffers.

Range 0–20

hexX

specifies the number of utility file buffers as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **0fX** specifies 15 buffers.

MIN

sets the minimum number of utility file buffers to 0, which causes SAS to use the minimum optimal value for the operating environment.

MAX

sets the number of utility file buffers to 20.

Details

The number of buffers is not a permanent attribute of the utility file and is valid only for the current SAS session or job. The UBUFNO= option applies to utility files that are opened for input, output, or update.

The optimal value for buffer system options is dependent on your operating environment. Experiment with various buffers sizes to determine the optimal value for these system options.

See Also

System Options:

- “BUFNO= System Option” on page 45
- “UBUFSIZE= System Option” on page 208

UBUFSIZE= System Option

Specifies the buffer size for utility files.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Files: SAS Files System Administration: Performance
PROC OPTIONS GROUP=	SASFILES PERFORMANCE
Default:	The shipped default is 0.
Note:	This option cannot be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

UBUFSIZE=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Syntax Description

n* | *nK* | *nM* | *nG* | *nT

specifies the utility file buffer size in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). For example, a value of **8** specifies 8 bytes, and a value of **3m** specifies 3,145,728 bytes.

hexX

specifies the utility file buffer size as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **2dx** sets the page size to 45 bytes.

MIN

sets the utility file buffer size to 0. A value of 0 specifies that the buffer size is set to the default page size for the operating environment.

MAX

sets the utility buffer size to the maximum possible number in your operating environment, up to the largest four-byte, signed integer, which is $2^{31}-1$, or approximately 2 billion bytes.

Details

The buffer size is the amount of data that can be transferred from a single input/output operation to one buffer. The buffer size is a permanent attribute of the utility file and is

used when the data set is processed. The `UBUFSIZE=` option sets the buffer size for utility files that SAS uses to process data sets.

The optimal value for buffer system options is dependent on your operating environment. Experiment with various buffers sizes to determine the optimal value for these system options.

See Also

System Options:

- [“BUFSIZE= System Option” on page 47](#)
- [“DATAPAGESIZE= System Option” on page 64](#)

USER= System Option

Specifies the default permanent SAS library.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Files

PROC OPTIONS GROUP= ENVFILES

Note: This option cannot be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

`USER=library-specification`

Syntax Description

library-specification

specifies the libref or physical name of a SAS library.

Requirements A libref must already be assigned.

The SAS library must be a directory name.

Details

If this option is specified, you can use one-level names to reference permanent SAS files in SAS statements. However, if `USER=WORK` is specified, SAS assumes that files referenced with one-level names refer to temporary work files.

USERCONFIG System Option

Specifies whether SAS processes the `sasv9.cfg` configuration file in a user's home directory.

Valid in: Configuration file, SAS invocation

Category:	Environment Control: Files
PROC OPTIONS GROUP=	ENVFILES
Default:	The shipped default is USERCONFIG
Restriction:	This option can be specified by an administrator only in the restricted options file.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

USERCONFIG | **NOUSERCONFIG**

Syntax Description

USERCONFIG

specifies that configuration processing processes a sasv9.cfg file in the user’s home directory.

NOUSERCONFIG

specifies that configuration processing does not process a sasv9.cfg file in the user’s home directory.

Details

A user’s home directory is defined in /etc/passwd. The home directory is not the value of the \$HOME environment variable or the shell’s interpretation of ~.

See Also

“Restricting SAS System Options” in *SAS Viya Administration: Workspace Server and Object Spawner*

UTILLOC= System Option

Specifies one or more file system locations in which enabled threaded applications can store utility files.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is Work.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

UTILLOC=WORK | *filename* | *location* | (*location-1 location-2 ...*)

Syntax Description

WORK

specifies that SAS creates utility files in the same directory as the Work library.

filename

specifies that SAS selects the utility-file location from a file that contains a list of directories and optional keywords. The directory that SAS chooses is valid as the utility-file location for the current SAS session.

location

specifies the location of an existing directory for utility files that are created by applications. Enclose *location* in single or double quotation marks when the location contains spaces.

(location-1 location-2 ...)

specifies a list of existing directories that can be accessed in parallel for utility files that are created by applications. A single utility file cannot span locations. Enclose a location in single or double quotation marks when the location contains spaces. Any location that does not exist is deleted from the value of the UTILLOC= system option.

Requirement When you specify more than one location, you must enclose the list of locations in parentheses.

Details

The Basics

The UTILLOC option specifies one or more locations for utility files that applications use for threaded processing.

Each location that is specified for the UTILLOC option identifies a single place at which utility files can be created. If multiple locations are specified, then the locations are used on a rotating basis by SAS applications as utility files are required.

For applications that use multiple utility files at the same time, specifying multiple locations that correspond to separate physical I/O devices might improve performance by reducing competition for device resources.

Allowing SAS to Select the Utility-File Location

The *filename* option contains a list of directories that is used to select the utility-file location. You can add one of the following methods to the file to indicate how you want SAS to select the utility-file location:

METHOD=RANDOM

specifies that SAS randomly selects the utility file location from the list of directories. SAS selects one utility-file location per SAS session. This selection enables you to balance the I/O load across multiple hardware systems. The file /sasinfo/utilfiles/ might look like this:

```
/disk1/sastempfiles
/disk2/sastempfiles
/disk3/sastempfiles
method=random
```

METHOD=SPACE

specifies that SAS selects the directory that has the most available space. Under UNIX, the file /sasinfo/utilfiles/ might look like this:

```

/disk1/sastempfiles
/disk2/sastempfiles
/disk3/sastempfiles
method=space

```

If the METHOD keyword is not specified, SAS defaults to randomly selecting a directory.

Utility Files and the SORT Procedure

For the SORT procedure, the UTILLOC= system option affects the placement of the utility files only if the multi-threaded SAS sort is used. The multi-threaded SAS sort can be invoked when the THREAD system option is specified and the value of the CPUCOUNT= system option is greater than 1. The multi-threaded SAS sort can also be invoked when you specify the THREADS option in the PROC SORT statement. The multi-threaded sort stores all temporary data in a single utility file within one of the locations that are specified by the UTILLOC= system option. The size of this utility file is proportional to the amount of data that is read from the input data set. A second utility file of the same size can be created in another of these locations when the amount of data that is read from the input data set is large or the amount of memory that is available to the SORT procedure is small.

See Also

Procedures:

- “SORT Procedure” in *SAS Viya Visual Data Management and Utility Procedures Guide*

System Options:

- “CPUCOUNT= System Option” on page 62
- “THREADS System Option” on page 204

VALIDFMTNAME= System Option

Specifies the maximum size (32 characters or 8 characters) that user-created format and informat names can be before an error or warning is issued.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Files: SAS Files

PROC OPTIONS GROUP= SASFILES

Default: The shipped default is LONG.

Note: This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

VALIDFMTNAME=LONG | FAIL | WARN

Syntax Description

LONG

specifies that format and informat names can be up to 32 alphanumeric characters.

FAIL

specifies that creating a format or informat name that is longer than eight characters results in an error message.

WARN

specifies that creating a format or informat name that is longer than eight characters results in a warning message to remind you that the format or informat cannot be used with releases prior to SAS 9.

Details

SAS 9 enables you to define format and informat names up to 32 characters. Previous releases were limited to eight characters. The VALIDFMTNAME= system option applies to format and informat names in both data sets and format catalogs. VALIDFMTNAME= does not control the length of format and informat names. It only controls the length of format and informat names that you associate with variables when you create a SAS data set.

If a SAS data set has a variable with a long format or informat name, which means that a release before SAS 9 cannot read it, then you can remove the long name so that the data set can be accessed by an earlier release. However, in order to retain the format attribute of the variable, an identical format with a short name would have to be applied to the variable.

Note: After you create a format or informat using a name that is longer than eight characters, if you rename it using eight or fewer characters, a release before SAS 9 cannot use the format or informat. You must re-create the format or informat using the shorter name.

See Also

Procedures:

- [“FORMAT Procedure” in SAS Viya Visual Data Management and Utility Procedures Guide](#)

VALIDMEMNAME= System Option

Specifies the rules for naming SAS data sets, SAS data views, and item stores.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is COMPATIBLE.
Applies to:	BASE engine
Restriction:	Regardless of the value of VALIDMEMNAME, a member name cannot end in the special character # followed by three digits. This is because it would conflict with the

naming conventions for generation data sets. Using such a member name results in an error.

Interaction: SAS Studio sets VALIDMEMNAME=COMPAT before and after each code submission. For more information, see [“System Options in SAS Studio” on page 7](#).

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

VALIDMEMNAME=[COMPATIBLE](#) | [EXTEND](#)

Syntax Description

COMPATIBLE

specifies that a SAS data set name, a SAS data view name, or an item store name must follow these rules:

- The length of the names can be up to 32 characters.
- Names must begin with a letter of the Latin alphabet (A–Z, a–z) or an underscore. Subsequent characters can be letters of the Latin alphabet, numerals, or underscores.
- Names cannot contain blanks or special characters except for the underscore.
- Names can contain mixed-case letters. SAS internally converts the member name to uppercase. Therefore, you cannot use the same member name with a different combination of uppercase and lowercase letters to represent different variables. For example, `customer`, `Customer`, and `CUSTOMER` all represent the same member name. How the name is saved on disk is determined by the operating environment.

Alias `COMPAT`

EXTEND

specifies that a SAS data set name, a SAS data view name, or an item store name must follow these rules:

- Names can include national characters.
- The name can include special characters, except for the `/ \ * ? " < > | : - .` characters.
- The name must contain at least one character (letters, numbers, valid special characters, and national characters).
- The length of the name can be up to 32 bytes.
- Null bytes are not allowed.
- Names cannot begin with a blank or a `'` (the period).
- Leading and trailing blanks are deleted when the member is created.
- Names can contain mixed-case letters. SAS internally converts the member name to uppercase. Therefore, you cannot use the same member name with a different combination of uppercase and lowercase letters to represent different variables. For example, `customer`, `Customer`, and `CUSTOMER` all represent the same member name. How the name appears is determined by the operating environment.

Requirement	When VALIDMEMNAME=EXTEND, SAS data set names, SAS data view names, and item store names must be written as a SAS name literal if the name includes blank spaces, special characters, or national characters. If you use either the percent sign (%) or the ampersand (&), then you must use single quotation marks in the name literal in order to avoid interaction with the SAS Macro Facility.
Operating environment	When you reference a SAS file directly by its physical name, the final embedded period is an extension delimiter. If a physical file reference includes a SAS member name that contains a period, you must add the file extension. For example, if you reference the data set name my.member as a physical file, you would add the file extension sas7bdat to the reference, as shown in this SET statement: set './saslib/my.member.sas7bdat'.
Tip	The name is displayed in uppercase letters.
Examples	data "August Purchases"n; data 'Años de empleo'n.;
CAUTION	Throughout SAS, using the name literal syntax with SAS member names that exceed the 32-byte limit or that have excessive embedded quotation marks might cause unexpected results. The intent of the VALIDMEMNAME=EXTEND system option is to enable compatibility with other DBMS member naming conventions, such as allowing embedded blanks and national characters.

Details

When VALIDMEMNAME= EXTEND, valid characters that are allowed in a SAS data set name, SAS data view name, and an item store name are extended to these characters:

- international characters
- characters supported by third-party databases
- characters that are commonly used in a filename

Only the DATA, VIEW, and ITEMSTOR SAS member types support the extension of characters. The other member types, such as CATALOG and PROGRAM, do not support the extended characters. INDEX and AUDIT types that exist only with the associated DATA member support extended characters.

See Also

System Options:

- [“VALIDVARNAME= System Option” on page 215](#)

VALIDVARNAME= System Option

Specifies the rules for valid SAS variable names that can be created and processed during a SAS session.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Files: SAS Files
PROC OPTIONS GROUP=	SASFILES
Default:	The shipped default is V7.
Interaction:	SAS Studio sets VALIDVARNAME= to the Preferences setting before each code submission. For more information, see “System Options in SAS Studio” on page 7 .
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

VALIDVARNAME=V7 | UPCASE | ANY

Syntax Description

V7

specifies that variable names must follow these rules:

- The length of a SAS variable names can be up to 32 characters.
- The first character must begin with a letter of the Latin alphabet (A–Z, a–z) or the underscore. Subsequent characters can be letters of the Latin alphabet, numerals, or underscores.
- Trailing blanks are ignored. The variable name alignment is left-justified.
- A variable name cannot contain blanks or special characters except for the underscore.
- A variable name can contain mixed-case letters. SAS stores and writes the variable name in the same case that is used in the first reference to the variable. However, when SAS processes a variable name, SAS internally converts it to uppercase. Therefore, you cannot use the same variable name with a different combination of uppercase and lowercase letters to represent different variables. For example, `cat`, `CaT`, and `CAT` all represent the same variable.
- Do not assign variables the names of special SAS automatic variables (such as `_N_` and `_ERROR_`) or variable list names (such as `_NUMERIC_`, `_CHARACTER_`, and `_ALL_`) to variables.

Examples `season='summer';`

```
_____
percent_of_profit=percent;
_____
```

UPCASE

specifies that the variable name follows the same rules as V7, except that the variable name is uppercase, as in earlier versions of SAS.

ANY

specifies that SAS variable names must follow these rules:

- The name can begin with or contain any characters, including blanks, national characters, special characters, and multi-byte characters.
- The name can be up to 32 bytes in length

- The name cannot contain any null bytes
- Leading blanks are preserved, but trailing blanks are ignored
- The name must contain at least one character. A name with all blanks is not permitted.
- The name contains mixed-case letters. SAS stores and writes the variable name in the same case that is used in the first reference to the variable. However, when SAS processes a variable name, SAS internally converts it to uppercase. Therefore, you cannot use the same variable name with a different combination of uppercase and lowercase letters to represent different variables. For example, `cat`, `CaT`, and `CAT` all represent the same variable.

See

Examples `'% of profit'n=percent;`

`'items@warehouse'n=itemnum;`

CAUTION **Throughout SAS, using the name literal syntax with SAS member names that exceed the 32-byte limit or have excessive embedded quotation marks might cause unexpected results.** The intent of the `VALIDVARNAME=ANY` system option is to enable compatibility with other DBMS variable (column) naming conventions, such as allowing embedded blanks and national characters.

See Also

System Options:

- [“VALIDMEMNAME= System Option” on page 213](#)

VARINITCHK= System Option

Specifies whether a DATA step stops or continues executing when a variable is not initialized and the type of message to write to the SAS log.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Files: SAS Files

PROC OPTIONS GROUP= SASFILES

Default: The shipped default is NOTE.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

`VARINITCHK=NONOTE | NOTE | WARN | ERROR`

Syntax Description**NONOTE**

specifies that the DATA step continues to execute without writing a message to the SAS log when a variable is not initialized.

NOTE

specifies that the DATA step continues to execute and writes a note to the SAS log when a variable is not initialized.

WARN

specifies that the DATA step continues to execute and writes a warning message to the SAS log when a variable is not initialized.

Alias WARNING

ERROR

specifies that the DATA step stops executing and writes an error message to the SAS log when a variable is not initialized.

Details

By default, SAS writes a note to the SAS log if a variable is not initialized. The VARINITCHK= option enables you to stop or continue executing if a variable is not initialized. You can also set the type of message that is written to the SAS log. SAS can issue a note, a warning, an error, or no note at all. When VARINITCHK=ERROR, SAS stops processing and writes an error message to the SAS log. For all other settings of VARINITCHK=, the DATA step continues executing.

Here are some of the contexts where a variable might not be initialized:

- the variable appears on the right side of an assignment operator or the SUM statement
- the variable is a parameter to a CALL routine
- the variable is contained in an array
- the variable can be set by a SET, MERGE, MODIFY, or UPDATE statement
- the variable is specified in an INPUT statement
- the variable is initialized in a RETAIN statement

VARLENCHK= System Option

Specifies the type of message to write to the SAS log when the input data set is read using the SET, MERGE, UPDATE, or MODIFY statements.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Files: SAS Files

PROC OPTIONS GROUP= SASFILES

Default: The shipped default is WARN.

Restriction: This option is not valid for input data read in CAS.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

VARLENCHK=NOWARN | WARN | ERROR

Syntax Description

NOWARN

specifies that no warning message is issued when the length of a variable that is being read is larger than the length that is defined for the variable.

WARN

specifies that a warning is issued when the length of a variable that is being read is larger than the length that is defined for the variable.

ERROR

specifies that an error message is issued when the length of a variable that is being read is larger than the length that is defined for the variable.

Details

CAUTION:

Data might be truncated if you change the length of a variable. After a variable is defined, the length of a variable can be changed only by a LENGTH statement. If a variable is read by the SET, MERGE, UPDATE, or MODIFY statements and the length of the variable is longer than a variable of the same name, SAS issues a warning message and uses the shorter, original length of the variable. Because SAS uses the shorter length, data might be truncated.

When you intentionally truncate data, perhaps to remove unnecessary blanks from character variables, SAS issues a warning message that might not be useful to you. To make it so that SAS does not issue the warning message or set a nonzero return code, you can set the VARLENCHK= system option to NOWARN. When VARLENCHK=NOWARN, SAS does not issue a warning message and sets the return code to SYSRC=0.

Alternatively, if you set VARLENCHK=ERROR and the length of a variable that is being read is larger than the length that is defined for the variable, SAS issues an error and sets the return code SYSRC=8.

The VARLENCHK= system option does not have any effect on BY variables named in a BY statement that follows a SET, MERGE, or UPDATE statement. The VARLENCHK= option applies only to variables with the same name that have different lengths in two or more data sets. BY variables are excluded by design.

Note: When a BY variable has different lengths in two or more data sets, a separate warning message is produced, which is the correct behavior.

WARNING: Multiple lengths were specified for the BY variable x by input data sets. This may cause unexpected results.

To avoid this warning message, you can specify the LENGTH statement prior to the SET, MERGE, or UPDATE statement to set the BY variable to the same length.

