MWSUG 2025 – Paper 9

You've Got Options: Ten Five-Star System Option Hacks

Louise S. Hadden, Independent Consultant

SAS® provides myriad opportunities for customizing programs and processes, including a wide variety of system options that can control and enhance SAS code from start to finish. This paper and presentation demonstrates methods of obtaining information on SAS system options and moves on to fully explicate ten SAS system option hacks, from COMPRESS to XWAIT. System options are highly dependent on platforms, security concerns, SAS versions and products: dependencies and defaults will be discussed. SAS practitioners will gain a deeper understanding of the powerful SAS system options they have seen, used, and automatically included in their code. This presentation is suitable for all skill and experience levels; platform and implementation differences are part of the discussion.

INTRODUCTION

This paper and presentation is the first part of a multi-part series: System Options, Data Set Options, Procedural Options, and SAS Functions. A walk through of discovery (finding out as much information as possible about the System Options available in each SAS Session) follows, as well as an in-depth discussion of ten five-star System Options.

DISCOVERING YOUR SAS SYSTEM OPTIONS

SAS DOCUMENTATION

SAS has robust documentation of SAS System Options, arranged in semi-alphabetical order, located at https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/lesysoptsref/p1tmgku1vq7pwqn1iqioeflxgec1.htm. You can toggle SAS System Options by Category and view SAS System Options documented in other SAS publications.

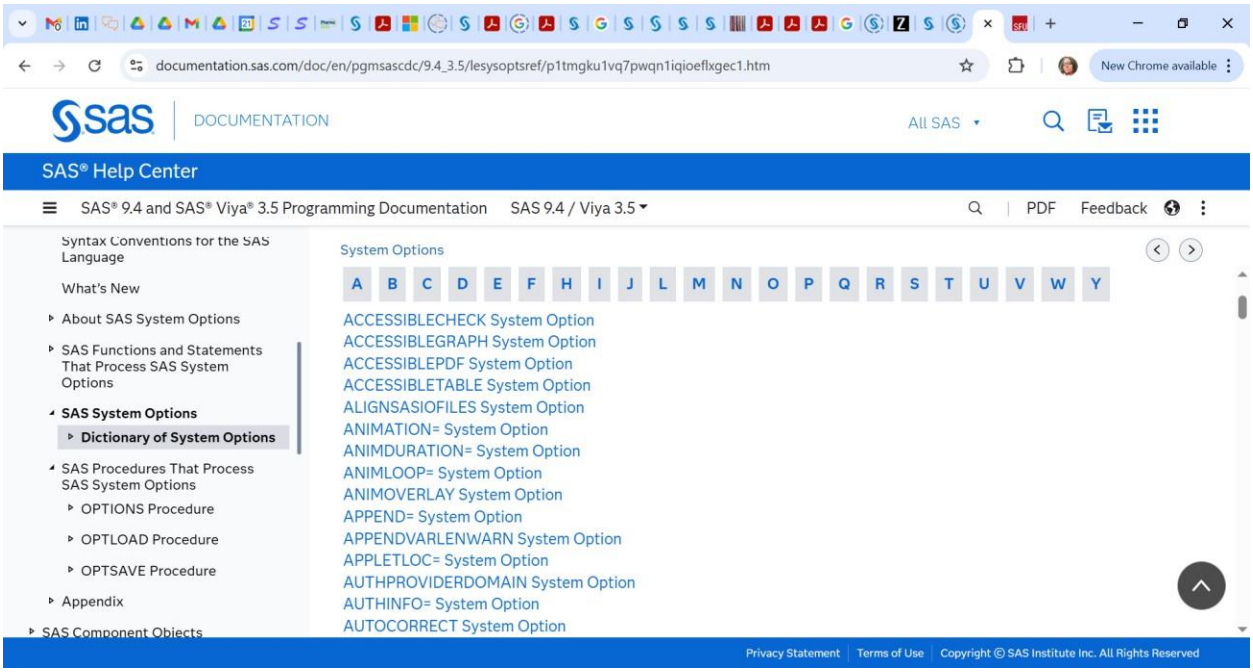


Figure 1. Screenshot of SAS documentation of System Options

Clicking on any of the system option links will get you to a short paragraph about the system option, what system(s) it is valid in, what category(s) it applies to, what PROC OPTIONS Group the system option belongs to, the default value, and whether it can be restricted by a site admin. Following this information, you can view syntax of the system option with some details.