Examples

Example 1: SAS Issues a Warning Message Merging Two Data Sets with Different Variable Lengths

This example merges two data sets, the `sashelp.class` data set and the `exam_schedule` data set. The length of the variable `Name` is set to 8 by the first SET statement, `set sashelp.class;`. The `exam_schedule` data set sets the length of `Name` to 10. When `exam_schedule` is read in the second SET statement, `set exam_schedule key=Name;`, SAS issues a warning message because the length of `Name` in the `exam_schedule` data set is longer than the length of `Name` in the `sashelp.class` data set, and data might have been truncated.

```

/* Create the exam_schedule data set. */
data exam_schedule(index=(Name));
  input Name : $10. Exam_Date : mmddyy10.;
  format Exam_Date mmddyy10.;
datalines;
Carol      06/09/2016
Hui        06/09/2016
Janet      06/09/2016
Geoffrey   06/09/2016
John       06/09/2016
Joyce      06/09/2016
Helga      06/09/2016
Mary       06/09/2016
Roberto    06/09/2016
Ronald     06/09/2016
Barbara    06/10/2016
Louise     06/10/2016
Alfred     06/11/2016
Alice      06/11/2016
Henri      06/11/2016
James      06/11/2016
Philip     06/11/2016
Tomas      06/11/2016
William    06/11/2016
;
run;

/* Merge the data sets sashelp.class and exam_schedule */
data exams;
  set sashelp.class;
  set exam_schedule key=Name;
run;

```

Here is the SAS log shows the warning message:

Output 2.8 The Warning Message in the SAS Log

```

83      /* Merge the data sets sashelp.class and exam_schedule */
84      data exams;
85          set sashelp.class;
86          set exam_schedule key=Name;
87          run;

WARNING: Multiple lengths were specified for the variable Name by input data
set(s). This can cause truncation of data.
Name=Henry Sex=M Age=14 Height=63.5 Weight=102.5 Exam_Date=06/09/2016
_ERROR_=1 _IORC_=1230015 _N_=5
Name=Jane Sex=F Age=12 Height=59.8 Weight=84.5 Exam_Date=06/11/2016 _ERROR_=1
_IORC_=1230015 _N_=7
Name=Jeffrey Sex=M Age=13 Height=62.5 Weight=84 Exam_Date=06/09/2016 _ERROR_=1
_IORC_=1230015 _N_=9
Name=Judy Sex=F Age=14 Height=64.3 Weight=90 Exam_Date=06/09/2016 _ERROR_=1
_IORC_=1230015 _N_=12
Name=Robert Sex=M Age=12 Height=64.8 Weight=128 Exam_Date=06/11/2016 _ERROR_=1
_IORC_=1230015 _N_=16
Name=Thomas Sex=M Age=11 Height=57.5 Weight=85 Exam_Date=06/09/2016 _ERROR_=1
_IORC_=1230015 _N_=18
NOTE: There were 19 observations read from the data set SASHELP.CLASS.

```

Example 2: Turn Off the Warning Message and Use the LENGTH Statement to Match Variable Lengths

In order to merge the two data sets, `sashelp.class` and `exam_schedule`, you can examine the values of `Name` in `exam_schedule`. You can see that there are no values that are greater than 8 and that you can change the length of `Name` without losing data.

To change the length of the variable `Name`, you use a `LENGTH=` statement in a `DATA` step before the `set exam_schedule;` statement. If the value of `VARLENCHK` is `WARN` (the default), SAS issues the warning message that the value of `Name` is truncated when it is read from `work.exam_schedule`. Because you know that data is not lost, you might want to turn the warning message off:

```

options varlenchk=nowarn;
data exam_schedule(index=(Name));
    length Name $ 8;
    set exam_schedule;
run;

```

Here is the SAS log output:

```

56      options varlenchk=nowarn;
57      data exam_schedule(index=(Name));
58          length Name $ 8;
59          set exam_schedule;
60          run;

NOTE: There were 19 observations read from the data set WORK.EXAM_SCHEDULE.
NOTE: The data set WORK.EXAM_SCHEDULE has 19 observations and 2 variables.

```

VBUFSIZE= System Option

Specifies the size of the view buffer.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Categories:	Input Control: Data Processing System Administration: Performance
PROC OPTIONS GROUP=	INPUTCONTROL PERFORMANCE
Default:	The shipped default is 65536.
Restriction:	The VBUFSIZE= system option does not apply to SQL views.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

VBUFSIZE=*n* | *nK* | *nM* | *nG* | *nT* | *hexX* | MIN | MAX

Note: You can also use the KB, MB, GB, and TB syntax notations.

Required Arguments

n | *nK* | *nM* | *nG* | *nT*

specifies the size of the view buffer in multiples of 1 (bytes); 1,024 (kilobytes); 1,048,576 (megabytes); 1,073,741,824 (gigabytes); or 1,099,511,627,776 (terabytes). For example, a value of **8** specifies 8 bytes, and a value of **3m** specifies 3,145,728 bytes.

hexX

specifies the size of the view buffer as a hexadecimal value. You must specify the value beginning with a number (0–9), followed by an X. For example, the value **0ffffeX** specifies a buffer size of 65,534 bytes.

MIN

sets the minimum number of buffers to 0.

MAX

sets the view buffer size to $2^{63}-1$, or approximately 9.2 quintillion bytes.

Note If you set VBUFSIZE=MAX and your system does not have enough memory, SAS stops processing the view.

Details

The view buffer is a segment of memory that is allocated to hold output observations that are generated for a view. The size of the buffer determines how much data can be held in memory at one time.

The view buffer is shared between the request that opens the view (for example, a SAS procedure) and the view itself. Two computer tasks coordinate between requesting data and generating and returning the data as follows:

- When a request task (such as a PRINT procedure) requests data, task switching occurs from the request task to the view task in order to execute the view and generate the observations. The view fills the view buffer with as many observations as possible.

- When the view buffer is full, task switching occurs from the view task back to the request task in order to return the requested data. The observations are cleared from the view buffer.

The optimal value for buffer system options is dependent on your operating environment. Experiment with various buffers sizes to determine the optimal value for these system options.

The size of the view buffer and the size of an observation determine how many observations can be held in the buffer. To determine the observation length, use PROC CONTENTS for the view. The number of observations then determines how many times the computer must switch between the request task and the view task. The larger the view buffer is, the less task switching is needed to process a view, which can speed up execution time.

To improve efficiency, first determine how many observations fit into the default buffer size. Then, set the view buffer so that it can hold more generated observations.

If OBSBUF= is set for a view, SAS uses the value of OBSBUF= and not the value of VBUFSIZE= to determine the size of the view buffer.

The view buffer is released when the view completes execution.

Comparisons

The VBUFSIZE= system option enables you to specify the size of the view buffer based on a number of bytes. The number of observations that can be read into the view buffer at one time is the value of VBUFSIZE= divided by the length of the observation. VBUFSIZE= is a system option and is set for the length of a SAS session.

The OBSBUF= data set option sets the view buffer size based on a specified number of observations that can be read into the view buffer at one time. The size of the view buffer is determined by the value of OBSBUF= multiplied by the length of the observation. OBSBUF= is a data set option and is set for the length of processing a view.

See Also

Data Set Options:

- [“OBSBUF= Data Set Option” in SAS Viya Data Set Options: Reference](#)

VERBOSE System Option

Specifies whether SAS writes the system option settings to the SAS log.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Log and Procedure Output Control: SAS Log
PROC OPTIONS GROUP=	LOGCONTROL
Default:	NOVERBOSE
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

VERBOSE | NOVERBOSE

Required Arguments

VERBOSE

writes the settings of SAS system options from the configuration file, the SAS command, and the SASV9_OPTIONS environment variable to the SAS log. For the CONFIG option, VERBOSE lists the names of the configuration files.

NOVERBOSE

does not write the settings of the system options to the SAS log.

Details

SAS creates a list of system options with their values and where the options were set. This list is first written to a global journal file, and then it is written to the SAS log. The advantage of writing to a global journal file is that if SAS fails to initialize, output is still available, even though a SAS log was not created.

See Also

System Options:

- [“OPLIST System Option” on page 149](#)

Other References:

- [“Customizing Your SAS Session By Using System Options” in *Batch and Line Mode Processing in SAS Viya*](#)

VNFERR System Option

Specifies whether SAS issues an error or warning when a BY variable exists in one data set but not another data set when the other data set is _NULL_. This option applies when processing the SET, MERGE, UPDATE, or MODIFY statements.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Error Handling

PROC OPTIONS GROUP= ERRORHANDLING

Default: The shipped default is VNFERR.

Restriction: This option is not valid in the CAS server.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

VNFERR | NOVNFERR

Syntax Description

VNFERR

specifies that SAS issue an error when a BY variable exists in one data set but not in another data set when the other data set is `_NULL_`. This option applies when processing the SET, MERGE, UPDATE, or MODIFY statements. When the error occurs, SAS enters into syntax-check mode.

NOVNFERR

specifies that SAS issue a warning when a BY variable exists in one data set but not in another data set when the other data set is `_NULL_`. This option applies when processing the SET, MERGE, UPDATE, or MODIFY statements. When the warning occurs, SAS does not enter into syntax-check mode.

Details

VNF stands for variable not found.

This option is useful when macro variables store data set names and these macro variables are used by the SET, MERGE, UPDATE, or MODIFY statements. If you set NOVNFERR and one of these statements contains a macro variable with a value `_NULL_`, SAS issues a warning instead of an error and processing continues.

Comparisons

- VNFERR is similar to the BYERR system option, which issues an error and enters into syntax-check mode if the SORT procedure attempts to sort a `_NULL_` data set.
- VNFERR is similar to the DSNFERR system option, which issues an error when a SAS data set is not found.

Examples

Example 1

This example shows the results of setting the VNFERR option and the NOVNFERR option:

```

/* treat variable not found on _NULL_ SAS data set as an error */

/* turn option off - should not get an error */
options novnferr; run;

data a;
  x = 1;
  y = 2;
run;

data b;
  x = 2;
  y = 3;
run;

data _null_;
  y = 2;
run;

/* option is off - should not get an error */

```

```

data result;
  merge a b _null_;
  by x;
run;

/* turn option on - should get an error */
options vnferr; run;

data result2;
  merge a b _null_;
  by x;
run;

```

Log 2.4 The SAS Log with Output for the VMFERR and NOVNFERR Options

```

66 /* treat variable not found on _NULL_ SAS data set as an error */
67
68 /* turn option off - should not get an error */
69 options novnferr; run;
70
71 data a;
72   x = 1;
73   y = 2;
74 run;

NOTE: The data set WORK.A has 1 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time          0.01 seconds
      cpu time           0.00 seconds

75
76 data b;
77   x = 2;
78   y = 3;
79 run;

NOTE: The data set WORK.B has 1 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds

80
81 data _null_;
82   y = 2;
83 run;

NOTE: The data set WORK._NULL has 1 observations and 1 variables.
NOTE: DATA statement used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds

84
85 /* option is off - should not get an error */
86 data result;
87   merge a b _null_;
88   by x;
89 run;

```



```

WARNING: BY variable x is not on input data set WORK._null_.
NOTE: There were 1 observations read from the data set WORK.A.
NOTE: There were 1 observations read from the data set WORK.B.
NOTE: The data set WORK.RESULT has 2 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds

90
91  /* turn option on - should get an error          */
92  options vnferr; run;
93
94  data result2;
95    merge a b _null_;
96    by x;
97  run;

ERROR: BY variable x is not on input data set WORK._null_.
NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.RESULT2 may be incomplete.  When this step was
stopped there were 0 observations and 2
      variables.

```

Example 2

In this example, the data set, Result, reads from three data sets by using the SET statement. The SET statement values are all macro variables. One of these macro variables, &dataset3, has a value of _NULL_. SAS issues a warning message when it reads &dataset3; and completes the DATA step without an error.

```

options novnferr;

data a;
  x = 1;
  y = 2;
run;

data b;
  x = 2;
  y = 3;
run;

%let dataset1=a;
%let dataset2=b;
%let dataset3=_null_;

data result;
  set &dataset1 &dataset2 &dataset3;
  by x;
run;

```

Log 2.5 The SAS Log Showing a Warning Message for a NULL Value

```

15  options novnferr;
16
17  data a;
18      x = 1;
19      y = 2;
20  run;

NOTE: The data set WORK.A has 1 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time          0.01 seconds
      cpu time           0.01 seconds

21  data b;
22      x = 2;
23      y = 3;
24  run;

NOTE: The data set WORK.B has 1 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds

25
26  %let dataset1=a;
27  %let dataset2=b;
28  %let dataset3=_null_;
29
30  data result;
31      set &dataset1 &dataset2 &dataset3;
32      by x;
33  run;

WARNING: BY variable x is not on input data set WORK._null_.
NOTE: There were 1 observations read from the data set WORK.A.
NOTE: There were 1 observations read from the data set WORK.B.

```

See Also**System Options:**

- [“BYERR System Option” on page 48](#)
- [“DSNFERR System Option” on page 74](#)

WORK System Option

Specifies the location of the Work library.

Valid in: Configuration file, SAS command, SASV9_OPTIONS environment variable

Category: Environment Control: Files

PROC OPTIONS ENVFILES

GROUP=

Default: Set in the installed !SASROOT/sasv9.cfg file

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

WORK *filename* | *directory*

Required Arguments

filename

specifies a file that contains a list of directories and optional keywords. SAS chooses a directory from the list in the file as the location for the Work library for the current SAS session.

directory

specifies a directory as the location for the Work library for the current SAS session.

Details

The Basics

If you use the *filename* option, SAS opens the file, and selects one of the directories to use as the location for the Work library. SAS either randomly selects a directory or selects a directory based on available space. You use the METHOD keyword to make your selection.

If you use the *directory* option, SAS continues its initialization using the specified directory as the location for the Work library.

Making the Allocation of Work Libraries More Dynamic

The *filename* option contains a list of directories that can be used for the Work library. Individual SAS Work libraries still reside in a single directory. You use METHOD=RANDOM to specify that the directory for the Work library is randomly chosen from the list of directories. SAS selects one directory per session as the location for the Work library. This selection enables you to balance the I/O load across multiple hardware systems. You use METHOD=SPACE to specify the directory that has the most available space. If the METHOD keyword is not specified, SAS defaults to randomly selecting a directory.

Examples

Example 1: Spreading a Processing Load across Multiple Volumes of Different Disks

The following example shows how to spread an I/O processing load across multiple volumes of different disks. In this case, you use METHOD=RANDOM. A file named `/sasinfo/workfiles` contains the following information:

```
/disk1/sastempfiles  
/disk2/sastempfiles  
/disk3/sastempfiles  
method=random
```

The Work library for a particular SAS session is placed on either disk1, disk2, or disk3. The configuration file or command line would include the following:

```
-work /sasinfo/workfiles
```

Example 2: Choosing the Directory That Has the Most Available Space

When you process your data, you can select the directory that has the most available space. In this case, you use METHOD=SPACE. In the following example, `/sasinfo/workfiles` contains the following directories:

```
/disk1/sastempfiles
/disk2/sastempfiles
/disk3/sastempfiles
method=space
```

The Work library is placed on the disk with the most available space.

See Also**System Options:**

- [“WORKINIT System Option” on page 230](#)

WORKINIT System Option

Specifies whether to initialize the Work library at SAS invocation.

Valid in:	Configuration file, SAS command, SASV9_OPTIONS environment variable
Category:	Environment Control: Files
PROC OPTIONS GROUP=	ENVFILES
Default:	The shipped default is WORKINIT.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

WORKINIT | **NOWORKINIT**

Syntax Description**WORKINIT**

specifies that a new subdirectory is to be created in the directory that is specified in the WORK option.

NOWORKINIT

specifies that the system is to use the directory specified by the WORK option.

- If the system does not find any old subdirectories, it creates a new one.
- If the system finds more than one old subdirectory, it uses the latest one.
- If file locking is in effect (see [“FILELOCKS System Option” on page 81](#)), the system looks for the latest unlocked directory. If the system does not find the latest unlocked directory, it creates a new one.

Details

The WORKINIT system option initializes the Work data library and erases all files from a previous SAS session at SAS invocation. The WORKTERM system option controls whether SAS erases Work files at the end of a SAS session.

See Also

System Options:

- [“FILELOCKS System Option” on page 81](#)
- [“WORKTERM System Option” on page 231](#)

WORKPERMS System Option

Sets the permissions of the SAS Work library when it is initially created.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Environment Control: Files

PROC OPTIONS GROUP= ENVFILES

Default: The shipped default is 700.

Note: This option can be restricted by a site administrator. For more information, see [“Restricted Options” on page 8](#).

Syntax

WORKPERMS *permission-value*

Required Argument

permission-value

specifies the octal value representing the permissions for the SAS Work directory. Values can be any octal value setting the permission of a Linux directory. Examples of values include umask, 700, 755, 770, 775, and 777.

Details

The WORKPERMS system option enables you to change or remove the current file mode creation mask value when you initially create a SAS Work library. This means that you can change the value of *permission-value* to change file permissions for a new Work library.

WORKTERM System Option

Specifies whether to erase the Work files when SAS terminates.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category:	Environment Control: Files
PROC OPTIONS GROUP=	ENVFILES
Default:	The shipped default is WORKTERM.
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

WORKTERM | **NOWORKTERM**

Syntax Description

WORKTERM

erases the Work files at the termination of a SAS session.

NOWORKTERM

does not erase the Work files.

Details

Although NOWORKTERM prevents the Work data sets from being deleted, it has no effect on initialization of the Work library by SAS. SAS normally initializes the Work library at the start of each session, which effectively destroys any pre-existing information.

Comparisons

Use the NOWORKINIT system option to prevent SAS from erasing existing Work files on invocation. Use the NOWORKTERM system option to prevent SAS from erasing existing Work files on termination.

See Also

System Options:

- [“WORKINIT System Option” on page 230](#)

XCMD System Option

Specifies whether the X command is valid in the SAS session.

Valid in:	Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable
Category:	Environment Control: Display
PROC OPTIONS GROUP=	ENVDISPLAY
Default:	XCMD
Note:	This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8 .

Syntax

XCMD | NOXCMD

Required Arguments

XCMD

specifies that the X command is valid in the current SAS session.

NOXCMD

specifies that the X command is not valid in the current SAS session.

Details

The XCMD system option specifies whether the X command is valid in the current SAS session.

You cannot use several SAS statements, objects, or facilities if you use the NOXCMD system option. Examples of these statements, objects, and facilities include the following:

- the PIPE device type in the FILENAME statement
- the CALL SYSTEM routine
- the %SYSEXEC macro
- any facility that SAS uses to execute a shell-level command

See Also

Commands:

- [“X Command: Linux”](#) in *Batch and Line Mode Processing in SAS Viya*

Macros:

- [“%SYSEXEC Statement”](#) in *SAS Viya Macro Language: Reference*

Other References:

- [“Executing Operating System Commands from Your SAS Session”](#) in *Batch and Line Mode Processing in SAS Viya*

YEARCUTOFF= System Option

Specifies the first year of a 100-year span that is used by date informats and functions to read a two-digit year.

Valid in: Configuration file, SAS command, OPTIONS statement, SASV9_OPTIONS environment variable

Category: Input Control: Data Processing

PROC OPTIONS GROUP= INPUTCONTROL

Default: The shipped default is 1926.

Note: This option can be restricted by a site administrator. For more information, see “Restricted Options” on page 8.

Syntax

YEARCUTOFF=*nnnn* | *nnnnn*

Syntax Description

nnnn | *nnnnn*

specifies the first year of the 100-year span.

Range 1582–19900

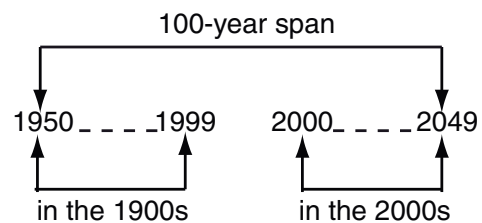
Details

The YEARCUTOFF= value is the default that is used by various date and datetime informats and functions.

If the default value of *nnnn* (1926) is in effect, the 100-year span begins with 1926 and ends with 2025. Therefore, any informat or function that uses a two-digit year value that ranges from 26 to 99 assumes a prefix of 19. For example, the value 92 refers to the year 1992.

The value that you specify in YEARCUTOFF= can result in a range of years that span two centuries. For example, if you specify YEARCUTOFF=1950, any two-digit value between 50 and 99 inclusive refers to the first half of the 100-year span, which is in the 1900s. Any two-digit value between 00 and 49, inclusive, refers to the second half of the 100-year span, which is in the 2000s. The following figure illustrates the relationship between the 100-year span and the two centuries if YEARCUTOFF=1950.

Figure 2.1 A 100-Year Span with Values in Two Centuries



Note: YEARCUTOFF= has no effect on existing SAS dates or dates that are read from input data that include a four-digit year, except years with leading zeros. For example, 0076 with yearcutoff=1990 indicates 2076.

Part 3

SAS Functions and Statements That Process SAS System Options

Chapter 3

SAS Functions That Process SAS System Options 237

Chapter 4

SAS Statements That Process SAS System Options 243

Chapter 3

SAS Functions That Process SAS System Options

Dictionary	237
GETOPTION Function	237

Dictionary

GETOPTION Function

Returns the value of a SAS system option.

Category: Special

Restriction: This function is not valid in the CAS server.

Syntax

GETOPTION(*option-name* <, *return-value-option*> <*return-value-formatting-options*>)

Required Argument

option-name

is a character constant, variable, or expression that specifies the name of the system option.

Tips Do not put an equal sign after the name. For example, write PAGESIZE= as PAGESIZE.

SAS options that are passwords, such as METAPASS, return the value **xxxxxxxx**, and not the actual password.

Return Value Options

DEFAULTVALUE

returns the default option value.

Restriction DEFAULTVALUE is valid only for SAS system options. SAS issues a warning message when the DEFAULTVALUE option is specified and *option-name* is a graphics option.

HOWSCOPE

returns a character string that specifies the scope of an option.

Restriction HOWSCOPE is valid only for SAS system options. SAS issues a warning message when the HOWSCOPE option is specified and *option-name* is a graphics option.

HOWSET

returns a character string that specifies how an option value was set.

Restriction HOWSET is valid only for SAS system options. SAS issues a warning message when the HOWSET option is specified and *option-name* is a graphics option.

STARTUPVALUE

returns the system option value that was used to start SAS either on the command line or in a configuration file.

Restriction STARTUPVALUE is valid only for SAS system options. SAS issues a warning message when the STARTUPVALUE option is specified and *option-name* is a graphics option.

Return Value Formatting Options**CM**

reports graphic units of measure in centimeters.

Restriction CM is valid only for graphics options and the following SAS system options: BOTTOMMARGIN, TOPMARGIN, RIGHTMARGIN, and LEFTMARGIN. SAS writes a note to the log when the CM option is specified and *option-name* is not a graphics option or an option that specifies a margin value.

EXPAND

for options that contain environment variables, returns the option value with the value of the environment variable.

Restriction EXPAND is valid only for character system option values. EXPAND is ignored if *option-name* has an option type of Boolean, such as CENTER or NOCENTER, or if the value of the option is numeric.

Note SAS issues a note when EXPAND is specified for Boolean options and for options that have numeric values. SAS issues a warning when EXPAND is specified and the option is a graphics option.

Tip By default, some option values are displayed with expanded variable values. Other options require the EXPAND option in the PROC OPTIONS statement. Use the DEFINE option in the PROC OPTIONS statement to determine whether an option value expands variables by default or if the EXPAND option is required. If the output from PROC OPTIONS DEFINE shows the following information, you must use the EXPAND option to expand variable values:

Expansion: Environment variables, within the option value,
are not expanded

KEYEXPAND

for options that contain environment variables, returns the value in the format **option-name=value**.

Restriction KEYEXPAND is valid only for character system option values. SAS issues an error message when the KEYEXPAND option is specified and *option-name* is a graphics option. KEYEXPAND is ignored if *option-name* has an option type of Boolean, such as CENTER or NOCENTER, or if the value of the option is numeric.

KEYWORD

returns option values in a **option-name=value** format that would be suitable for direct use in the SAS OPTIONS or GOPTIONS global statements.

Restrictions KEYWORD is not valid when it is used with the HEXVALUE, EXPAND, KEYEXPAND, or LOGNUMBERFORMAT options. SAS writes a note to the log when the GETOPTION function contains conflicting options.

KEYWORD is valid only for character or numeric system option values. KEYWORD is ignored for system options whose option type is Boolean, such as CENTER or NOCENTER. SAS issues an error message when the KEYWORD option is specified and *option-name* is a graphics option.

Note For a system option with a null value, the GETOPTION function returns a value of ' ' (single quotation marks with a blank space between them). An example is EMAILID=' '.

HEXVALUE

returns the option value as a hexadecimal value.

Restriction HEXVALUE is valid only for character or numeric system option values. If HEXVALUE is specified for system options whose option type is Boolean, such as CENTER or NOCENTER, or if *option-name* is a graphics option, SAS issues an error message.

IN

reports graphic units of measure in inches.

Restriction IN is valid only for graphics options and the following SAS system options: BOTTOMMARGIN, TOPMARGIN, RIGHTMARGIN, and LEFTMARGIN. SAS writes a note to the log when the IN option is specified and *option-name* is not a graphics option or an option that specifies a margin value.

LOGNUMBERFORMAT

formats SAS system option values using locale-specific punctuation.

Restriction Do not use LOGNUMBERFORMAT if the returned value is used to set an option value by using the OPTIONS statement. The OPTIONS statement does not accept commas in numeric values.

Examples

Example 1: Using GETOPTION to Save and Restore the YEARCUTOFF Option

This example saves the value of the YEARCUTOFF option, processes SAS statements based on the value of the YEARCUTOFF option, and then resets the value to 1926 if it is not already 1926.

```
/* Save the value of the YEARCUTOFF system option */
%let cutoff=%sysfunc(getoption(yearcutoff,keyword));

data ages;
  if getoption('yearcutoff') = '1926' then
    do;
      ...more SAS statements...
    end;
  else do;
      ...more SAS statements...
      /* Reset YEARCUTOFF */
      options &cutoff;
    end;
run;
```

Example 2: Using GETOPTION to Obtain Different Reporting Options

This example defines a macro to illustrate the use of the GETOPTION function to obtain the value of system and graphics options by using different reporting options.

```
%macro showopts;
  %put PAGESIZE= %sysfunc(
    getoption(PAGESIZE));
  %put PAGESIZESETBY= %sysfunc(
    getoption(PAGESIZE, HOWSET));
  %put PAGESIZESCOPE= %sysfunc(
    getoption(PAGESIZE, HOWSCOPE));
  %put FORMCHAR= %sysfunc(
    getoption(FORMCHAR));

%mend;
options PAGESIZE=67;
%showopts
```

Here is the SAS log:

```

56      %macro showopts;
57          %put PAGESIZE= %sysfunc(
58              getoption(PAGESIZE));
59          %put PAGESIZESETBY= %sysfunc(
60              getoption(PAGESIZE, HOWSET));
61          %put PAGESIZESCOPE= %sysfunc(
62              getoption(PAGESIZE, HOWSCOPE));
63          %put FORMCHAR= %sysfunc(
64              getoption(FORMCHAR));
65
66      %mend;
67      options PAGESIZE=67;
68      %showopts
PAGESIZE= 67
PAGESIZESETBY= Options Statement
PAGESIZESCOPE= IOM ROOT COMP ENV
FORMCHAR= |---+|---+=|-\<>*
69          %put %sysfunc(getoption(papersize,keyword));
PAPERSIZE=LETTER

```

Example 3: Returning Default Values

The following code is an example of setting the DATESTYLE= system option to YMD and then to its default value:

```

/* Check the value of datestyle before we change it.          */
/*
/* The current value is DMY as this value was set in the      */
/* autoexec file when SAS Studio started.                    */
/*
      %put %sysfunc(getoption(datestyle,keyword));

/* Change the DATESTYLE value and check the change.          */
/*
      options datestyle='YMD';

      %put %sysfunc(getoption(datestyle,keyword));

/* Change DATESTYLE back to the default value and check it. */
/* RESULT:  MDY                                             */
/*
      %let defstyl = %sysfunc(getoption(datestyle,keyword,defaultvalue)) ;
      options &defstyl; run;
      %put %sysfunc(getoption(datestyle,keyword));

```

The SAS log displays the following lines:

```
5
56      /* Check the value of datestyle before we change it.    */
57
58      /* The current value is DMY as this value was set in the    */
59      /* autoexec file when SAS Studio started.                */
60
61      %put %sysfunc(getoption(datestyle,keyword));
DATESTYLE=DMY
62
63      /* Change the DATESTYLE value and check the change.      */
64
65      options datestyle='YMD';
66
67      %put %sysfunc(getoption(datestyle,keyword));
DATESTYLE=YMD
68
69      /* Change DATESTYLE back to the default value and check it. */
70      /* RESULT: MDY                                           */
71
72      %let defstyl =
%sysfunc(getoption(datestyle,keyword,defaultvalue)) ;
73      options &defstyl; run;
74      %put %sysfunc(getoption(datestyle,keyword));
DATESTYLE=MDY
75
```


Chapter 4

SAS Statements That Process SAS System Options

Dictionary	243
OPTIONS Statement	243

Dictionary

OPTIONS Statement

Specifies or changes the value of one or more SAS system options.

Valid in:	Anywhere
Category:	Program Control
Restriction:	This statement is not valid on the CAS server.

Syntax

OPTIONS *option(s)*;

Arguments

option

specifies one or more SAS system options to be changed.

Details

The change that is made by the OPTIONS statement remains in effect for the rest of the job, session, SAS process, or until you issue another OPTIONS statement to change the options again. You can specify SAS system options through the OPTIONS statement, at SAS invocation, and at the initiation of a SAS process.

If you attempt to set an option that is restricted by your site administrator, SAS issues a note that the option is restricted and cannot be changed. For more information, see [“Restricted Options” on page 8](#).

Note: If you want a particular group of options to be in effect for all your SAS jobs or sessions, store an OPTIONS statement in an autoexec file or list the system options in a configuration file or custom_option_set.

Note: For a system option with a null value, the GETOPTION function returns a value of ' ' (single quotation marks with a blank space between them), for example, **EMAILID= ' '**. This GETOPTION value can then be used in the OPTIONS statement.

An OPTIONS statement can appear at any place in a SAS program, except within data lines.

Example: Changing the Value of a System Option

This example suppresses the date that is normally written to SAS output and sets a line size of 72:

```
options nodate linesize=72;
```

See Also

[“Definition of System Options” on page 3](#)

Part 4

SAS Procedures That Process SAS System Options

<i>Chapter 5</i>	
OPTIONS Procedure	247

Chapter 5

OPTIONS Procedure

Overview: OPTIONS Procedure	247
Syntax: OPTIONS Procedure	248
PROC OPTIONS Statement	248
Displaying a List of System Options	252
Displaying Information about One or More Options	253
Displaying Information about System Option Groups	255
Displaying Restricted Options	257
Results: OPTIONS Procedure	258
Examples: OPTIONS Procedure	259
Example 1: Producing the Short Form of the Options Listing	259
Example 2: Displaying the Setting of a Single Option	259
Example 3: Displaying Expanded Path Environment Variables	261
Example 4: List the Options That Can Be Specified by the INSERT and APPEND Options	262

Overview: OPTIONS Procedure

The OPTIONS procedure lists the current settings of SAS system options in the SAS log.

SAS system options control how SAS formats output, handles files, processes data sets, interacts with the operating environment, and does other tasks that are not specific to a single SAS program or data set. You use the OPTIONS procedure to obtain information about an option or a group of options. Here is some of the information that the OPTIONS procedure provides:

- the current value of an option and how it was set
- a description of an option
- valid syntax for the option, valid option values, and the range of values
- where you can set the system option
- if the option can be restricted by your site administrator
- if the option has been restricted
- system options that belong to a system option group

- system options that are specific for an operating environment
- if an option value has been modified by the INSERT or APPEND system options

For additional information about SAS system options, see [SAS Viya System Options: Reference](#).

Syntax: OPTIONS Procedure

PROC OPTIONS <option(s)>;

Statement	Task	Example
PROC OPTIONS	List the current system option settings to the SAS Log	Ex. 1, Ex. 2, Ex. 3, Ex. 4

PROC OPTIONS Statement

Lists the current settings of SAS system options in the SAS log.

Examples: [“Example 1: Producing the Short Form of the Options Listing” on page 259](#)
[“Example 2: Displaying the Setting of a Single Option” on page 259](#)
[“Example 3: Displaying Expanded Path Environment Variables” on page 261](#)
[“Example 4: List the Options That Can Be Specified by the INSERT and APPEND Options” on page 262](#)

Syntax

PROC OPTIONS <option(s)>;

Summary of Optional Arguments

LISTGROUPS

lists the system option groups as well as a description of each group.

Choose the format of the listing

DEFINE

displays the short description of the option, the option group, and the option type.

EXPAND

when displaying a character option, replaces an environment variable in the option value with the value of the environment variable. EXPAND is ignored if the option is a Boolean option, such as CENTER or NOCENTER, or if the value of the option is numeric.

HEXVALUE

displays system option character values as hexadecimal values.

LOGNUMBERFORMAT

displays numeric system option values using locale-specific punctuation.

LONG

lists each system option on a separate line with a description.

NOEXPAND

when displaying a path, displays the path using environment variable(s) and not the value of the environment variable(s). This is the default.

NOLOGNUMBERFORMAT

displays numeric system option values without using punctuation, such as a comma or a period. This is the default.

SHORT

specifies to display a compressed listing of options without descriptions.

VALUE

displays the option's value and scope, as well as how the value was set.

Restrict the number of options displayed

GROUP=group-name

GROUP=(group-name-1 ... group-name-n)

displays the options in one or more groups specified by *group-name*.

HOST

displays only host options.

LISTINSERTAPPEND

lists the system options whose value can be modified by the INSERT and APPEND system options.

LISTRESTRICT

lists the system options that can be restricted by your site administrator.

NOHOST

displays only portable options.

OPTION=option-name

OPTION=(option-name-1 ... option-name-n)

displays information about one or more system options.

RESTRICT

displays system options that the site administrator has restricted from being updated.

Optional Arguments**DEFINE**

displays the short description of the option, the option group, and the option type. SAS displays information about when the option can be set, whether an option can be restricted, and the valid values for the option.

Interaction This option is ignored when SHORT is specified.

Example [“Example 2: Displaying the Setting of a Single Option” on page 259](#)

EXPAND

when displaying a character option, replaces an environment variable in the option value with the value of the environment variable. EXPAND is ignored if the option is a Boolean option, such as CENTER or NOCENTER, or if the value of the option is numeric.

Tip By default, some option values are displayed with expanded variables. Other options require the EXPAND option in the PROC OPTIONS statement. Use the DEFINE option in the PROC OPTIONS statement to

determine whether an option value expands variables by default or if the EXPAND option is required. If the output from PROC OPTIONS DEFINE shows the following information, you must use the EXPAND option to expand variable values:

Expansion: Environment variables, within the option value,
are not expanded

See [“NOEXPAND” on page 251](#) option to view paths that display the environment variable

Example [“Example 3: Displaying Expanded Path Environment Variables” on page 261](#)

GROUP=group-name

GROUP=(group-name-1 ... group-name-n)

displays the options in one or more groups specified by *group-name*.

Requirement When you specify more than one group, enclose the group names in parenthesis and separate the group names by a space.

See [“Displaying Information about System Option Groups” on page 255](#)

HEXVALUE

displays system option character values as hexadecimal values.

HOST

displays only host options.

See [“NOHOST” on page 251](#) option to display only portable options.

LISTINSERTAPPEND

lists the system options whose value can be modified by the INSERT and APPEND system options. The INSERT option specifies a value that is inserted as the first value of a system option value list. The APPEND option specifies a value that is appended as the last value of a system option value list. Use the LISTINERTAPPEND option to display which system options can have values inserted at the beginning or appended at the end of their value lists.

See [“INSERT= System Option” on page 99](#) and

[“APPEND= System Option” on page 39](#)

Example [“Example 4: List the Options That Can Be Specified by the INSERT and APPEND Options” on page 262](#)

LISTGROUPS

lists the system option groups as well as a description of each group.

See [“Displaying Information about System Option Groups” on page 255](#)

LISTRESTRICT

lists the system options that can be restricted by your site administrator.

See [“RESTRICT” on page 251](#) option to list options that have been restricted by the site administrator

LONG

lists each system option on a separate line with a description. This is the default. Alternatively, you can create a compressed listing without descriptions.

See [“SHORT” on page 252](#) option to produce a compressed listing without descriptions

Example [“Example 1: Producing the Short Form of the Options Listing” on page 259](#)

LOGNUMBERFORMAT

displays numeric system option values using locale-specific punctuation.

See [“NOLOGNUMBERFORMAT” on page 251](#) option to display numeric option values without using commas

Example [“Example 2: Displaying the Setting of a Single Option” on page 259](#)

NOEXPAND

when displaying a path, displays the path using environment variable(s) and not the value of the environment variable(s). This is the default.

See [“EXPAND” on page 249](#) option to display a path by expanding the value of environment variables

NOHOST

displays only portable options.

Alias PORTABLE or PORT

See [“HOST” on page 250](#) option to display only host options

NOLOGNUMBERFORMAT

displays numeric system option values without using punctuation, such as a comma or a period. This is the default.

See [“LOGNUMBERFORMAT” on page 251](#) option to display numeric system options using commas

OPTION=*option-name***OPTION=(*option-name-1* ... *option-name-n*)**

displays a short description and the value (if any) of the option specified by *option-name*. DEFINE and VALUE options provide additional information about the option.

option-name

specifies the option to use as input to the procedure.

Requirement If a SAS system option uses an equal sign, such as PAGESIZE=, do not include the equal sign when specifying the option to OPTION=.

Example [“Example 2: Displaying the Setting of a Single Option” on page 259](#)

RESTRICT

displays the system options that have been set by your site administrator in a restricted options configuration file. These options cannot be changed by the user. For each option that is restricted, the RESTRICT option displays the option's value, scope, and how it was set.

If your site administrator has not restricted any options, then the following message appears in the SAS log:

```
Your Site Administrator has not restricted any SAS options.
```

See [“LISTRESTRICT” on page 250](#) option to list options that can be restricted by the site administrator

SHORT

specifies to display a compressed listing of options without descriptions.

See [“LONG” on page 251](#) option to create a listing with descriptions of the options.

VALUE

displays the option's value and scope, as well as how the value was set. If the value was set using a configuration file, the SAS log displays the name of the configuration file. If the option was set using the INSERT or APPEND system options, the SAS log displays the value that was inserted or appended.

Interaction This option has no effect when SHORT is specified.

Note SAS options that are passwords, such as METAPASS, return the value XXXXXXXX and not the actual password.

Example [“Example 2: Displaying the Setting of a Single Option” on page 259](#)

Displaying a List of System Options

The log that results from running PROC OPTIONS can show the system options for the options that are available for all operating environment and those that are specific to a single operating environment. Options that are available for all operating environments are referred to as portable options. Options that are specific to a single operating environment are referred to as host options.

The following example shows a partial log that displays the settings of session options. Your listing might differ.

```
proc options;
run;
```

Log 5.1 The SAS Log Showing a Partial Listing of SAS System Options

Portable Options:

APPEND=	Specifies an option=value pair to insert the value at the end of the existing option value.
APPLETLOC= <i>site-specific-path</i>	Specifies the location of Java applets, which is typically a URL.
AUTOCORRECT	Automatically corrects misspelled procedure names and keywords, and global statement names.
AUTOEXEC= <i>/server-path/autoexec.sas</i>	Specifies the location of the SAS AUTOEXEC files.

The log displays both portable and host options when you submit `proc options;`. The host options are specific for the Linux operating environment.