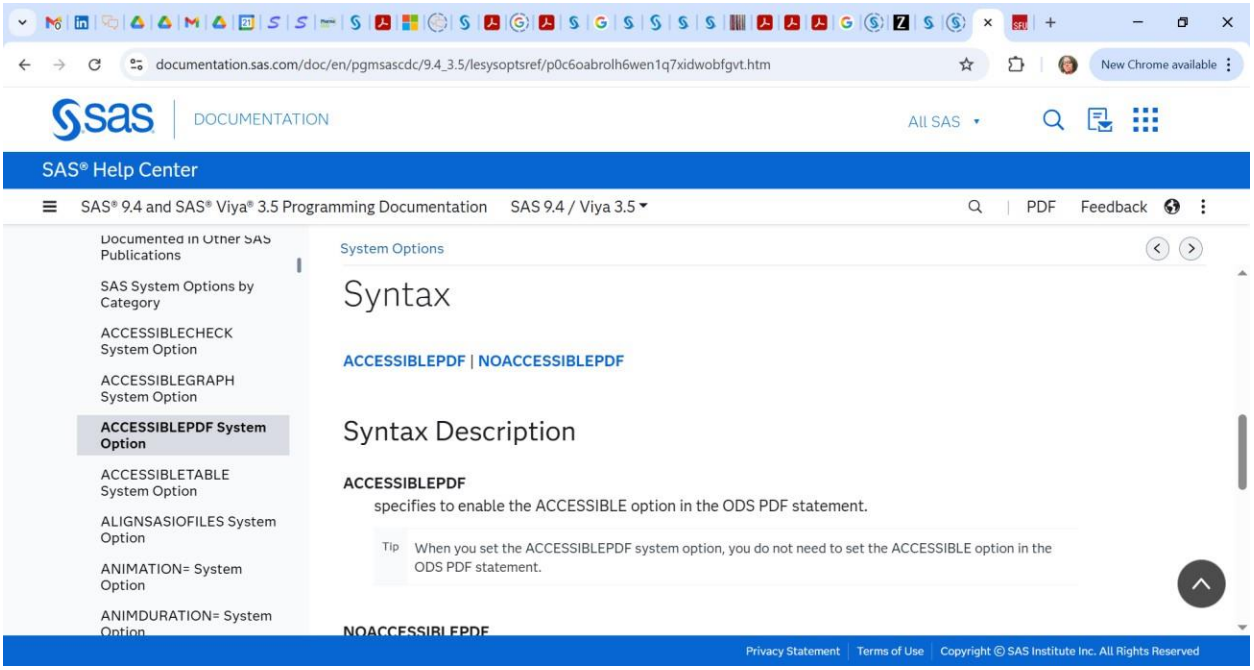


Figure 2. Screenshot of Syntax Detail for System Syntax Documentation

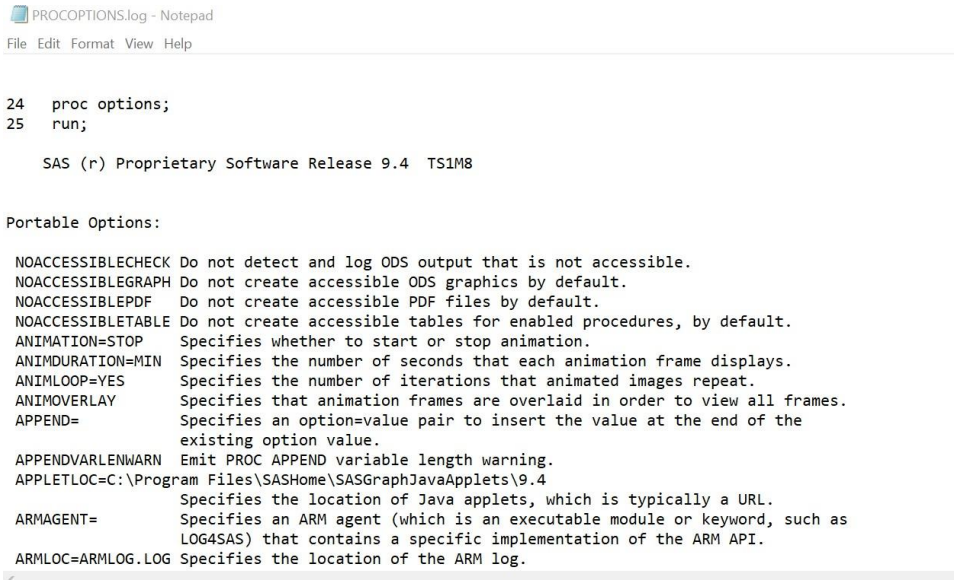
PROC OPTIONS

PROC OPTIONS provides a programmatic solution and writes a list of options *currently in place during a SAS session* to a SAS Log. In this case, we use PROC PRINTTO to generate an external file.

```
libname dd 'C:\SAS_DATA\Options';
filename odsout 'C:\SAS_DATA\Options';
title1 "You've Got Options";
run;

proc printto log="C:\SAS_DATA\Options\PROCOPTIONS.log" new;

proc options;
run;

proc printto;
run;
```

Program 1. PROC OPTIONS

The listing in the log output snippet below is informative to an extent but does not really offer a lot of information on each option; additionally, the formatting is inutile. Note that PROC OPTIONS reports on system options AS IS; that is, it details the options that are in place at the current time, in the current session, the current implementation of SAS, and the current platform.



PROCOPTIONS.log - Notepad

File Edit Format View Help

```
24   proc options;
25   run;

    SAS (r) Proprietary Software Release 9.4  TS1M8


Portable Options:

 NOACCESSIBLECHECK Do not detect and log ODS output that is not accessible.
 NOACCESSIBLEGRAPH Do not create accessible ODS graphics by default.
 NOACCESSIBLEPDF   Do not create accessible PDF files by default.
 NOACCESSIBLETABLE Do not create accessible tables for enabled procedures, by default.
 ANIMATION=STOP    Specifies whether to start or stop animation.
 ANIMDURATION=MIN  Specifies the number of seconds that each animation frame displays.
 ANIMLOOP=YES      Specifies the number of iterations that animated images repeat.
 ANIMOVERLAY       Specifies that animation frames are overlaid in order to view all frames.
 APPEND=           Specifies an option=value pair to insert the value at the end of the
                   existing option value.
 APPENDVARLENWARN  Emit PROC APPEND variable length warning.
 APPLETLOC=C:\Program Files\SASHome\SASGraphJavaApplets\9.4
                   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=ARMLOG.LOG Specifies the location of the ARM log.
```

Figure 3. PROC OPTIONS Log Listing

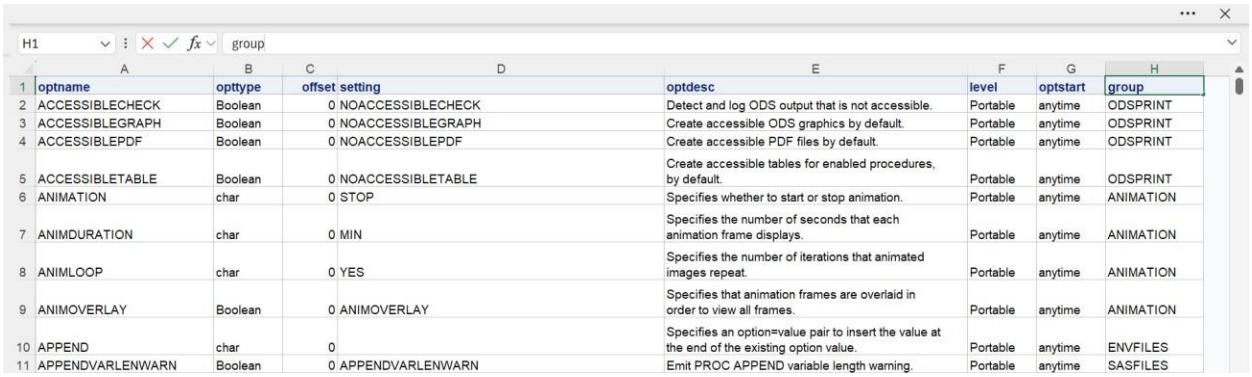
Options for PROC OPTIONS include adding group= so you can subset an options group of the list of almost 400 options.