To view only host options, use this version of the `OPTIONS` procedure. Your listing might differ.

```
proc options host;
run;
```

Log 5.2 The SAS Log Showing a Partial List of Host Options

Host Options:

ALIGNSASIOFILES	Aligns SAS files on a page boundary for improved performance.
ALTLOG=	Specifies the location for a copy of the SAS log when SAS is running in batch mode.
ALTPRINT=	Specifies the location for a copy of the SAS procedure output when SAS is running in batch mode.
BLKSIZE=256	Specifies the number of bytes that are read or written in one I/O operation.

Displaying Information about One or More Options

To view the setting of one or more particular options, you can use the `OPTION=` and `DEFINE` options in the `PROC OPTIONS` statement. The following example shows a log that `PROC OPTIONS` produces for a single SAS system option. Your output might differ.

```
proc options option=errorcheck define;
run;
```

Log 5.3 *The Setting of a Single SAS System Option*

```

56      proc options option=errorcheck define; run;

      SAS (r) Proprietary Software Release V.03.02  TS1M0

      ERRORCHECK=NORMAL
      Option Definition Information for SAS Option ERRORCHECK
      Group= ERRORHANDLING
      Group Description: Error messages and error conditions settings
      Description: Specifies whether SAS enters syntax-check mode when errors are found in the LIBNAME,
      FILENAME, %INCLUDE, and LOCK statements.
      Type: The option value is of type CHARACTER
      Maximum Number of Characters: 10
      Casing: The option value is retained uppercased
      Quotes: If present during "set", start and end quotes are removed
      Parentheses: The option value does not require enclosure within parentheses. If present,
      the parentheses are retained.
      Expansion: Environment variables, within the option value, are not expanded
      Number of valid values: 2
      Valid value: NORMAL
      Valid value: STRICT
      When Can Set: Startup or anytime during the SAS Session
      Restricted: Your Site Administrator can restrict modification of this option

```

To view the settings for more than one option, enclose the options in parentheses and separate the options with a space:

```

      proc options option=(append insert) define;
      run;

```

Log 5.4 *The Settings of Two SAS System Options*

```

      SAS (r) Proprietary Software Release V.03.02  TS1M0

      APPEND=
      Option Definition Information for SAS Option APPEND
      Group= ENVFILES
      Group Description: SAS library and file location information
      Description: Specifies an option=value pair to insert the value at the end of the existing option
      value.
      Type: The option value is of type CHARACTER
      Maximum Number of Characters: 32000
      Casing: The option value is retained with original casing
      Quotes: If present during "set", start and end quotes are removed
      Parentheses: The option value does not require enclosure within parentheses. If present,
      the parentheses are retained.
      Expansion: Environment variables, within the option value, are not expanded
      When Can Set: Startup or anytime during the SAS Session
      Restricted: Your Site Administrator cannot restrict modification of this option

```

```

INSERT=
Option Definition Information for SAS Option INSERT
Group= ENVFILES
Group Description: SAS library and file location information
Description: Specifies an option=value pair to insert the value at the beginning of the existing
option value.
Type: The option value is of type CHARACTER
Maximum Number of Characters: 32000
Casing: The option value is retained with original casing
Quotes: If present during "set", start and end quotes are removed
Parentheses: The option value does not require enclosure within parentheses. If present,
the parentheses are retained.
Expansion: Environment variables, within the option value, are not expanded
When Can Set: Startup or anytime during the SAS Session
Restricted: Your Site Administrator cannot restrict modification of this option
    
```

Displaying Information about System Option Groups

Each SAS system option belongs to one or more groups, which are based on functionality, such as error handling or sorting. You can display a list of system-option groups and the system options that belong to one or more of the groups.

Use the LISTGROUPS option to display a list of system-option groups. Your listing might differ.

```

proc options listgroups;
run;
    
```

Log 5.5 List of SAS System Option Groups

```

56          proc options listgroups; run;

SAS (r) Proprietary Software Release V.03.02  TS1M0

Option Groups

GROUP=CAS                CAS Options
GROUP=CODEGEN            Code generation
GROUP=COMMUNICATIONS    Networking and encryption
GROUP=DATACOM            Datacom
GROUP=ENVFILES           Files
GROUP=ERRORHANDLING     Error handling
GROUP=EXECMODES         Initialization and operation
GROUP=EXTFILES           External files
GROUP=INPUTCONTROL      Data Processing
GROUP=INSTALL           Installation
    
```

GROUP=LANGUAGECONTROL	Language control
GROUP=LISTCONTROL	Procedure output
GROUP=LOGCONTROL	SAS log
GROUP=LOG_LISTCONTROL	SAS log and procedure output
GROUP=MACRO	SAS macro
GROUP=MEMORY	Memory
GROUP=ODSPRINT	ODS Printing
GROUP=PDF	PDF
GROUP=PERFORMANCE	Performance
GROUP=SASFILES	SAS Files
GROUP=SECURITY	Security
GROUP=SORT	Procedure options
GROUP=SQL	SQL

Use the `GROUP=` option to display system options that belong to a particular group. You can specify one or more groups.

```
proc options group=(cas memory);  
run;
```

Log 5.6 Sample Output Using the GROUP= Option

```

56      proc options group=(cas memory); run;

      SAS (r) Proprietary Software Release V.03.02 TS1M0

Group=CAS
CASAUTHINFO=      Specifies an authinfo or netrc file that includes authentication information.
CASHOST=cloud.sas.com
                  The CAS server name associated with a CAS session.
CASLIB=           Specify the default CASLIB name.
CASNAME=CASAUTO   Identify the name to associate with a generated CAS session.
CASWORKERS=ALL    Specify the number of workers to use with a CAS session.
CASPORT=5570      The port associated with a CAS session.
CASSESSOPTS=      Identify CAS server session options.
CASTIMEOUT=60     The CAS session timeout in seconds.
CASUSER=          The userid associated with a CAS session.

Group=MEMORY
SORTSIZE=1073741824
                  Specifies the amount of memory that is available to the SORT procedure.
SUMSIZE=0         Specifies a limit on the amount of memory that is available for data summarization
procedures when class
                  variables are active.
MAXMEMQUERY=268435456
                  For certain procedures, specifies the maximum amount of memory that can be
allocated per request.
LOADMEMSIZE=0     Specifies a suggested amount of memory that is needed for executable programs
loaded by SAS.
MEMSIZE=2147483648
                  Specifies the limit on the amount of virtual memory that can be used during a SAS
session.
REALMEMSIZE=0     Specifies the amount of real memory SAS can expect to allocate.

```

You can use the following group names as values for the GROUP= option to list the system options in a group:

ANIMATION	EXECMODES	MEMORY
CAS	EXTFILES	ODSPRINT
CODEGEN	INPUTCONTROL	PDF
COMMUNICATIONS	INSTALL	PERFORMANCE
EMAIL	LANGUAGECONTROL	SASFILES
ENVDISPLAY	LOGCONTROL	SECURITY
ENVFILES	LOG_LISTCONTROL	SORT
ERRORHANDLING	MACRO	SQL

Displaying Restricted Options

Your site administrator can restrict some system options so that your SAS session conforms to options that are set for your site. Restricted options can be modified only by your site administrator. The OPTIONS procedure provides two options that display information about restricted options. The RESTRICT option lists the system options that your site administrator has restricted. The LISTRESTRICT option lists the options that can be restricted by your site administrator. For more information, see [the listing of options that cannot be restricted](#) .

The following SAS logs shows the output when the RESTRICT option is specified and partial output when the LISTRESTRICT option is specified. Your output might differ.

Log 5.7 A List of Options That Have Been Restricted by the Site Administrator

```

1      proc options restrict;
2      run;
      SAS (r) Proprietary Software Release V.03.02 TS1M0

Option Value Information For SAS Option BUFNO
Value: 10
Scope: SAS Session
How option value set: Config File
Config file name:
/opt/sas/viya/SASFoundation/sasv9.cfg

```

Log 5.8 A Partial Log That Lists Options That Can Be Restricted

```

56      proc options listrestrict; run;

      SAS (r) Proprietary Software Release V.03.02 TS1M0

Your Site Administrator can restrict the ability to modify the following Portable Options:

      APPLETLOC      Specifies the location of Java applets, which is typically a URL.
      AUTOCORRECT    Automatically corrects misspelled procedure names and keywords, and global
statement names.
      BINDING        Specifies the binding edge type of duplexed printed output.
      BUFNO          Specifies the number of buffers for processing SAS data sets.
      BUFSIZE        Specifies the size of a buffer page for output SAS data sets.
      BYERR          SAS issues an error message and stops processing if the SORT procedure
attempts to sort a _NULL_ data set.
      BYLINE         Prints the BY line above each BY group.
      BYSORTED       Requires observations in one or more data sets to be sorted in alphabetic or
numeric order.

```

Results: OPTIONS Procedure

SAS writes the options list to the SAS log. SAS system options of the form *option* | **NOoption** are listed as either *option* or **NOoption**, depending on the current setting. They are always sorted by the positive form. For example, NOCAPS would be listed under the Cs.

The OPTIONS procedure displays passwords in the SAS log as eight Xs, regardless of the actual password length.

Examples: OPTIONS Procedure

Example 1: Producing the Short Form of the Options Listing

Features: PROC OPTIONS statement option
SHORT

Details

This example shows how to generate the short form of the listing of SAS system option settings. Compare this short form with the long form that is shown in [“Displaying a List of System Options”](#) on page 252.

Program

```
proc options short;
run;
```

Program Description

List all options and their settings. SHORT lists the SAS system options and their settings without any descriptions. Your output might differ.

```
proc options short;
run;
```

Log

Log 5.9 Partial Listing of the SHORT Option

```
56      proc options short; run;

      SAS (r) Proprietary Software Release V.03.02  TS1M0

      Portable Options:

      APPEND= APPLETLOC=/pathname  AUTOCORRECT AUTOEXEC=/opt/sas/viya/etc/workspaceserver/autoexec.sas
      BINDING=DEFAULT  BOTMOMMARGIN=0.000  IN BUFNO=1  BUFSIZE=0  BYERR  BYLINE  BYSORTED  NOCAPS  NOCARDIMAGE
      CASAUTHINFO=  CASHOST=hostname  CASINSTALL=  CASLIB=  CASLIFETIME=1000000  CASLOGCNTL=NOMETRICS
      CASNAME=CASAUTO  CASNWORKERS=ALL
      CASPORT=7314  CASSESSOPTS=  CASTIMEOUT=60  CASUSER=  CATCACHE=0  CBUFNO=0  CENTER  NOCHARCODE  NOCHKPTCLEAN
```

Example 2: Displaying the Setting of a Single Option

Features: PROC OPTIONS statement option

```

OPTION=
DEFINE
LOGNUMBERFORMAT
VALUE

```

Details

This example shows how to display the setting of a single SAS system option. The log shows the current setting of the SAS system option MEMSIZE. The DEFINE and VALUE options display additional information. The LOGNUMBERFORMAT displays the value using commas.

Program

```

proc options option=memsize define value lognumberformat;
run;

```

Program Description

Specify the MEMSIZE SAS system option. OPTION=MEMSIZE displays option value information. DEFINE and VALUE display additional information. LOGNUMBERFORMAT specifies to format the value using commas.

```

proc options option=memsize define value lognumberformat;
run;

```

Log

Log 5.10 Log Output from Specifying the MEMSIZE Option

```

56      proc options option=memsize define value lognumberformat;
57      run;

      SAS (r) Proprietary Software Release V.03.02 TS1M0

Option Value Information For SAS Option MEMSIZE
Value: 2,147,483,648
Scope: SAS Session
How option value set: Config File
Config file name:
      /opt/sas/viya/SASFoundation/sasv9.cfg

Option Definition Information for SAS Option MEMSIZE
Group= MEMORY
Group Description: Memory settings
Group= PERFORMANCE
Group Description: Performance settings
Description: Specifies the limit on the amount of virtual memory that can be used during a SAS
session.
Type: The option value is of type INTMAX
Range of Values: The minimum is 0 and the maximum is 9223372036854775807
Valid Syntax(any casing): MIN|MAX|n|nK|nM|nG|nT|hexadecimal
Numeric Format: Usage of LOGNUMBERFORMAT impacts the value format
When Can Set: Session startup (command line or config) only
Restricted: Your Site Administrator can restrict modification of this option

```

Example 3: Displaying Expanded Path Environment Variables

Features: PROC OPTIONS statement options
OPTION=
EXPAND
NOEXPAND
HOST

Details

This example shows the value of an environment variable within an option value when the path is displayed.

Program

```
proc options option=msg expand;
run;
proc options option=msg noexpand;
run;
```

Program Description

Show the value of the environment variables within an option value: The EXPAND option causes the values of environment variables within the option value to display in place of the environment variable. The NOEXPAND option causes the environment variable within the options value to display. In this example, the environment variable is !sasroot

```
proc options option=msg expand;
run;
proc options option=msg noexpand;
run;
```

Log**Log 5.11** *Displaying an Expanded and Nonexpanded Pathname Using the OPTIONS Procedure*

```

56      proc options option=msg expand;
57      run;

      SAS (r) Proprietary Software Release V.03.02 TS1M0

      MSG=/opt/sas/viya/SASFoundation/sasmsg
              Specifies the path to the library that contains SAS messages.
NOTE: PROCEDURE OPTIONS used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds

58      proc options option=msg noexpand;
59      run;

      SAS (r) Proprietary Software Release V.03.02 TS1M0

      MSG=!SASROOT/sasmsg
              Specifies the path to the library that contains SAS messages.

```

Example 4: List the Options That Can Be Specified by the INSERT and APPEND Options

Features: PROC OPTIONS statement option
LISTINSERTAPPEND

Details

This example shows how to display the options that can be specified by the INSERT and APPEND system options.

Program

```

proc options listinsertappend;
run;

```

Program Description

List all options that can be specified by the INSERT and APPEND options. The LISTINSERTAPPEND option provides a list and a description of these options. Your listing might differ.

```

proc options listinsertappend;
run;

```

Log

Log 5.12 Displaying the Options That Can Be Specified by the INSERT and APPEND Options

```
56      proc options listinsertappend; run;

SAS (r) Proprietary Software Release V.03.02  TS1M0

Core options that can utilize INSERT and APPEND

AUTOEXEC           Specifies the location of the SAS AUTOEXEC files.
FMTSEARCH           Specifies the order in which format catalogs are searched.
SASAUTOS            Specifies the location of one or more autocall libraries.
SASHELP             Specifies the location of the Sashelp library.
SASSCRIPT           Specifies one or more locations of SAS/CONNECT server sign-on script files.

Host options that can utilize INSERT and APPEND

MSG                 Specifies the path to the library that contains SAS messages.
SET                 Defines an environment variable.
```


Part 5

Appendix

<i>Appendix 1</i>	
Time Zone IDs and Time Zone Names	267

Appendix 1

Time Zone IDs and Time Zone Names

Area: Africa	267
Area: America–North, Central, and South	270
Area: Antarctica	279
Area: Asia	279
Area: Atlantic	283
Area: Australia	284
Area: Miscellaneous	286
Area: Europe	291
Area: Pacific	296

Area: Africa

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Africa/Abidjan	GMT	CI	Greenwich Mean Time	00:00
Africa/Accra	GMT	GH	Greenwich Mean Time	00:00
Africa/Bamako	GMT	ML	Greenwich Mean Time	00:00
Africa/Banjul	GMT	GM	Greenwich Mean Time	00:00
Africa/Bissau	GMT	GW	Greenwich Mean Time	00:00
Africa/Conakry	GMT	GN	Greenwich Mean Time	00:00
Africa/Dakar	GMT	SN	Greenwich Mean Time	00:00
Africa/Freetown	GMT	SL	Greenwich Mean Time	00:00
Africa/Lome	GMT	TG	Greenwich Mean Time	00:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Africa/Monrovia	GMT	LR	Greenwich Mean Time	00:00
Africa/Nouakchott	GMT	MR	Greenwich Mean Time	00:00
Africa/Ouagadougou	GMT	BF	Greenwich Mean Time	00:00
Africa/Sao_Tome	GMT	ST	Greenwich Mean Time	00:00
Africa/Timbuktu	GMT	ML	Greenwich Mean Time	0.00
Africa/Casablanca	WET WEST	MA MA	Western European Time Western European Summer Time	00:00 01.00
Africa/El_Aaiun	WET WEST	EH EH	Western European Time Western European Summer Time	00:00 01.00
Africa/Algiers	CET	DZ	Central European Time	01:00
Africa/Ceuta	CET CEST	ES	Central European Time Central European Summer Time	01:00 02:00
Africa/Tunis	CET	TN	Central European Time	01:00
Africa/Bangui	WAT	CF	West Africa Time	01:00
Africa/Brazzaville	WAT	CG	West Africa Time	01:00
Africa/Douala	WAT	CM	West Africa Time	01:00
Africa/Kinshasa	WAT	CD	West Africa Time	01:00
Africa/Lagos	WAT	NG	West Africa Time	01:00
Africa/Libreville	WAT	GA	West Africa Time	01:00
Africa/Luanda	WAT	AO	West Africa Time	01:00
Africa/Malabo	WAT	GQ	West Africa Time	01:00
Africa/Ndjamena	WAT	TD	West Africa Time	01:00
Africa/Niamey	WAT	NE	West Africa Time	01:00
Africa/Porto-Novo	WAT	BJ	West Africa Time	01:00
Africa/Windhoek	WAT WAST	NA	West Africa Time West Africa Summer Time	01:00 02:00
Africa/Blantyre	CAT	MW	Central Africa Time	02:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Africa/Bujumbura	CAT	BI	Central Africa Time	02:00
Africa/Gaborone	CAT	BW	Central Africa Time	02:00
Africa/Harare	CAT	ZW	Central Africa Time	02:00
Africa/Kigali	CAT	RW	Central Africa Time	02:00
Africa/Lubumbashi	CAT	CD	Central Africa Time	02:00
Africa/Lusaka	CAT	ZM	Central Africa Time	02:00
Africa/Maputo	CAT	MZ	Central Africa Time	02:00
Africa/Cairo	EET EET	EG EG	Eastern European Time Eastern European Summer Tim	02:00 2:00
Africa/Tripoli	EET	LY	Eastern European Time	02:00
Africa/Johannesburg	SAST	ZA	South Africa Standard Time	02:00
Africa/Maseru	SAST	LS	South Africa Standard Time	02:00
Africa/Mbabane	SAST	SZ	South Africa Standard Time	02:00
Africa/Addis_Ababa	EAT	ET	East Africa Time	03:00
Africa/Asmara	EAT	ER	East Africa Time	03:00
Africa/Asmera	EAT	ER	East Africa Time	03:00
Africa/Dar_es_Salaam	EAT	TZ	East Africa Time	03:00
Africa/Djibouti	EAT	DJ	East Africa Time	03:00
Africa/Juba	EAT	SS	East Africa Time	03:00
Africa/Kampala	EAT	UG	East Africa Time	03:00
Africa/Khartoum	EAT	SD	East Africa Time	03:00
Africa/Mogadishu	EAT	SO	East Africa Time	03:00
Africa/Nairobi	EAT	KE	East Africa Time	03:00