SAS DICTIONARY TABLES / V TABLES

A second procedural option is to use SAS Dictionary Tables (yes, SAS has a Dictionary table and view for Options, as well as many other SAS structures).

```
ods excel file="C:\SAS_DATA\Options\voption_desc.xlsx" ;
proc print data=sashelp.voption noobs;
run;
ods excel close;
```

Program 2. ACCESS SASHELP.VOPTION table



H1 group

	A	B	C	D	E	F	G	H
1	optname	opttype	offset	setting	optdesc	level	optstart	group
2	ACCESSIBLECHECK	Boolean	0	NOACCESSIBLECHECK	Detect and log ODS output that is not accessible.	Portable	anytime	ODSPRINT
3	ACCESSIBLEGRAPH	Boolean	0	NOACCESSIBLEGRAPH	Create accessible ODS graphics by default.	Portable	anytime	ODSPRINT
4	ACCESSIBLEPDF	Boolean	0	NOACCESSIBLEPDF	Create accessible PDF files by default.	Portable	anytime	ODSPRINT
5	ACCESSIBLETABLE	Boolean	0	NOACCESSIBLETABLE	Create accessible tables for enabled procedures, by default.	Portable	anytime	ODSPRINT
6	ANIMATION	char	0	STOP	Specifies whether to start or stop animation.	Portable	anytime	ANIMATION
7	ANIMDURATION	char	0	MIN	Specifies the number of seconds that each animation frame displays.	Portable	anytime	ANIMATION
8	ANIMLOOP	char	0	YES	Specifies the number of iterations that animated images repeat.	Portable	anytime	ANIMATION
9	ANIMOVERLAY	Boolean	0	ANIMOVERLAY	Specifies that animation frames are overlaid in order to view all frames.	Portable	anytime	ANIMATION
10	APPEND	char	0		Specifies an option=value pair to insert the value at the end of the existing option value.	Portable	anytime	ENVFILES
11	APPENDVARLENWARN	Boolean	0	APPENDVARLENWARN	Emit PROC APPEND variable length warning.	Portable	anytime	SASFILES

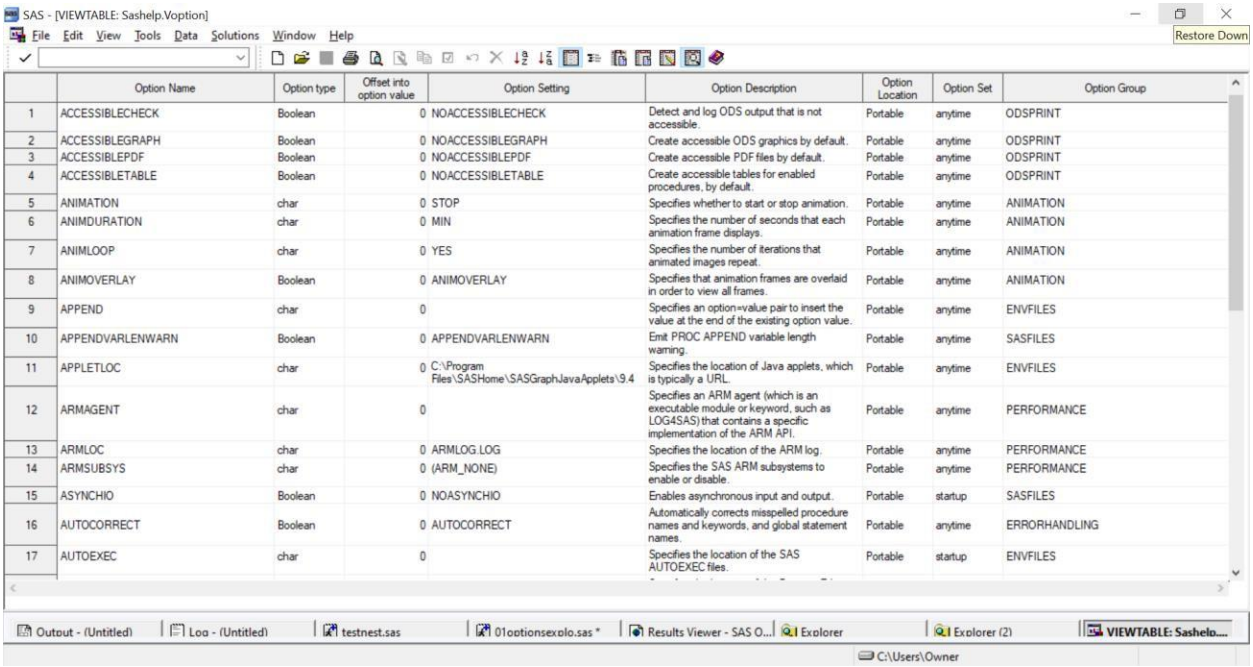
Figure 3. Output Excel Sheet of SASHELP.VOPTION table