Area: America–North, Central, and South

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
America/Adak	HAST	US	Hawaii-Aleutian Standard Time	-10:00
America/Adak	HADT	US	Hawaii-Aleutian Daylight Time	-9:00
America/Anchorage	AKST	US	Alaska Standard Time	-9:00
America/Anchorage	AKDT	US	Alaska Daylight Time	-8:00
America/Anguilla	AST	AI	Atlantic Standard Time	-4:00
America/Antigua	AST	AG	Atlantic Standard Time	-4:00
America/Araguaina	BRT	BR	Brasilia Standard Time	-3:00
America/Argentina/Buenos_Aires	ART	AR	Argentina Standard Time	-3:00
America/Argentina/Catamarca	ART	AR	Argentina Standard Time	-3:00
America/Argentina/ComodRivadavia	ART	AR	Argentina Standard Time	-3:00
America/Argentina/Cordoba	ART	AR	Argentina Standard Time	-3:00
America/Argentina/Jujuy	ART	AR	Argentina Standard Time	-3:00
America/Argentina/La_Rioja	ART	AR	Argentina Standard Time	-3:00
America/Argentina/Mendoza	ART	AR	Argentina Standard Time	-3:00
America/Argentina/Rio_Gallegos	ART	AR	Argentina Standard Time	-3:00
America/Argentina/Salta	ART	AR	Argentina Standard Time	-3:00
America/Argentina/San_Juan	ART	AR	Argentina Standard Time	-3:00
America/Argentina/San_Luis	ART	AR	Western Argentina Standard Time	-3:00
America/Argentina/Tucuman	ART	AR	Argentina Standard Time	-3:00
America/Argentina/Ushuaia	ART	AR	Argentina Standard Time	-3:00
America/Aruba	AST	AW	Atlantic Standard Time	-4:00
America/Asuncion	PYT	PY	Paraguay Standard Time	-4:00
America/Asuncion	PYST	PY	Paraguay Summer Time	-3:00
America/Atikokan	EST	CA	Eastern Standard Time	-5:00
America/Atka	HAST	US	Hawaii-Aleutian Standard Time	-10:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
America/Atka	HADT	US	Hawaii-Aleutian Daylight Time	-9:00
America/Bahia	BRT	BR	Brasilia Standard Time	-3:00
America/Bahia_Banderas	CST	MX	Central Standard Time	-6:00
America/Bahia_Banderas	CDT	MX	Central Daylight Time	-5:00
America/Barbados	AST	BB	Atlantic Standard Time	-4:00
America/Belem	BRT	BR	Brasilia Standard Time	-3:00
America/Belize	CST	BZ	Central Standard Time	-6:00
America/Blanc-Sablon	AST	CA	Atlantic Standard Time	-4:00
America/Boa_Vista	AMT	BR	Amazon Standard Time	-4:00
America/Bogota	COT	CO	Colombia Standard Time	-5:00
America/Boise	MST	US	Mountain Standard Time	-7:00
America/Boise	MDT	US	Mountain Daylight Time	-6:00
America/Buenos_Aires	ART	AR	Argentina Standard Time	-3:00
America/Cambridge_Bay	MST	CA	Mountain Standard Time	-7:00
America/Cambridge_Bay	MDT	CA	Mountain Daylight Time	-6:00
America/Campo_Grande	AMT	BR	Amazon Standard Time	-4:00
America/Campo_Grande	AMST	BR	Amazon Summer Time	-3:00
America/Cancun	CST	MX	Central Standard Time	-6:00
America/Cancun	CDT	MX	Central Daylight Time	-5:00
America/Caracas	VET	VE	Venezuela Time	-1:27
America/Catamarca	ART	AR	Argentina Standard Time	-3:00
America/Cayenne	GFT	GF	French Guiana Time	-3:00
America/Cayman	EST	KY	Eastern Standard Time	-5:00
America/Chicago	CST	US	Central Standard Time	-6:00
America/Chicago	CDT	US	Central Daylight Time	-5:00
America/Chihuahua	MST	MX	Mountain Standard Time	-7:00
America/Chihuahua	MDT	MX	Mountain Daylight Time	-6:00
America/Coral_Harbour	EST	CA	Eastern Standard Time	-5:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
America/Cordoba	ART	AR	Argentina Standard Time	-3:00
America/Costa_Rica	CST	CR	Central Standard Time	-6:00
America/Creston	MST	CA	Mountain Standard Time	-7:00
America/Cuiaba	AMT	BR	Amazon Standard Time	-4:00
America/Cuiaba	AMST	BR	Amazon Summer Time	-3:00
America/Curacao	AST	CW	Atlantic Standard Time	-4:00
America/Danmarkshavn	GMT	GL	Greenwich Mean Time	0:00
America/Dawson	PST	CA	Pacific Standard Time	-8:00
America/Dawson	PDT	CA	Pacific Daylight Time	-7:00
America/Dawson_Creek	MST	CA	Mountain Standard Time	-7:00
America/Denver	MST	US	Mountain Standard Time	-7:00
America/Denver	MDT	US	Mountain Daylight Time	-6:00
America/Detroit	EST	US	Eastern Standard Time	-5:00
America/Detroit	EDT	US	Eastern Daylight Time	-4:00
America/Dominica	AST	DM	Atlantic Standard Time	-4:00
America/Edmonton	MST	CA	Mountain Standard Time	-7:00
America/Edmonton	MDT	CA	Mountain Daylight Time	-6:00
America/Eirunepe	ACT	BR	Amazon Standard Time	-5:00
America/El_Salvador	CST	SV	Central Standard Time	-6:00
America/Ensenada	PST	MX	Pacific Standard Time	-8:00
America/Ensenada	PDT	MX	Pacific Daylight Time	-7:00
America/Fort_Wayne	EST	US	Eastern Standard Time	-5:00
America/Fort_Wayne	EDT	US	Eastern Daylight Time	-4:00
America/Fortaleza	BRT	BR	Brasilia Standard Time	-3:00
America/Glace_Bay	AST	CA	Atlantic Standard Time	-4:00
America/Glace_Bay	ADT	CA	Atlantic Daylight Time	-3:00
America/Godthab	WGT	GL	West Greenland Standard Time	-3:00
America/Godthab	WGST	GL	West Greenland Summer Time	-2:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
America/Goose_Bay	AST	CA	Atlantic Standard Time	-4:00
America/Goose_Bay	ADT	CA	Atlantic Daylight Time	-3:00
America/Grand_Turk	EST	TC	Eastern Standard Time	-5:00
America/Grand_Turk	EDT	TC	Eastern Daylight Time	-4:00
America/Grenada	AST	GD	Atlantic Standard Time	-4:00
America/Guadeloupe	AST	GP	Atlantic Standard Time	-4:00
America/Guatemala	CST	GT	Central Standard Time	-6:00
America/Guayaquil	ECT	EC	Ecuador Time	-5:00
America/Guyana	GYT	GY	Guyana Time	-4:00
America/Halifax	AST	CA	Atlantic Standard Time	-4:00
America/Halifax	ADT	CA	Atlantic Daylight Time	-3:00
America/Havana	CST	CU	Cuba Standard Time	-5:00
America/Havana	CDT	CU	Cuba Daylight Time	-4:00
America/Hermosillo	MST	MX	Mountain Standard Time	-7:00
America/Indiana/Indianapolis	EST	US	Eastern Standard Time	-5:00
America/Indiana/Indianapolis	EDT	US	Eastern Daylight Time	-4:00
America/Indiana/Knox	CST	US	Central Standard Time	-6:00
America/Indiana/Knox	CDT	US	Central Daylight Time	-5:00
America/Indiana/Marengo	EST	US	Eastern Standard Time	-5:00
America/Indiana/Marengo	EDT	US	Eastern Daylight Time	-4:00
America/Indiana/Petersburg	EST	US	Eastern Standard Time	-5:00
America/Indiana/Petersburg	EDT	US	Eastern Daylight Time	-4:00
America/Indiana/Tell_City	CST	US	Central Standard Time	-6:00
America/Indiana/Tell_City	CDT	US	Central Daylight Time	-5:00
America/Indiana/Vevay	EST	US	Eastern Standard Time	-5:00
America/Indiana/Vevay	EDT	US	Eastern Daylight Time	-4:00
America/Indiana/Vincennes	EST	US	Eastern Standard Time	-5:00
America/Indiana/Vincennes	EDT	US	Eastern Daylight Time	-4:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
America/Indiana/Winamac	EST	US	Eastern Standard Time	-5:00
America/Indiana/Winamac	EDT	US	Eastern Daylight Time	-4:00
America/Indianapolis	EST	US	Eastern Standard Time	-5:00
America/Indianapolis	EDT	US	Eastern Daylight Time	-4:00
America/Inuvik	MST	CA	Mountain Standard Time	-7:00
America/Inuvik	MDT	CA	Mountain Daylight Time	-6:00
America/Iqaluit	EST	CA	Eastern Standard Time	-5:00
America/Iqaluit	EDT	CA	Eastern Daylight Time	-4:00
America/Jamaica	EST	JM	Eastern Standard Time	-5:00
America/Jujuy	ART	AR	Argentina Standard Time	-3:00
America/Juneau	AKST	US	Alaska Standard Time	-9:00
America/Juneau	AKDT	US	Alaska Daylight Time	-8:00
America/Kentucky/Louisville	EST	US	Eastern Standard Time	-5:00
America/Kentucky/Louisville	EDT	US	Eastern Daylight Time	-4:00
America/Kentucky/Monticello	EST	US	Eastern Standard Time	-5:00
America/Kentucky/Monticello	EDT	US	Eastern Daylight Time	-4:00
America/Knox_IN	CST	US	Central Standard Time	-6:00
America/Knox_IN	CDT	US	Central Daylight Time	-5:00
America/Kralendijk	AST	BQ	Atlantic Standard Time	-4:00
America/La_Paz	BOT	BO	Bolivia Time	-4:00
America/Lima	PET	PE	Peru Standard Time	-5:00
America/Los_Angeles	PST	US	Pacific Standard Time	-8:00
America/Los_Angeles	PDT	US	Pacific Daylight Time	-7:00
America/Louisville	EST	US	Eastern Standard Time	-5:00
America/Louisville	EDT	US	Eastern Daylight Time	-4:00
America/Lower_Princes	AST	SX	Atlantic Standard Time	-4:00
America/Maceio	BRT	BR	Brasilia Standard Time	-3:00
America/Managua	CST	NI	Central Standard Time	-6:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
America/Manaus	AMT	BR	Amazon Standard Time	-4:00
America/Marigot	AST	MF	Atlantic Standard Time	-4:00
America/Martinique	AST	MQ	Atlantic Standard Time	-4:00
America/Matamoros	CST	MX	Central Standard Time	-6:00
America/Matamoros	CDT	MX	Central Daylight Time	-5:00
America/Mazatlan	MST	MX	Mountain Standard Time	-7:00
America/Mazatlan	MDT	MX	Mountain Daylight Time	-6:00
America/Mendoza	ART	AR	Argentina Standard Time	-3:00
America/Menominee	CST	US	Central Standard Time	-6:00
America/Menominee	CDT	US	Central Daylight Time	-5:00
America/Merida	CST	MX	Central Standard Time	-6:00
America/Merida	CDT	MX	Central Daylight Time	-5:00
America/Metlakatla	MEST	US	Pacific Standard Time	-8:00
America/Mexico_City	CST	MX	Central Standard Time	-6:00
America/Mexico_City	CDT	MX	Central Daylight Time	-5:00
America/Miquelon	PMST	PM	Saint Pierre and Miquelon Standard Time	-3:00
America/Miquelon	PMDT	PM	Saint Pierre and Miquelon Daylight Time	-2:00
America/Moncton	AST	CA	Atlantic Standard Time	-4:00
America/Moncton	ADT	CA	Atlantic Daylight Time	-3:00
America/Monterrey	CST	MX	Central Standard Time	-6:00
America/Monterrey	CDT	MX	Central Daylight Time	-5:00
America/Montevideo	UYT	UY	Uruguay Standard Time	-3:00
America/Montevideo	UYST	UY	Uruguay Summer Time	-2:00
America/Montserrat	AST	MS	Atlantic Standard Time	-4:00
America/Nassau	EST	BS	Eastern Standard Time	-5:00
America/Nassau	EDT	BS	Eastern Daylight Time	-4:00
America/New_York	EST	US	Eastern Standard Time	-5:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
America/New_York	EDT	US	Eastern Daylight Time	-4:00
America/Nipigon	EST	CA	Eastern Standard Time	-5:00
America/Nipigon	EDT	CA	Eastern Daylight Time	-4:00
America/Nome	AKST	US	Alaska Standard Time	-9:00
America/Nome	AKDT	US	Alaska Daylight Time	-8:00
America/Noronha	FNT	BR	Fernando de Noronha Standard Time	-2:00
America/North_Dakota/Beulah	CST	US	Central Standard Time	-6:00
America/North_Dakota/Beulah	CDT	US	Central Daylight Time	-5:00
America/North_Dakota/Center	CST	US	Central Standard Time	-6:00
America/North_Dakota/Center	CDT	US	Central Daylight Time	-5:00
America/North_Dakota/ New_Salem	CST	US	Central Standard Time	-6:00
America/North_Dakota/ New_Salem	CDT	US	Central Daylight Time	-5:00
America/Ojinaga	MST	MX	Mountain Standard Time	-7:00
America/Ojinaga	MDT	MX	Mountain Daylight Time	-6:00
America/Panama	EST	PA	Eastern Standard Time	-5:00
America/Pangnirtung	EST	CA	Eastern Standard Time	-5:00
America/Pangnirtung	EDT	CA	Eastern Daylight Time	-4:00
America/Paramaribo	SRT	SR	Suriname Time	-3:00
America/Phoenix	MST	US	Mountain Standard Time	-7:00
America/Port-au-Prince	EST	HT	Eastern Standard Time	-5:00
America/Port-au-Prince	EDT	HT	Eastern Daylight Time	-4:00
America/Port_of_Spain	AST	TT	Atlantic Standard Time	-4:00
America/Porto_Acre	ACT	BR	Amazon Standard Time	-5:00
America/Porto_Velho	AMT	BR	Amazon Standard Time	-4:00
America/Puerto_Rico	AST	PR	Atlantic Standard Time	-4:00
America/Rainy_River	CST	CA	Central Standard Time	-6:00
America/Rainy_River	CDT	CA	Central Daylight Time	-5:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
America/Rankin_Inlet	CST	CA	Central Standard Time	-6:00
America/Rankin_Inlet	CDT	CA	Central Daylight Time	-5:00
America/Recife	BRT	BR	Brasilia Standard Time	-3:00
America/Regina	CST	CA	Central Standard Time	-6:00
America/Resolute	CST	CA	Central Standard Time	-6:00
America/Resolute	CDT	CA	Central Daylight Time	-5:00
America/Rio_Branco	ACT	BR	Amazon Standard Time	-5:00
America/Rosario	ART	AR	Argentina Standard Time	-3:00
America/Santa_Isabel	PST	MX	Pacific Standard Time	-8:00
America/Santa_Isabel	PDT	MX	Pacific Daylight Time	-7:00
America/Santarem	BRT	BR	Brasilia Standard Time	-3:00
America/Santiago	CLT	CL	Chile Standard Time	-4:00
America/Santiago	CLST	CL	Chile Summer Time	-3:00
America/Santo_Domingo	AST	DO	Atlantic Standard Time	-4:00
America/Sao_Paulo	BRT	BR	Brasilia Standard Time	-3:00
America/Sao_Paulo	BRST	BR	Brasilia Summer Time	-2:00
America/Scoresbysund	EGT	GL	East Greenland Standard Time	-1:00
America/Scoresbysund	EGST	GL	East Greenland Summer Time	0:00
America/Shiprock	MST	US	Mountain Standard Time	-7:00
America/Shiprock	MDT	US	Mountain Daylight Time	-6:00
America/Sitka	AKST	US	Alaska Standard Time	-9:00
America/Sitka	AKDT	US	Alaska Daylight Time	-8:00
America/St_Barthelemy	AST	BL	Atlantic Standard Time	-4:00
America/St_Johns	NST	CA	Newfoundland Standard Time	-1:27
America/St_Johns	NDT	CA	Newfoundland Daylight Time	-1:27
America/St_Kitts	AST	KN	Atlantic Standard Time	-4:00
America/St_Lucia	AST	LC	Atlantic Standard Time	-4:00
America/St_Thomas	AST	VI	Atlantic Standard Time	-4:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
America/St_Vincent	AST	VC	Atlantic Standard Time	-4:00
America/Swift_Current	CST	CA	Central Standard Time	-6:00
America/Tegucigalpa	CST	HN	Central Standard Time	-6:00
America/Thule	AST	GL	Atlantic Standard Time	-4:00
America/Thule	ADT	GL	Atlantic Daylight Time	-3:00
America/Thunder_Bay	EST	CA	Eastern Standard Time	-5:00
America/Thunder_Bay	EDT	CA	Eastern Daylight Time	-4:00
America/Tijuana	PST	MX	Pacific Standard Time	-8:00
America/Tijuana	PDT	MX	Pacific Daylight Time	-7:00
America/Toronto	EST	CA	Eastern Standard Time	-5:00
America/Toronto	EDT	CA	Eastern Daylight Time	-4:00
America/Tortola	AST	VG	Atlantic Standard Time	-4:00
America/Vancouver	PST	CA	Pacific Standard Time	-8:00
America/Vancouver	PDT	CA	Pacific Daylight Time	-7:00
America/Virgin	AST	TT	Atlantic Standard Time	-4:00
America/Whitehorse	PST	CA	Pacific Standard Time	-8:00
America/Whitehorse	PDT	CA	Pacific Daylight Time	-7:00
America/Winnipeg	CST	CA	Central Standard Time	-6:00
America/Winnipeg	CDT	CA	Central Daylight Time	-5:00
America/Yakutat	AKST	US	Alaska Standard Time	-9:00
America/Yakutat	AKDT	US	Alaska Daylight Time	-8:00
America/Yellowknife	MST	CA	Mountain Standard Time	-7:00
America/Yellowknife	MDT	CA	Mountain Daylight Time	-6:00

Area: Antarctica

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Antarctica/Casey	WST	AQ	Australian Western Standard Time	8:00
Antarctica/Davis	DAVT	AQ	Davis Time	7:00
Antarctica/DumontDUrville	DDUT	AQ	Dumont-d\u2019Urville Time	10:00
Antarctica/Macquarie	MIST	AU	Macquarie Island Time	11:00
Antarctica/Mawson	MAWT	AQ	Mawson Time	5:00
Antarctica/McMurdo	NZST	AQ	New Zealand Standard Time	12:00
Antarctica/McMurdo	NZDT	AQ	New Zealand Daylight Time	13:00
Antarctica/Palmer	CLT	AQ	Chile Standard Time	-4:00
Antarctica/Palmer	CLST	AQ	Chile Summer Time	-3:00
Antarctica/Rothera	ROTT	AQ	Rothera Time	-3:00
Antarctica/South_Pole	NZST	NZ	New Zealand Standard Time	12:00
Antarctica/South_Pole	NZDT	NZ	New Zealand Daylight Time	13:00
Antarctica/Syowa	SYOT	AQ	Syowa Time	3:00
Antarctica/Vostok	VOST	AQ	Vostok Time	6:00