Columns available in the SASHELP.VOPTION table include: optname (Option Name), optype (Option type [char, num, boolean], offset (Offset into option value), setting (Option setting), optdesc (Option Description), level (Option Location), optstart (Option Set), and group (Option Group). Note that the setting column (D) shows the current setting of System Options, and not all setting possibilities.

Both PROC OPTIONS and the Options Dictionary Table / SASHELP.Voptions will only inform you of the “current state” of both your program and/or system. Site administrators disallow some system options,

and you will not even see those options in a listing. For example, one of the sites I work with does not allow the use of the DLCREATEDIR option, while the site I am testing this paper on does. If you find a system option you would like to use, you will need to consult your site administrators. Additionally, some options are SAS instance specific, for example, AZURETENANTID.

You can also see the SASHELP.VOPTION VIEWTABLE by going to the Explorer window in display manager and clicking on the SASHELP.VOPTION icon. Column labels are displayed in this view, but the file is identical to the one explored above.



	Option Name	Option type	Offset into option value	Option Setting	Option Description	Option Location	Option Set	Option Group
1	ACCESSIBLECHECK	Boolean	0	NOACCESSIBLECHECK	Detect and log ODS output that is not accessible.	Portable	anytime	ODSPRINT
2	ACCESSIBLEGRAPH	Boolean	0	NOACCESSIBLEGRAPH	Create accessible ODS graphics by default.	Portable	anytime	ODSPRINT
3	ACCESSIBLEPDF	Boolean	0	NOACCESSIBLEPDF	Create accessible PDF files by default.	Portable	anytime	ODSPRINT
4	ACCESSIBLETABLE	Boolean	0	NOACCESSIBLETABLE	Create accessible tables for enabled procedures, by default.	Portable	anytime	ODSPRINT
5	ANIMATION	char	0	STOP	Specifies whether to start or stop animation.	Portable	anytime	ANIMATION
6	ANIMDURATION	char	0	MIN	Specifies the number of seconds that each animation frame displays.	Portable	anytime	ANIMATION
7	ANIMLOOP	char	0	YES	Specifies the number of iterations that animated images repeat.	Portable	anytime	ANIMATION
8	ANIMOVERLAY	Boolean	0	ANIMOVERLAY	Specifies that animation frames are overlaid in order to view all frames.	Portable	anytime	ANIMATION
9	APPEND	char	0		Specifies an option=value pair to insert the value at the end of the existing option value.	Portable	anytime	ENVFILES
10	APPENDVARLENWARN	Boolean	0	APPENDVARLENWARN	Emit PROC APPEND variable length warning.	Portable	anytime	SASFILES
11	APPLETLOC	char	0	C:\Program Files\SASHome\SASGraphJavaApplets\9.4	Specifies the location of Java applets, which is typically a URL.	Portable	anytime	ENVFILES
12	ARMAGENT	char	0		Specifies an ARM agent (which is an executable module or keyword, such as LOG4SAS) that contains a specific implementation of the ARM API.	Portable	anytime	PERFORMANCE
13	ARMLOC	char	0	ARMLOG.LOG	Specifies the location of the ARM log.	Portable	anytime	PERFORMANCE
14	ARMSUBSYS	char	0	(ARM_NONE)	Specifies the SAS ARM subsystems to enable or disable.	Portable	anytime	PERFORMANCE
15	ASYNCHIO	Boolean	0	NOASYNCHIO	Enables asynchronous input and output.	Portable	startup	SASFILES
16	AUTOCORRECT	Boolean	0	AUTOCORRECT	Automatically corrects misspelled procedure names and keywords, and global statement names.	Portable	anytime	ERRORHANDLING
17	AUTOEXEC	char	0		Specifies the location of the SAS AUTOEXEC files.	Portable	startup	ENVFILES

SAS SYSTEM OPTIONS CATEGORIES, GROUPS, AND SUBGROUPS

https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/hostwin/n0qn87565ybxoun12srorc6xhpz5.htm

TOP TEN SYSTEM OPTION HACKS

It was incredibly difficult to just choose ten incredible system options from all the choices available. I am going to cheat just a little bit by grouping options that naturally go together in my programming work. Here are my choices with an explanation of what they do for me, and hopefully for you. If you are running in a non-windowing environment, I hope you run PROC OPTIONS and have a fun time finding the differences in defaults for options and/or new-to-you options and let me know.

VALIDVARNAME

I use this option every programming day, as the main transfer formats for data in my workplace are XLS, XLSX, and CSV. The VALIDVARNAME= system option can have several values, each controlling how SAS handles column names from different systems. “V7” is often the default, but not in every system. Invalid characters, in particular blanks or nulls, are replaced with underscores, and SAS automatically adds a counter to variable names that are not unique. Other options include “UPCASE” which converts all variables to uppercase, and “ANY” which may require the use of a name literal.

My client receives data with spaces and special characters in column names, making it impossible to import XLS, XLSX, or CSV files with this option.

Syntax: OPTIONS VALIDVARNAME=V7;

VARLENCHK

It is common to receive a warning from this option when there is a mismatch between the length of a variable being read in and either a length statement for that variable or another data set being ready in the same data step. If I had a dime for every time I have seen this warning in the thousands of logs I have reviewed I would be a rich woman. The value options are NOWARN, WARN, or ERROR; that is, you can receive no warning in the log, a warning, or an error for the respective option values. For development, I like to use ERROR, so I can fix the problem. For production, I leave it on WARN which is the default.

Syntax: OPTIONS VARLENCHK=ERROR;

MERGENOBY

This option is exactly what it sounds like; SAS determines whether a MERGE statement is occurring without a BY statement, and will ignore (i.e., NOWARN), warn (WARN), or trigger an error in the log (ERROR). Again, for development I like to use ERROR, and for production put it on WARN. The default on my system is NOWARN, so I need to reset this option for my runs. There are occasions (when I am merging a one-line summary with a one-line summary) where I willfully ignore the warning. For runs that will be reviewed by an auditor, the default will work (no warnings or errors generated).

Syntax: OPTIONS MERGENOBY=ERROR;

DLCREATEDIR

This is a fantastic option which allows the automatic creation of subdirectories where one does not exist. The default is different depending on what system you are using: under UNIX, Windows, and SAS Viya, the shipped default is NODLCREATEDIR. Under z/OS the shipped default is DLCREATEDIR. However, with good reason, this option is frequently restricted by site admins. While this option will not overwrite existing data in place, if it is used in conjunction with a macro, it could spawn hundreds of subdirectories accidentally. Nonetheless, it is one of my favorite options, and I have used it monthly to create a complicated directory structure for processing (Hadden, “Get Smart”, 2017).