Area: Asia

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Asia/Aden	AST	YE	Arabian Standard Time	3:00
Asia/Almaty	ALMT	KZ	East Kazakhstan Time	6:00
Asia/Amman	EET	JO	Eastern European Standard Time	2:00
Asia/Amman	EEST	JO	Eastern European Summer Time	3:00
Asia/Anadyr	ANAT	RU	Magadan Standard Time	12:00
Asia/Aqtau	AQTT	KZ	West Kazakhstan Time	5:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Asia/Aqtobe	AQTT	KZ	West Kazakhstan Time	5:00
Asia/Ashgabat	TMT	TM	Turkmenistan Standard Time	5:00
Asia/Ashkhabad	TMT	TM	Turkmenistan Standard Time	5:00
Asia/Baghdad	AST	IQ	Arabian Standard Time	3:00
Asia/Bahrain	AST	BH	Arabian Standard Time	3:00
Asia/Baku	AZT	AZ	Azerbaijan Standard Time	4:00
Asia/Baku	AZST	AZ	Azerbaijan Summer Time	5:00
Asia/Bangkok	ICT	TH	Indochina Time	7:00
Asia/Beijing	CST	CN	China Standard Time	8:00
Asia/Beirut	EET	LB	Eastern European Standard Time	2:00
Asia/Beirut	EEST	LB	Eastern European Summer Time	3:00
Asia/Bishkek	KGT	KG	Kyrgyzstan Time	6:00
Asia/Brunei	BNT	BN	Brunei Darussalam Time	8:00
Asia/Calcutta	IST	IN	India Standard Time	5:30
Asia/Choibalsan	CHOT	MN	Choibalsan Standard Time	8:00
Asia/Chongqing	CST	CN	China Standard Time	8:00
Asia/Chungking	CST	CN	China Standard Time	8:00
Asia/Colombo	IST	LK	India Standard Time	5:30
Asia/Dacca	BDT	BD	Bangladesh Standard Time	6:00
Asia/Damascus	EET	SY	Eastern European Standard Time	2:00
Asia/Damascus	EEST	SY	Eastern European Summer Time	3:00
Asia/Dhaka	BDT	BD	Bangladesh Standard Time	6:00
Asia/Dili	TLT	TL	East Timor Time	9:00
Asia/Dubai	GST	AE	Gulf Standard Time	4:00
Asia/Dushanbe	TJT	TJ	Tajikistan Time	5:00
Asia/Gaza	EET	PS	Eastern European Standard Time	2:00
Asia/Gaza	EEST	PS	Eastern European Summer Time	3:00
Asia/Harbin	CST	CN	China Standard Time	8:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Asia/Hebron	EET	PS	Eastern European Standard Time	2:00
Asia/Hebron	EEST	PS	Eastern European Summer Time	3:00
Asia/Ho_Chi_Minh	ICT	VN	Indochina Time	7:00
Asia/Hong_Kong	HKT	HK	Hong Kong Standard Time	8:00
Asia/Hovd	HOVT	MN	Hovd Standard Time	7:00
Asia/Irkutsk	IRKT	RU	Irkutsk Standard Time	9:00
Asia/Istanbul	EET	TR	Eastern European Standard Time	2:00
Asia/Istanbul	EEST	TR	Eastern European Summer Time	3:00
Asia/Jakarta	WIB	ID	Western Indonesia Time	7:00
Asia/Jayapura	WIT	ID	Eastern Indonesia Time	9:00
Asia/Jerusalem	IST	IL	Israel Standard Time	2:00
Asia/Jerusalem	IDT	IL	Israel Daylight Time	3:00
Asia/Kabul	AFT	AF	Afghanistan Time	4:30
Asia/Kamchatka	PETT	RU	Magadan Standard Time	12:00
Asia/Karachi	PKT	PK	Pakistan Standard Time	5:00
Asia/Kashgar	CST	CN	China Standard Time	8:00
Asia/Kathmandu	NPT	NP	Nepal Time	5:45
Asia/Katmandu	NPT	NP	Nepal Time	5:45
Asia/Khandyga	YAKT	RU	Yakutsk Standard Time	10:00
Asia/Kolkata	IST	IN	India Standard Time	5:30
Asia/Krasnoyarsk	KRAT	RU	Krasnoyarsk Standard Time	8:00
Asia/Kuala_Lumpur	MYT	MY	Malaysia Time	8:00
Asia/Kuching	MYT	MY	Malaysia Time	8:00
Asia/Kuwait	AST	KW	Arabian Standard Time	3:00
Asia/Macao	CST	MO	China Standard Time	8:00
Asia/Macau	CST	MO	China Standard Time	8:00
Asia/Magadan	MAGT	RU	Magadan Standard Time	12:00
Asia/Makassar	WITA	ID	Central Indonesia Time	8:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Asia/Manila	PHT	PH	Philippine Standard Time	8:00
Asia/Muscat	GST	OM	Gulf Standard Time	4:00
Asia/Nicosia	EET	CY	Eastern European Standard Time	2:00
Asia/Nicosia	EEST	CY	Eastern European Summer Time	3:00
Asia/Novokuznetsk	NOVT	RU	Novosibirsk Standard Time	7:00
Asia/Novosibirsk	NOVT	RU	Novosibirsk Standard Time	7:00
Asia/Omsk	OMST	RU	Omsk Standard Time	7:00
Asia/Oral	ORAT	KZ	West Kazakhstan Time	5:00
Asia/Osaka	JST	JP	Japan Standard Time	9:00
Asia/Phnom_Penh	ICT	KH	Indochina Time	7:00
Asia/Pontianak	WIB	ID	Western Indonesia Time	7:00
Asia/Pyongyang	KST	KP	Korean Standard Time	9:00
Asia/Qatar	AST	QA	Arabian Standard Time	3:00
Asia/Qyzylorda	QYZT	KZ	East Kazakhstan Time	6:00
Asia/Rangoon	MMT	MM	Myanmar Time	6:30
Asia/Riyadh	AST	SA	Arabian Standard Time	3:00
Asia/Saigon	ICT	VN	Indochina Time	7:00
Asia/Sakhalin	SAKT	RU	Sakhalin Standard Time	11:00
Asia/Samarkand	UZT	UZ	Uzbekistan Standard Time	5:00
Asia/Sapporo	JST	JP	Japan Standard Time	9:00
Asia/Seoul	KST	KR	Korean Standard Time	9:00
Asia/Shanghai	CST	CN	China Standard Time	8:00
Asia/Singapore	SGT	SG	Singapore Standard Time	8:00
Asia/Taipei	CST	TW	Taipei Standard Time	8:00
Asia/Tashkent	UZT	UZ	Uzbekistan Standard Time	5:00
Asia/Tbilisi	GET	GE	Georgia Standard Time	4:00
Asia/Tehran	IRST	IR	Iran Standard Time	3:30
Asia/Tel_Aviv	IST	IL	Israel Standard Time	2:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Asia/Tel_Aviv	IDT	IL	Israel Daylight Time	3:00
Asia/Thimbu	BTT	BT	Bhutan Time	6:00
Asia/Thimphu	BTT	BT	Bhutan Time	6:00
Asia/Tokyo	JST	JP	Japan Standard Time	9:00
Asia/Ujung_Pandang	WITA	ID	Central Indonesia Time	8:00
Asia/Ulaanbaatar	ULAT	MN	Ulan Bator Standard Time	8:00
Asia/Ulan_Bator	ULAT	MN	Ulan Bator Standard Time	8:00
Asia/Urumqi	CST	CN	China Standard Time	8:00
Asia/Ust-Nera	VLAT	RU	Vladivostok Standard Time	11:00
Asia/Vientiane	ICT	LA	Indochina Time	7:00
Asia/Vladivostok	VLAT	RU	Vladivostok Standard Time	11:00
Asia/Yakutsk	YAKT	RU	Yakutsk Standard Time	10:00
Asia/Yekaterinburg	YEKT	RU	Yekaterinburg Standard Time	6:00
Asia/Yerevan	AMT	AM	Armenia Standard Time	4:00

Area: Atlantic

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Atlantic/Azores	AZOT	PT	Azores Standard Time	-1:00
Atlantic/Azores	AZOST	PT	Azores Summer Time	0:00
Atlantic/Bermuda	AST	BM	Atlantic Standard Time	-4:00
Atlantic/Bermuda	ADT	BM	Atlantic Daylight Time	-3:00
Atlantic/Canary	WET	ES	Western European Standard Time	0:00
Atlantic/Canary	WEST	ES	Western European Summer Time	1:00
Atlantic/Cape_Verde	CVT	CV	Cape Verde Standard Time	-1:00
Atlantic/Faeroe	WET	FO	Western European Standard Time	0:00
Atlantic/Faeroe	WEST	FO	Western European Summer Time	1:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Atlantic/Faroe	WET	FO	Western European Standard Time	0:00
Atlantic/Faroe	WEST	FO	Western European Summer Time	1:00
Atlantic/Jan_Mayen	CET	NO	Central European Standard Time	1:00
Atlantic/Jan_Mayen	CEST	NO	Central European Summer Time	2:00
Atlantic/Madeira	WET	PT	Western European Standard Time	0:00
Atlantic/Madeira	WEST	PT	Western European Summer Time	1:00
Atlantic/Reykjavik	GMT	IS	Greenwich Mean Time	0:00
Atlantic/South_Georgia	GST	GS	South Georgia Time	-2:00
Atlantic/St_Helena	GMT	SH	Greenwich Mean Time	0:00
Atlantic/Stanley	FKST	FK	Falkland Islands Standard Time	-3:00

Area: Australia

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Australia/ACT	EST	AU	Australian Eastern Standard Time	10:00
Australia/ACT	EDT	AU	Australian Eastern Daylight Time	11:00
Australia/Adelaide	CST	AU	Australian Central Standard Time	9:30
Australia/Adelaide	CDT	AU	Australian Central Daylight Time	10:30
Australia/Brisbane	EST	AU	Australian Eastern Standard Time	10:00
Australia/Broken_Hill	CST	AU	Australian Central Standard Time	9:30
Australia/Broken_Hill	CDT	AU	Australian Central Daylight Time	10:30
Australia/Canberra	EST	AU	Australian Eastern Standard Time	10:00
Australia/Canberra	EDT	AU	Australian Eastern Daylight Time	11:00
Australia/Currie	EST	AU	Australian Eastern Standard Time	10:00
Australia/Currie	EDT	AU	Australian Eastern Daylight Time	11:00
Australia/Darwin	CST	AU	Australian Central Standard Time	9:30

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Australia/Eucla	CWST	AU	Australian Central Western Standard Time	8:45
Australia/Hobart	EST	AU	Australian Eastern Standard Time	10:00
Australia/Hobart	EDT	AU	Australian Eastern Daylight Time	11:00
Australia/LHI	LHST	AU	Lord Howe Standard Time	10:30
Australia/LHI	LHDT	AU	Lord Howe Daylight Time	11:00
Australia/Lindeman	EST	AU	Australian Eastern Standard Time	10:00
Australia/Lord_Howe	LHST	AU	Lord Howe Standard Time	10:30
Australia/Lord_Howe	LHDT	AU	Lord Howe Daylight Time	11:00
Australia/Melbourne	EST	AU	Australian Eastern Standard Time	10:00
Australia/Melbourne	EDT	AU	Australian Eastern Daylight Time	11:00
Australia/NSW	EST	AU	Australian Eastern Standard Time	10:00
Australia/NSW	EDT	AU	Australian Eastern Daylight Time	11:00
Australia/North	CST	AU	Australian Central Standard Time	9:30
Australia/Perth	WST	AU	Australian Western Standard Time	8:00
Australia/Queensland	EST	AU	Australian Eastern Standard Time	10:00
Australia/South	CST	AU	Australian Central Standard Time	9:30
Australia/South	CDT	AU	Australian Central Daylight Time	10:30
Australia/Sydney	EST	AU	Australian Eastern Standard Time	10:00
Australia/Sydney	EDT	AU	Australian Eastern Daylight Time	11:00
Australia/Tasmania	EST	AU	Australian Eastern Standard Time	10:00
Australia/Tasmania	EDT	AU	Australian Eastern Daylight Time	11:00
Australia/Victoria	EST	AU	Australian Eastern Standard Time	10:00
Australia/Victoria	EDT	AU	Australian Eastern Daylight Time	11:00
Australia/West	WST	AU	Australian Western Standard Time	8:00
Australia/Yancowinna	CST	AU	Australian Central Standard Time	9:30
Australia/Yancowinna	CDT	AU	Australian Central Daylight Time	10:30

Area: Miscellaneous

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Arctic/Longyearbyen	CET	SJ	Central European Standard Time	1:00
Arctic/Longyearbyen	CEST	SJ	Central European Summer Time	2:00
Brazil/Acre	ACT	BR	Amazon Standard Time	-5:00
Brazil/DeNoronha	FNT	BR	Fernando de Noronha Standard Time	-2:00
Brazil/East	BRT	BR	Brasilia Standard Time	-3:00
Brazil/East	BRST	BR	Brasilia Summer Time	-2:00
Brazil/West	AMT	BR	Amazon Standard Time	-4:00
CST6CDT	CST		Central Standard Time	-6:00
CST6CDT	CDT		Central Daylight Time	-5:00
Canada/Atlantic	AST	CA	Atlantic Standard Time	-4:00
Canada/Atlantic	ADT	CA	Atlantic Daylight Time	-3:00
Canada/Central	CST	CA	Central Standard Time	-6:00
Canada/Central	CDT	CA	Central Daylight Time	-5:00
Canada/East-Saskatchewan	CST	CA	Central Standard Time	-6:00
Canada/Eastern	EST	CA	Eastern Standard Time	-5:00
Canada/Eastern	EDT	CA	Eastern Daylight Time	-4:00
Canada/Mountain	MST	CA	Mountain Standard Time	-7:00
Canada/Mountain	MDT	CA	Mountain Daylight Time	-6:00
Canada/Newfoundland	NST	CA	Newfoundland Standard Time	-1:27
Canada/Newfoundland	NDT	CA	Newfoundland Daylight Time	-1:27
Canada/Pacific	PST	CA	Pacific Standard Time	-8:00
Canada/Pacific	PDT	CA	Pacific Daylight Time	-7:00
Canada/Saskatchewan	CST	CA	Central Standard Time	-6:00
Canada/Yukon	PST	CA	Pacific Standard Time	-8:00
Canada/Yukon	PDT	CA	Pacific Daylight Time	-7:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Chile/Continental	CLT	CL	Chile Standard Time	-4:00
Chile/Continental	CLST	CL	Chile Summer Time	-3:00
Chile/EasterIsland	EAST	CL	Easter Island Standard Time	-6:00
Chile/EasterIsland	EASST	CL	Easter Island Summer Time	-5:00
Cuba	CST	CU	Cuba Standard Time	-5:00
Cuba	CDT	CU	Cuba Daylight Time	-4:00
EST5EDT	EST		Eastern Standard Time	-5:00
EST5EDT	EDT		Eastern Daylight Time	-4:00
Egypt	EET	EG	Eastern European Standard Time	3:00
	EET	EG	Eastern European Summer Time	0:00
Eire	GMT	IE	Greenwich Mean Time	0:00
Eire	IST	IE		1:00
Etc/GMT	GMT		Greenwich Mean Time	0:00
Etc/GMT+0	GMT		Greenwich Mean Time	0:00
Etc/GMT+1	GMT-01		1 hours behind GMT	-1:00
Etc/GMT+10	GMT-10		10 hours behind GMT	-10:00
Etc/GMT+11	GMT-11		11 hours behind GMT	-11:00
Etc/GMT+12	GMT-12		12 hours behind GMT	-12:00
Etc/GMT+2	GMT-02		2 hours behind GMT	-2:00
Etc/GMT+3	GMT-03		3 hours behind GMT	-3:00
Etc/GMT+4	GMT-04		4 hours behind GMT	-4:00
Etc/GMT+5	GMT-05		5 hours behind GMT	-5:00
Etc/GMT+6	GMT-06		6 hours behind GMT	-6:00
Etc/GMT+7	GMT-07		7 hours behind GMT	-7:00
Etc/GMT+8	GMT-08		8 hours behind GMT	-8:00
Etc/GMT+9	GMT-09		9 hours behind GMT	-9:00
Etc/GMT-0	GMT		Greenwich Mean Time	0:00
Etc/GMT-1	GMT+01		1 hours ahead of GMT	1:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Etc/GMT-10	GMT+10		10 hours ahead of GMT	10:00
Etc/GMT-11	GMT+11		11 hours ahead of GMT	11:00
Etc/GMT-12	GMT+12		12 hours ahead of GMT	12:00
Etc/GMT-13	GMT+13		13 hours ahead of GMT	13:00
Etc/GMT-14	GMT+14		14 hours ahead of GMT	14:00
Etc/GMT-2	GMT+02		2 hours ahead of GMT	2:00
Etc/GMT-3	GMT+03		3 hours ahead of GMT	3:00
Etc/GMT-4	GMT+04		4 hours ahead of GMT	4:00
Etc/GMT-5	GMT+05		5 hours ahead of GMT	5:00
Etc/GMT-6	GMT+06		6 hours ahead of GMT	6:00
Etc/GMT-7	GMT+07		7 hours ahead of GMT	7:00
Etc/GMT-8	GMT+08		8 hours ahead of GMT	8:00
Etc/GMT-9	GMT+09		9 hours ahead of GMT	9:00
Etc/GMT0	GMT		Greenwich Mean Time	0:00
Etc/Greenwich	GMT		Greenwich Mean Time	0:00
Etc/UCT	UCT		Greenwich Mean Time	0:00
Etc/UTC	UTC		Greenwich Mean Time	0:00
Etc/Universal	UTC		Greenwich Mean Time	0:00
Etc/Zulu	UTC		Greenwich Mean Time	0:00
GB	GMT	GB	Greenwich Mean Time	0:00
GB	BST	GB		1:00
GB-Eire	GMT	GB	Greenwich Mean Time	0:00
GB-Eire	BST	GB		1:00
GMT	GMT		Greenwich Mean Time	0:00
GMT+0	GMT		Greenwich Mean Time	0:00
GMT-0	GMT		Greenwich Mean Time	0:00
GMT0	GMT		Greenwich Mean Time	0:00
Greenwich	GMT		Greenwich Mean Time	0:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Hongkong	HKT	HK	Hong Kong Standard Time	8:00
Iceland	GMT	IS	Greenwich Mean Time	0:00
Indian/Antananarivo	EAT	MG	East Africa Time	3:00
Indian/Chagos	IOT	IO	Indian Ocean Time	6:00
Indian/Christmas	CXT	CX	Christmas Island Time	7:00
Indian/Cocos	CCT	CC	Cocos Islands Time	6:30
Indian/Comoro	EAT	KM	East Africa Time	3:00
Indian/Kerguelen	TFT	TF	French Southern and Antarctic Time	5:00
Indian/Mahe	SCT	SC	Seychelles Time	4:00
Indian/Maldives	MVT	MV	Maldives Time	5:00
Indian/Mauritius	MUT	MU	Mauritius Standard Time	4:00
Indian/Mayotte	EAT	YT	East Africa Time	3:00
Indian/Reunion	RET	RE	Reunion Time	4:00
Iran	IRST	IR	Iran Standard Time	3:30
Israel	IST	IL	Israel Standard Time	2:00
Israel	IDT	IL	Israel Daylight Time	3:00
Jamaica	EST	JM	Eastern Standard Time	-5:00
Japan	JST	JP	Japan Standard Time	9:00
Kwajalein	MHT	MH	Marshall Islands Time	12:00
Libya	EET	LY	Central European Standard Time	2:00
MST7MDT	MST		Mountain Standard Time	-7:00
MST7MDT	MDT		Mountain Daylight Time	-6:00
Mexico/BajaNorte	PST	MX	Pacific Standard Time	-8:00
Mexico/BajaNorte	PDT	MX	Pacific Daylight Time	-7:00
Mexico/BajaSur	MST	MX	Mountain Standard Time	-7:00
Mexico/BajaSur	MDT	MX	Mountain Daylight Time	-6:00
Mexico/General	CST	MX	Central Standard Time	-6:00
Mexico/General	CDT	MX	Central Daylight Time	-5:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
NZ	NZST	NZ	New Zealand Standard Time	12:00
NZ	NZDT	NZ	New Zealand Daylight Time	13:00
NZ-CHAT	CHAST	NZ	Chatham Standard Time	12:45
NZ-CHAT	CHADT	NZ	Chatham Daylight Time	13:45
Navajo	MST	US	Mountain Standard Time	-7:00
Navajo	MDT	US	Mountain Daylight Time	-6:00
PRC	CST	CN	China Standard Time	8:00
PST8PDT	PST		Pacific Standard Time	-8:00
PST8PDT	PDT		Pacific Daylight Time	-7:00
Poland	CET	PL	Central European Standard Time	1:00
Poland	CEST	PL	Central European Summer Time	2:00
Portugal	WET	PT	Western European Standard Time	0:00
Portugal	WEST	PT	Western European Summer Time	1:00
ROC	CST	TW	Taipei Standard Time	8:00
ROK	KST	KR	Korean Standard Time	9:00
Singapore	SGT	SG	Singapore Standard Time	8:00
Turkey	EET	TR	Eastern European Standard Time	2:00
Turkey	EEST	TR	Eastern European Summer Time	3:00
UCT	UCT		Greenwich Mean Time	0:00
US/Alaska	AKST	US	Alaska Standard Time	-9:00
US/Alaska	AKDT	US	Alaska Daylight Time	-8:00
US/Aleutian	HAST	US	Hawaii-Aleutian Standard Time	-10:00
US/Aleutian	HADT	US	Hawaii-Aleutian Daylight Time	-9:00
US/Arizona	MST	US	Mountain Standard Time	-7:00
US/Central	CST	US	Central Standard Time	-6:00
US/Central	CDT	US	Central Daylight Time	-5:00
US/East-Indiana	EST	US	Eastern Standard Time	-5:00
US/East-Indiana	EDT	US	Eastern Daylight Time	-4:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
US/Eastern	EST	US	Eastern Standard Time	-5:00
US/Eastern	EDT	US	Eastern Daylight Time	-4:00
US/Hawaii	HST	US	Hawaii-Aleutian Standard Time	-10:00
US/Indiana-Starke	CST	US	Central Standard Time	-6:00
US/Indiana-Starke	CDT	US	Central Daylight Time	-5:00
US/Michigan	EST	US	Eastern Standard Time	-5:00
US/Michigan	EDT	US	Eastern Daylight Time	-4:00
US/Mountain	MST	US	Mountain Standard Time	-7:00
US/Mountain	MDT	US	Mountain Daylight Time	-6:00
US/Pacific	PST	US	Pacific Standard Time	-8:00
US/Pacific	PDT	US	Pacific Daylight Time	-7:00
US/Pacific-New	PST	US	Pacific Standard Time	-8:00
US/Pacific-New	PDT	US	Pacific Daylight Time	-7:00
US/Samoa	SST	AS	Samoa Standard Time	-11:00
UTC	UTC		Greenwich Mean Time	0:00
Universal	UTC		Greenwich Mean Time	0:00
W-SU	MSK	RU	Moscow Standard Time	4:00
Zulu	UTC		Greenwich Mean Time	0:00