Syntax: OPTIONS DLCREATEDIR;

FMTERR

This option has two values, FMTERR (generate an error message if a specified format is not found for a variable), or NOFMTERR (allow processing to continue, replacing the specified format with the default w. or $w. format and putting a note in the log.) This is invaluable if you are working with a formatted data set and did not receive an accompanying format catalog that is available on your platform, CNTLIN SAS data set, or program to generate a format catalog. Format catalogs are notoriously difficult to transfer across systems and platforms, as well. This is a case where I wish there were a warning level because the lack of appropriate format is a serious concern, but often, we are under the gun to get data processed so we can clean and address problems. For example, during the pandemic we received data from a text-based survey system (REDCAP) from multiple sites, and if one site collected a data element and another did not, the formats would often mismatch. (Hadden, “Putting the Meta”, 2022). We often had two to three days to turn around incoming data for the CDC and White House briefings, so time was of the essence and the head start from NOFMTERR facilitated on time delivery.

Syntax: OPTIONS NOFMTERR; or OPTIONS FMTERR;

COMPRESS

The COMPRESS option has four values: NO, YES, CHAR, and BINARY. NO indicates no compression, YES, CHAR, and BINARY enables compression. YES and BINARY use run-length encoding (RLE): this compression type reduces repeated characters to two- or three-byte representations. BINARY uses Ross Data Compression which is more efficient for mostly numeric data sets. The shipped default is NO. Compression comes with a cost, and you need to consider the hit from compression on processing. There are times when you get notes in the log saying that compression increases the size of the data set (mostly when the data sets are small.) On the other hand, if you are dealing with billions of records such as Medicare claims, which often have a large amount of character data, compression can make an

enormous difference in data set size. Because of the potential for a hit on processing, this option may be restricted by your site SAS admins. For clinical trials work, where the data sets are small, COMPRESS=NO (the default) is appropriate. For claims work, COMPRESS=CHAR can make a substantial difference in storage and efficiency. For numeric data (such as banking or insurance) that is large to medium sized COMPRESS=BINARY could be utile, but compression in general should be avoided for smaller data sets.

Syntax: OPTIONS COMPRESS=NO;

SOURCE2 (NOTES, LIMITPROCNOTES)

For programmers who are using or reviewing programs that have one or more %include statements, the SOURCE2 system option is a lifesaver, providing the source code from the include file(s) in the log. The default is NOSOURCE2, which is appropriate in production – for example, a lengthy routine creating formats and style templates adds to the log. In development, seeing the log for the code that is spawning an error is essential.

The NOTES system option allows you to turn NOTES off in the log, reducing the burden of review in production. It is never appropriate to turn the NOTES system option off (NONOTES) in development and testing. LIMITPROCNOTES (new in M6) limits the number of notes that can be written to the SAS log by a SAS procedure. You can set a min (100 or more) and max (up to 2147483647) and specify note-count, so that you know how many notes were suppressed. LIMITPROCNOTES can be useful in that it does not "count" notes generated by titles and footnotes, because they are not created by procedures. I ran programs monthly to generate 15,000+ individual multi-page PDFs and the log had so many notes in it that it was almost impossible to review. Options NONOTES for the section generating the PDFs was a great space and time saver. (Hadden, "Programming the Provider Previews", 2012).

Syntax: OPTIONS SOURCE2;

Syntax: OPTIONS NONOTES;

Syntax: OPTIONS LIMITPROCNOTES

MFILE (MPRINT, MPRINTNEST, MLOGIC, MLOGICNEST)

Macro options are my very favorite. I could easily have chosen the top ten macro functions! The MFILE system option is used in conjunction with MPRINT and determines whether or not the log information saved with MPRINT is written to an external file for review. The default is NOMFILE, and MPRINT must be specified in the system options. Both MPRINT and MFILE can bog down processing, so it is appropriate for debugging purposes at the development and review stages, but not production. If you have not encountered MPRINTNEST yet, you are in for a treat. Macro nesting information (i.e., the call stack) is displayed in MPRINT output.

```
options MPRINT MPRINTNEST;
%macro furcoat;
  data _null_;
    %longunderwear;
  run;
%mend furcoat;
%macro longunderwear;
  put 'longunderwear macro';
%mend longunderwear;
%furcoat;
```