Area: Europe

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Europe/Amsterdam	CET	NL	Central European Standard Time	1:00
Europe/Amsterdam	CEST	NL	Central European Summer Time	2:00
Europe/Andorra	CET	AD	Central European Standard Time	1:00
Europe/Andorra	CEST	AD	Central European Summer Time	2:00
Europe/Athens	EET	GR	Eastern European Standard Time	2:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Europe/Athens	EEST	GR	Eastern European Summer Time	3:00
Europe/Belfast	GMT	GB	Greenwich Mean Time	0:00
Europe/Belfast	BST	GB		1:00
Europe/Belgrade	CET	RS	Central European Standard Time	1:00
Europe/Belgrade	CEST	RS	Central European Summer Time	2:00
Europe/Berlin	CET	DE	Central European Standard Time	1:00
Europe/Berlin	CEST	DE	Central European Summer Time	2:00
Europe/Bratislava	CET	SK	Central European Standard Time	1:00
Europe/Bratislava	CEST	SK	Central European Summer Time	2:00
Europe/Brussels	CET	BE	Central European Standard Time	1:00
Europe/Brussels	CEST	BE	Central European Summer Time	2:00
Europe/Bucharest	EET	RO	Eastern European Standard Time	2:00
Europe/Bucharest	EEST	RO	Eastern European Summer Time	3:00
Europe/Budapest	CET	HU	Central European Standard Time	1:00
Europe/Budapest	CEST	HU	Central European Summer Time	2:00
Europe/Busingen	CET	DE	Central European Standard Time	1:00
Europe/Busingen	CEST	DE	Central European Summer Time	2:00
Europe/Chisinau	EET	MD	Eastern European Standard Time	2:00
Europe/Chisinau	EEST	MD	Eastern European Summer Time	3:00
Europe/Copenhagen	CET	DK	Central European Standard Time	1:00
Europe/Copenhagen	CEST	DK	Central European Summer Time	2:00
Europe/Dublin	GMT	IE	Greenwich Mean Time	0:00
Europe/Dublin	IST	IE		1:00
Europe/Gibraltar	CET	GI	Central European Standard Time	1:00
Europe/Gibraltar	CEST	GI	Central European Summer Time	2:00
Europe/Guernsey	GMT	GG	Greenwich Mean Time	0:00
Europe/Guernsey	BST	GG		1:00
Europe/Helsinki	EET	FI	Eastern European Standard Time	2:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Europe/Helsinki	EEST	FI	Eastern European Summer Time	3:00
Europe/Isle_of_Man	GMT	IM	Greenwich Mean Time	0:00
Europe/Isle_of_Man	BST	IM		1:00
Europe/Istanbul	EET	TR	Eastern European Standard Time	2:00
Europe/Istanbul	EEST	TR	Eastern European Summer Time	3:00
Europe/Jersey	GMT	JE	Greenwich Mean Time	0:00
Europe/Jersey	BST	JE		1:00
Europe/Kaliningrad	FET	RU	Eastern European Standard Time	3:00
Europe/Kiev	EET	UA	Eastern European Standard Time	2:00
Europe/Kiev	EEST	UA	Eastern European Summer Time	3:00
Europe/Lisbon	WET	PT	Western European Standard Time	0:00
Europe/Lisbon	WEST	PT	Western European Summer Time	1:00
Europe/Ljubljana	CET	SI	Central European Standard Time	1:00
Europe/Ljubljana	CEST	SI	Central European Summer Time	2:00
Europe/London	GMT	GB	Greenwich Mean Time	0:00
Europe/London	BST	GB		1:00
Europe/Luxembourg	CET	LU	Central European Standard Time	1:00
Europe/Luxembourg	CEST	LU	Central European Summer Time	2:00
Europe/Madrid	CET	ES	Central European Standard Time	1:00
Europe/Madrid	CEST	ES	Central European Summer Time	2:00
Europe/Malta	CET	MT	Central European Standard Time	1:00
Europe/Malta	CEST	MT	Central European Summer Time	2:00
Europe/Mariehamn	EET	AX	Eastern European Standard Time	2:00
Europe/Mariehamn	EEST	AX	Eastern European Summer Time	3:00
Europe/Minsk	FET	BY	Eastern European Standard Time	3:00
Europe/Monaco	CET	MC	Central European Standard Time	1:00
Europe/Monaco	CEST	MC	Central European Summer Time	2:00
Europe/Moscow	MSK	RU	Moscow Standard Time	4:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Europe/Nicosia	EET	CY	Eastern European Standard Time	2:00
Europe/Nicosia	EEST	CY	Eastern European Summer Time	3:00
Europe/Oslo	CET	NO	Central European Standard Time	1:00
Europe/Oslo	CEST	NO	Central European Summer Time	2:00
Europe/Paris	CET	FR	Central European Standard Time	1:00
Europe/Paris	CEST	FR	Central European Summer Time	2:00
Europe/Podgorica	CET	ME	Central European Standard Time	1:00
Europe/Podgorica	CEST	ME	Central European Summer Time	2:00
Europe/Prague	CET	CZ	Central European Standard Time	1:00
Europe/Prague	CEST	CZ	Central European Summer Time	2:00
Europe/Riga	EET	LV	Eastern European Standard Time	2:00
Europe/Riga	EEST	LV	Eastern European Summer Time	3:00
Europe/Rome	CET	IT	Central European Standard Time	1:00
Europe/Rome	CEST	IT	Central European Summer Time	2:00
Europe/Samara	SAMT	RU	Moscow Standard Time	4:00
Europe/San_Marino	CET	SM	Central European Standard Time	1:00
Europe/San_Marino	CEST	SM	Central European Summer Time	2:00
Europe/Sarajevo	CET	BA	Central European Standard Time	1:00
Europe/Sarajevo	CEST	BA	Central European Summer Time	2:00
Europe/Simferopol	MSK	UA	Eastern European Standard Time	4:00
Europe/Skopje	CET	MK	Central European Standard Time	1:00
Europe/Skopje	CEST	MK	Central European Summer Time	2:00
Europe/Sofia	EET	BG	Eastern European Standard Time	2:00
Europe/Sofia	EEST	BG	Eastern European Summer Time	3:00
Europe/Stockholm	CET	SE	Central European Standard Time	1:00
Europe/Stockholm	CEST	SE	Central European Summer Time	2:00
Europe/Tallinn	EET	EE	Eastern European Standard Time	2:00
Europe/Tallinn	EEST	EE	Eastern European Summer Time	3:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Europe/Tirane	CET	AL	Central European Standard Time	1:00
Europe/Tirane	CEST	AL	Central European Summer Time	2:00
Europe/Tiraspol	EET	MD	Eastern European Standard Time	2:00
Europe/Tiraspol	EEST	MD	Eastern European Summer Time	3:00
Europe/Uzhgorod	EET	UA	Eastern European Standard Time	2:00
Europe/Uzhgorod	EEST	UA	Eastern European Summer Time	3:00
Europe/Vaduz	CET	LI	Central European Standard Time	1:00
Europe/Vaduz	CEST	LI	Central European Summer Time	2:00
Europe/Vatican	CET	VA	Central European Standard Time	1:00
Europe/Vatican	CEST	VA	Central European Summer Time	2:00
Europe/Vienna	CET	AT	Central European Standard Time	1:00
Europe/Vienna	CEST	AT	Central European Summer Time	2:00
Europe/Vilnius	EET	LT	Eastern European Standard Time	2:00
Europe/Vilnius	EEST	LT	Eastern European Summer Time	3:00
Europe/Volgograd	VOLT	RU	Volgograd Standard Time	4:00
Europe/Warsaw	CET	PL	Central European Standard Time	1:00
Europe/Warsaw	CEST	PL	Central European Summer Time	2:00
Europe/Zagreb	CET	HR	Central European Standard Time	1:00
Europe/Zagreb	CEST	HR	Central European Summer Time	2:00
Europe/Zaporozhye	EET	UA	Eastern European Standard Time	2:00
Europe/Zaporozhye	EEST	UA	Eastern European Summer Time	3:00
Europe/Zurich	CET	CH	Central European Standard Time	1:00
Europe/Zurich	CEST	CH	Central European Summer Time	2:00

Area: Pacific

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Pacific/Apia	WST	WS	Samoa Standard Time	13:00
Pacific/Apia	WSDT	WS	Samoa Daylight Time	14:00
Pacific/Auckland	NZST	NZ	New Zealand Standard Time	12:00
Pacific/Auckland	NZDT	NZ	New Zealand Daylight Time	13:00
Pacific/Chatham	CHAST	NZ	Chatham Standard Time	12:45
Pacific/Chatham	CHADT	NZ	Chatham Daylight Time	13:45
Pacific/Chuuk	CHUT	FM	Chuuk Time	10:00
Pacific/Easter	EAST	CL	Easter Island Standard Time	-6:00
Pacific/Easter	EASST	CL	Easter Island Summer Time	-5:00
Pacific/Efate	VUT	VU	Vanuatu Standard Time	11:00
Pacific/Enderbury	PHOT	KI	Phoenix Islands Time	13:00
Pacific/Fakaofu	TKT	TK	Tokelau Time	13:00
Pacific/Fiji	FJT	FJ	Fiji Standard Time	12:00
	FJST	FJ	Fiji Summer Time	12:00
Pacific/Funafuti	TVT	TV	Tuvalu Time	12:00
Pacific/Galapagos	GALT	EC	Galapagos Time	-6:00
Pacific/Gambier	GAMT	PF	Gambier Time	-9:00
Pacific/Guadalcanal	SBT	SB	Solomon Islands Time	11:00
Pacific/Guam	CHST	GU	Chamorro Standard Time	10:00
Pacific/Honolulu	HST	US	Hawaii-Aleutian Standard Time	-10:00
Pacific/Johnston	HST	UM	Hawaii-Aleutian Standard Time	-10:00
Pacific/Kiritimati	LINT	KI	Line Islands Time	14:00
Pacific/Kosrae	KOST	FM	Kosrae Time	11:00
Pacific/Kwajalein	MHT	MH	Marshall Islands Time	12:00
Pacific/Majuro	MHT	MH	Marshall Islands Time	12:00

Time Zone Information	Time Zone Name	Region	Time Zone Description	Time Zone Offset
Pacific/Marquesas	MART	PF	Marquesas Time	-6:27
Pacific/Midway	SST	UM	Samoa Standard Time	-11:00
Pacific/Nauru	NRT	NR	Nauru Time	12:00
Pacific/Niue	NUT	NU	Niue Time	-11:00
Pacific/Norfolk	NFT	NF	Norfolk Island Time	11:30
Pacific/Noumea	NCT	NC	New Caledonia Standard Time	11:00
Pacific/Pago_Pago	SST	AS	Samoa Standard Time	-11:00
Pacific/Palau	PWT	PW	Palau Time	9:00
Pacific/Pitcairn	PST	PN	Pitcairn Time	-8:00
Pacific/Pohnpei	PONT	FM	Ponape Time	11:00
Pacific/Ponape	PONT	FM	Ponape Time	11:00
Pacific/Port_Moresby	PGT	PG	Papua New Guinea Time	10:00
Pacific/Rarotonga	CKT	CK	Cook Islands Standard Time	-10:00
Pacific/Saipan	CHST	MP	Chamorro Standard Time	10:00
Pacific/Samoa	SST	AS	Samoa Standard Time	-11:00
Pacific/Tahiti	TAHT	PF	Tahiti Time	-10:00
Pacific/Tarawa	GILT	KI	Gilbert Islands Time	12:00
Pacific/Tongatapu	TOT	TO	Tonga Standard Time	13:00
Pacific/Truk	CHUT	FM	Chuuk Time	10:00
Pacific/Wake	WAKT	UM	Wake Island Time	12:00
Pacific/Wallis	WFT	WF	Wallis and Futuna Time	12:00
Pacific/Yap	CHUT	FM	Chuuk Time	10:00

Recommended Reading

Here is the recommended reading list for this title:

- *SAS Viya Visual Data Management and Utility Procedures Guide*
- *Cody's Data Cleaning Techniques Using SAS, Second Edition*
- *SAS Viya Data Set Options: Reference*
- *SAS Viya Formats and Informats: Reference*
- *SAS Viya Functions and CALL Routines: Reference*
- *SAS Functions by Example, Second Edition*
- *SAS Viya Statements: Reference*
- *SAS Viya XML LIBNAME Engine: User's Guide*

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books

Index

Special Characters

`_LAST_` = system option 108
`%INCLUDE` statement
 starting position for reading variable-sized record input 168

A

address
 of metadata server 131
 alignment of output 55
`ALIGNSASIOFILES` system option 37
 alternative configuration file 61
`ALTLOG` system option 38
`APPEND=` system option 39
 modify system option values 13
 autocalled libraries
 specifying 170
 autocalled macro files
 starting position for reading variable-sized record input 168
`AUTOCORRECT` system option 41
`AUTOEXEC` file
 echoing to the log 76
 autoexec files
 specifying 43
 starting position for reading variable-sized record input 168
`AUTOEXEC` system option 43

B

batch mode
 destination for output 154
 Log window destination 111
 source code, default location of 201
 batch processing
 DATA and PROC step checkpoint-restart data 192
 error handling 79
 labeled code sections checkpoint-restart data 107
 recording checkpoint-restart data for 104, 189

buffers

 aligning data for I/O optimization 37
 extra buffers for navigating index files 95
 number for data sets 45
 optimal size 64
 page buffers for catalogs 54
 size of 47
 utility 207
 view size 221
 writing to disk 85
`BUFNO=` system option 45
`BUFSIZE=` system option 47
 BY lines
 printing 49
 BY variables
 existing in one data set but not another 224
`BYERR` system option 48
`BYLINE` system option 49
`BYSORTED` system option 50
 bytes
 number sorted 178

C

`CAPS` system option 51
`CARDIMAGE` system option 52
 case
 for notes, warnings, and messages 136
 catalogs
 error handling 72
 number to keep open 53
 page buffers 54
 search order for 88
`CATCACHE=` system option 53
`CBUFNO=` system option 54
`CENTER` system option 55
`CGOPTIMIZE=` system option 56
 character combinations 57
`CHARCODE` system option 57
 checkpoint-restart data
 erase Work library 58

- libref for DATA and PROC step library 191
 - libref for labeled code section library 105
 - recording for DATA and PROC steps 189
 - recording for labeled code sections 104
 - specifying batch programs with 107, 192
 - CHKPTCLEAN system option 58
 - code compilation
 - optimization level during 56
 - code submission
 - options changed by 7
 - commands
 - executing statements automatically 43
 - for printing tasks 118, 154
 - compilation
 - optimization level during 56
 - COMPRESS= system option 59
 - compressed data sets
 - reusing space when adding observations 159
 - compressing observations 59
 - CONFIG system option 61
 - configuration files
 - specifying 61
 - connection profiles
 - for connecting to metadata server 122
 - XML document containing 128
 - CPUCOUNT= system option 62
 - CPUID system option 63
- D**
- damaged data sets or catalogs 72
 - data lines
 - as card images 52
 - length of sequence field 173
 - data set options
 - system options and 17
 - data sets
 - buffer size 47, 64
 - compressing on output 59
 - damaged 72
 - most recently created 108
 - not found 74
 - number of buffers 45
 - number of bytes sorted 178
 - replacing permanently stored 158
 - reusing space in compressed data sets 159
 - DATA statement
 - keywords allowed in 65
 - DATA step 73
 - DATALINES statement
 - length of data 163
 - DATAPAGESIZE= system option 64
 - DATASTMTCHK= system option 65
 - date and time data 193
 - date informats and functions
 - year cutoff 233
 - date stamp 66
 - DATE system option 66
 - DATESTYLE= system option 66
 - datetime informats and functions
 - year cutoff 233
 - DECIMALCONV= system option 67
 - DEFINE option
 - PROC OPTIONS statement 249
 - DETAILS system option 68
 - directory
 - creating 71
 - displaying system option information 253
 - DKRCOND= system option 69
 - DKROCOND= system option 70
 - DLCREATEDIR system option 71
 - DLDMGACTION= system option 72
 - DROP statement
 - error detection for output data sets 70
 - DROP= data set option
 - error detection for input data sets 69
 - DROP= DATA step option
 - error detection for output data sets 70
 - DSACCEL= system option 73
 - DSNFERR system option 74
 - DTRESET system option 75
- E**
- ECHO system option 75
 - ECHOAUTO system option 76
 - echoing messages to computer 75
 - encryption 124
 - level of 125
 - environment variables
 - specifying multiple in OPTIONS statement 172
 - error detection levels
 - input data sets 69
 - output data sets 70
 - error handling
 - BY-group processing 78
 - catalogs 72
 - format not found 88
 - in batch processing 79
 - numeric data 102
 - error messages
 - BY variable exists in one data set but not another 224
 - maximum number printed 80
 - overprinting 150

- SORT procedure 48
- error processing
 - variables not initialized 217
- error response 77
- ERRORABEND system option 77
- ERRORBYABEND system option 78
- ERRORCHECK= system option 79
- errors
 - displaying messages in uppercase 136
 - library of SAS error messages 135
 - specifying stdin, stdout, and stderr 189
- ERRORS= system option 80
- executing SAS statements
 - autoexec file 43
- external files
 - logical record length for reading and writing 117
- external programs
 - passing parameter strings to 152

F

- file locking 81
- file permissions
 - locking files 81
 - Work data library 231
- FILELOCKS system option 81
- FILELOCKWAIT= system option 83
- FILELOCKWAITMAX= system option 84
- FILESYNC= system option 85
- FIRSTOBS= system option 86
- FMTERR system option 88
- FMTSEARCH= system option 88
- format catalog
 - associated locale 88
- format catalogs
 - search order for 88
- formats
 - name length 212
 - not found 88
- formatting characters 91
- FORMCHAR= system option 91
- FULLSTIMER system option 92, 193

G

- GETOPTION function 237
 - changing YEARCUTOFF system option with 240
 - obtaining reporting options 240
- graphics options
 - returning value of 237
- GROUP= option
 - PROC OPTIONS statement 250

H

- hardware information, writing to SAS log 63
- hexadecimal values
 - for system options 6
- HEXVALUE option
 - PROC OPTIONS statement 250
- holidays
 - user-supplied 101
- host name or address 131
- HOST option
 - PROC OPTIONS statement 250
- host sort utility
 - passing options to 176
 - passing parameters to 182
 - specifying if used 182
 - temporary files used by 179
 - when to use, based on quantity of observations 177
- HOSTINFOLONG system option 95

I

- I/O optimization 37
- IBUFNO= system option 95
- IBUFSIZE 97
- IBUFSIZE= system option 97
- index files
 - extra buffers for navigating 95
- informats
 - ambiguous data 66
 - name length 212
- INITSTMT= system option 98
- input
 - as card images 52
 - uppercasing 51
- input data sets
 - error detection levels 69
- input source lines
 - length of numeric portion of sequence field 173
- INSERT system option 99
- INSERT= system option
 - modify system option values 13
- Integrated Object Model (IOM) server 141
- interval values
 - user-supplied holidays 101
- INTERVALDS= system option
 - user-supplied holidays 101
- invalid data
 - numeric 102
- INVALIDDATA= system option 102
- IS= system option 98
- item stores
 - naming 213

J

Java Runtime Environment options 103
 JRE (Java Runtime Environment) options 103
 JREOPTIONS system option 103

K

KEEP statement
 error detection for output data sets 70
 KEEP= data set option
 error detection for input data sets 69
 KEEP= DATA step option
 error detection for output data sets 70
 keyboard 57
 keywords, allowed in DATA statement 65

L

LABEL system option 103
 LABELCHKPT system option 104
 LABELCHKPTLIB system option 105
 LABELRESTART system option 107
 labels
 using with variables in SAS procedures 103
 libraries
 creating a nonexistent library 71
 damaged data sets or catalogs 72
 default permanent SAS library 209
 error messages 135
 listing details 68
 SAS library to use as SASUSER library 172
 Work library 228, 231
 librefs
 assigning user-defined, at startup 188
 license information
 altering 175
 LINESIZE= system option 109
 LISTGROUPS option
 PROC OPTIONS statement 250
 LISTINSERTAPPEND option
 PROC OPTIONS statement 250
 LISTRESTRICT option
 PROC OPTIONS statement 250
 locale
 associated catalog 88
 LOCKDOWN system option 110
 locked files
 maximum wait time for 84
 locking files 81
 log
 destination for 38, 111
 writing all system performance statistics to 92, 193

 writing some system performance statistics to 193, 197
 writing system option settings to 149
 log files 112
 LOG system option 111
 Log window
 echoing messages to 75
 logical record length
 for reading and writing external files 117
 LOGNUMBERFORMAT option
 PROC OPTIONS statement 251
 LOGPARM= system option 112
 LONG option
 PROC OPTIONS statement 251
 lp command 118, 154
 lpr command 118, 154
 LPTYPE system option 118
 LRECL= system option 117
 LS= system option 109

M

macro facility
 autocall libraries 170
 memory for in-memory macro variables 138
 memory for macro variable symbol tables 138
 MAXMEMQUERY system option 118
 memory
 allocating for certain procedures 118
 allocating for in-memory macro variables 138
 allocating for macro variable symbol tables 138
 allocating for SAS sessions 119
 amount of real memory available 157
 SORT procedure 183
 MEMSIZE system option 119
 MERGE processing
 without BY statement 122
 MERGENOBY system option 122
 messages
 detail level of 136
 MERGE processing without BY statement 122
 news file for writing to SAS log 139
 printing to SAS log, all vs. top-level 155
 METACONNECT 122
 METACONNECT= system option 122
 metadata server
 connection profile for connecting to 122
 encryption level 125

- encryption type 124
 - host name or address 131
 - network protocol for connecting to 129
 - password for 125
 - SAS Metadata Repository for 130
 - SPN for 132
 - TCP port for 127
 - user ID for 133
 - XML document containing connection profiles for 128
 - METAENCRYPTALG system option 124
 - METAENCRYPTALGORITHM 124
 - METAENCRYPTLEVEL 125
 - METAENCRYPTLEVEL system option 125
 - METAPASS 125
 - METAPASS= system option 125
 - METAPORT 127
 - METAPORT= system option 127
 - METAPROFILE system option 128
 - METAPROTOCOL 129
 - METAPROTOCOL= system option 129
 - METAREPOSITORY= system option 130
 - METASERVER= system option 131
 - METASPN= system option 132
 - METAUSER= system option 133
 - missing values
 - character to print for numeric values 134
 - MISSING= system option 134
 - misspelled names
 - autocorrection of 41
 - MSG system option 135
 - MSGCASE system option 136
 - MSGLEVEL= system option 136
 - MSYMTABMAX system option 138
 - MVARSIZE system option 138
- N**
- names
 - host name of metadata server 131
 - network protocol
 - for connecting to metadata server 129
 - NEWS= system option 139
 - NOEXPAND option
 - PROC OPTIONS statement 251
 - NOHOST option
 - PROC OPTIONS statement 251
 - NOLOGNUMBERFORMAT option
 - PROC OPTIONS statement 251
 - notes
 - writing to SAS log 140
 - NOTES system option 140
 - notes, displaying in uppercase 136
 - NUMBER system option 140
 - numeric data
 - character to print for missing values 134
 - invalid 102
- O**
- OBJECTSERVER system option 141
 - OBS= system option 142
 - comparisons 143
 - data set with deleted observations 147
 - details 143
 - examples 144
 - WHERE processing with 145
 - observations
 - compressing on output 59
 - error messages, number printed 80
 - starting at a specific 86
 - stopping processing 142
 - observations, sorting
 - host sort utility, passing options to 176
 - host sort utility, passing parameters to 182
 - host sort versus SAS sort 182
 - location of temporary files for 179
 - number of observations and 177
 - OPLIST system option 149
 - optimization
 - during code compilation 56
 - OPTION= option
 - PROC OPTIONS statement 251
 - OPTIONS procedure
 - display settings for a group of options 255
 - overview 247
 - results 258
 - syntax 248
 - OPTIONS statement 243
 - specifying system options 5
 - values changed by SAS Studio 7
 - output
 - aligning 55
 - compressing 59
 - destination for 154
 - echoing messages to computer 75
 - formatting characters 91
 - overprinting error messages 150
 - page size 152
 - spooling 187
 - output data sets
 - error detection 70
 - optimizing performance 37
 - overprinting error messages 150
 - OVP system option 150

P

page buffers
 for catalogs 54
 page numbers
 printing in title line of each page 140
 resetting 151
 page size 152
 PAGENO= system option 151
 PAGESIZE= system option 152
 parallel processing 73
 parameter strings
 passing to external programs 152
 PARM= system option 152
 PARMCARDS statement
 file reference to open for 153
 PARMCARDS= system option 153
 passwords
 for metadata server 125
 performance 198
 amount of real memory available 157
 sorting method, influenced by number
 of observations 177
 writing all statistics to log 92, 193
 writing some statistics to log 193, 197
 permissions
 Work data library 231
 port
 TCP port for metadata server 127
 PORTABLE option
 OPTIONS procedure 251
 PRINT system option 154
 PRINTCMD system option 154
 printing
 overprinting error messages 150
 page numbers in title line of each page
 140
 printing output
 commands and settings for 118, 154
 PRINTMSGLIST system option 155
 PROC OPTIONS statement 248
 procedure output
 aligning 55
 destination for 154
 linesize 109
 procedures
 allocating memory for requests 118
 using labels with variables 103
 PS= system option 152
 punctuating numbers 251

Q

QUOTELENMAX system option 156

R

reading external files
 resources read during execution 161,
 162
 reading variable-sized record input 168
 REALMEMSIZE system option 157
 records
 stopping processing 142
 RENAME statement
 error detection for output data sets 70
 RENAME= data set option
 error detection for input data sets 69
 RENAME= DATA step option
 error detection for output data sets 70
 REPLACE system option 158
 repository
 SAS Metadata Repository for metadata
 server 130
 resetting system option values 15, 237
 RESTRICT option
 PROC OPTIONS statement 251
 restricted options 8
 REUSE= system option 159
 RSASUSER system option 160
 RTRACE system option 161
 RTRACELOC system option 162

S

S= system option 163
 S2= system option 165
 S2V= system option 168
 SAS data set
 optimizing 64
 SAS data sets
 naming 213
 SAS data views
 naming 213
 SAS files
 resources read during execution 161,
 162
 SAS invocation
 initializing Work library at 230
 SAS log
 AUTOEXEC input 76
 date and time, printing 66
 detail level of messages 136
 news file for writing messages to 139
 printing messages to, all vs. top-level
 155
 writing hardware information to 63
 writing host information to 95
 writing notes to 140
 writing secondary source statements to
 186
 writing source statements to 186

- SAS Metadata Repository
 - for metadata server 130
- SAS OPTIONS window, compared to
 - OPTIONS statement 244
- SAS procedures
 - using labels with variables 103
- SAS sessions
 - allocating memory for 119
 - associating terminal with 202
 - executing statements at termination 203
 - X command, specifying if valid 232
- SAS statements
 - autoexec file 43
 - executing at startup 98
 - writing to utility data set in Work data
 - library 187
- SAS system options
 - changing values of 243
- SASAUTOS system option 170
- SASHELP library
 - location of 169
- SASHELP= system option 169
- SASMSG
 - MSG system option 135
- SASUSER library
 - opening for read or read-write access
 - 160
 - SAS library to use as 172
- SASUSER= system option 172
- search order
 - format catalogs 88
- secondary source statements
 - writing to SAS log 186
- security
 - locking files 81
 - permissions on Work data library 231
- SEQ= system option 173
- sequence field
 - length of numeric portion 173
- SERVICESBASEURL system option 173
- SET system option 174
- SETINIT system option 175
- SHORT option
 - PROC OPTIONS statement 251, 252
- site license information
 - altering 175
- sort order
 - verifying user-specified 184
- SORT procedure
 - error messages 48
 - memory for 183
 - removing duplicate variables 179
 - verifying user-specified sort order 184
- SORTANOM system option 176
- SORTCUT system option 177
- SORTCUTP system option 178
- SORTDEV system option 179
- SORTDUP= system option 179
- SORTEQUALS 180
- SORTEQUALS system option 180
- sorting
 - number of bytes sorted 178
- sorting observations
 - host sort utility, passing options to 176
 - host sort utility, passing parameters to
 - 182
 - host sort versus SAS sort 182
 - location of temporary files for 179
 - number of observations and 177
- SORTNAME system option 181
- SORTPARM system option 182
- SORTPGM system option 182
- SORTSIZE= system option 183
- SORTVALIDATE system option 184
- source lines
 - as card images 52
- source statements
 - length of 163, 165
 - writing secondary statements to SAS
 - log 186
 - writing to SAS log 186
- SOURCE system option 186
- SOURCE2 system option 186
- special characters not on keyboard 57
- SPN (service principal name) 132
- SPOOL system option 187
- standard input/output
 - specifying if SAS should use 189
- starting position
 - for reading variable-sized record input
 - 168
- STARTLIB system option 188
- startup
 - assigning user-defined permanent librefs
 - at 188
 - executing SAS statements at 98
- statements
 - executing at termination of SAS
 - sessions 203
 - length of 163, 165
 - writing to utility data set in Work data
 - library 187
- stderr 189
- stdin 189
- STDIO system option 189
- stdout 189
- STEPCHKPT system option 189
- STEPCHKPTLIB= system option 191
- STEPRESTART system option 192
- STIMEFMT system option 193
- STIMER system option 197
 - controlling format of 193

- STRIPESIZE= system option 198
- syncsort utility
 - location of temporary files used by 179
 - passing options to 176
 - passing parameters to 182
 - specifying if used 182
 - when to use, based on quantity of observations 177
- syntax checking 200
- SYNTAXCHECK system option 200
- SYSIN system option 201
- system options 3
 - change using INSERT and APPEND 13
 - comparisons 17
 - data set options and 17
 - default settings 4
 - determining current settings 6
 - determining how set 10
 - determining how value was set 11
 - determining restricted options 8
 - display setting for single option 259
 - display settings for a group 255
 - displaying a list 252
 - displaying information about 253
 - displaying restricted options 257
 - duration of settings 7
 - hexadecimal values for 6
 - in other publications 3
 - information about 12
 - list of current settings 247
 - OPTIONS procedure 247
 - order of precedence 17
 - resetting default and starting values 15, 237
 - restricted 8
 - returning value of 237
 - short form listing 259
 - spaces in the value 4
 - specifying in configuration file 4
 - specifying in OPTIONS statement 5
 - specifying on command line 4
 - syntax 4
 - writing settings to log 149
 - writing settings to terminal 223
- T**
- TCP port
 - for metadata server 127
- TERMINAL system option 202
- terminals
 - associating with SAS session 202
- TERMSTMT= system option 203
- threaded processing 204
- threads
 - concurrent processing 62
- THREADS system option 204
- time stamp 66
- TIMESZONE= system option 205
- title lines
 - printing page numbers in 140
- trace information 161, 162
- TZ= system option 205
- U**
- UBUFNO= system option 207
- UBUFSIZE= system option 208
- uppercase
 - translating input to 51
- user ID
 - for metadata server 133
- user-defined librefs
 - assigning at startup 188
- user-supplied holidays 101
- USER= system option 209
- USERCONFIG system option 209
- utility files
 - buffer size 64, 208
 - number of buffers 207
- UTILLOC= system option 210
- V**
- VALIDFMTNAME 212
- VALIDFMTNAME= system option 212
- VALIDMEMNAME system option 213
- VALIDVARNAME= system option 215
- VALUE option
 - PROC OPTIONS statement 252
- variable names
 - rules for valid names 215
- variable-sized record input
 - starting position for reading 168
- variables
 - format not found 88
 - using labels with, in SAS procedures 103
- VARINITCHK= system option 217
- VARLENCHK= system option 218
- VBUFSIZE= system option 221
- VERBOSE system option 223
- VNFERR system option 224
- W**
- warnings
 - displaying in uppercase 136
- WHERE processing
 - OBS= system option with 145
- Work data library
 - erasing checkpoint-restart data 58

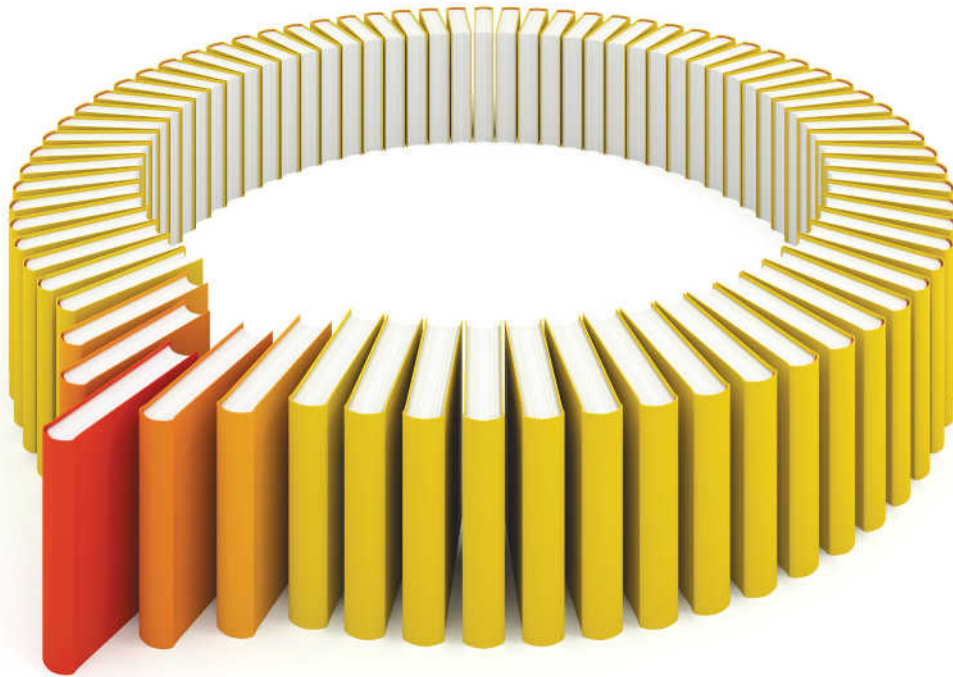
- initializing at SAS invocation 230
- writing SAS statements to utility data
 - set in 187
- Work files
 - erasing at end of session 231
- Work library
 - name of 228
 - setting permissions when created 231
- WORK system option 228
- WORKINIT system option 230
- WORKPERMS system option 231
- WORKTERM system option 231

X

- X command
 - specifying if valid 232
- XCMD system option 232
- XML documents
 - containing connection profiles 128

Y

- YEARCUTOFF= system option 233
 - changing with GETOPTION function 240



Gain Greater Insight into Your SAS[®] Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

 support.sas.com/bookstore
for additional books and resources.


THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