Program 2. Nested Macro Test

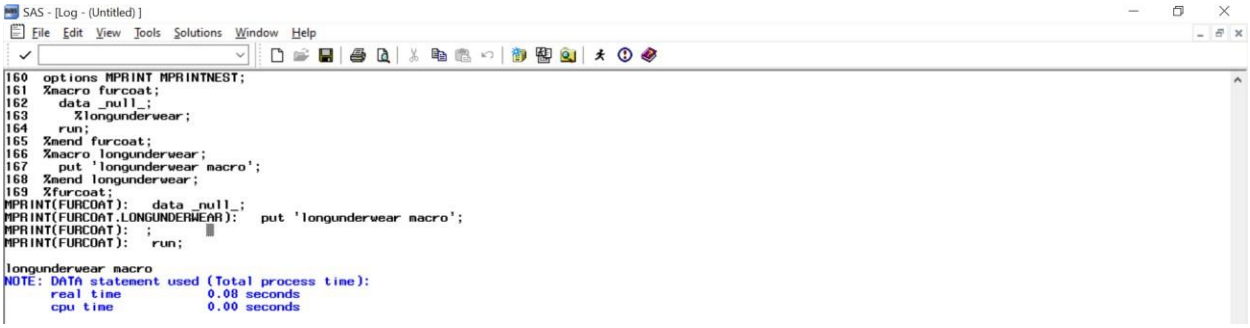


Figure 4. Log from Nested Macro Test

The MLOGIC and MLOGICNEST options work in a similar fashion. MLOGIC must be specified in the system options for MLOGICNEST to work. MLOGIC causes the macro processor to trace its execution and to write the trace information to the log, which is immensely helpful in debugging complex macros. The downside is the log becomes very verbose. MLOGICNEST goes a step further by enabling the macro nesting information to be displayed in the MLOGIC output in the log.

```
%macro furcoat;
     %put THIS IS FURCOAT;
     %sweatshirt;
%mend furcoat;
%macro sweatshirt;
     %put THIS IS SWEATSHIRT;
     %longunderwear;
%mend sweatshirt;
%macro longunderwear;
     %put THIS IS LONG UNDERWEAR;
%mend;
     options mlogic mlogicnest;
     %furcoat
```

Program 3. MLOGIC and MLOGICNEST test

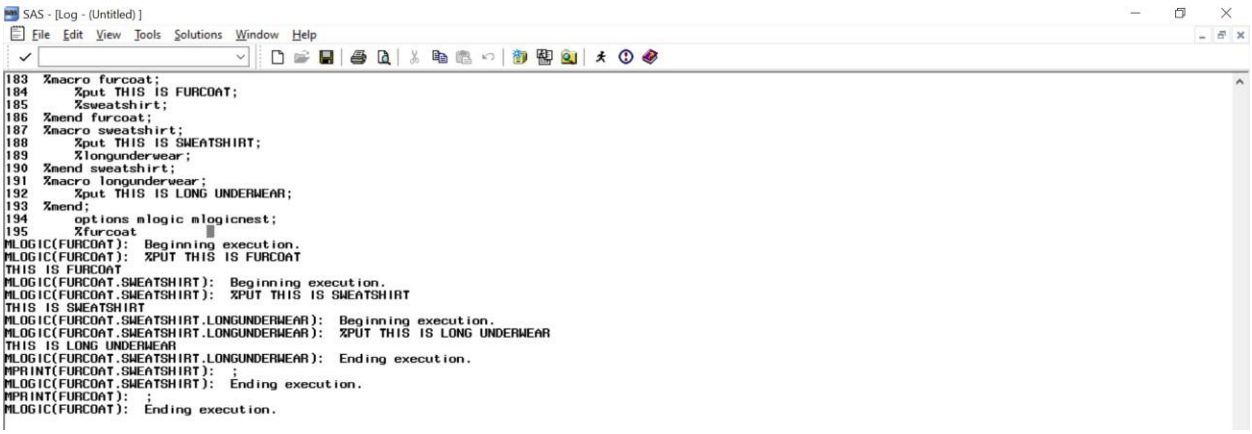


Figure 5. Log from MLOGIC and MLOGICNEST test

If we rerun the code above replacing MLOGICNEST with NOMLOGICTEST, we get the following output in the log:

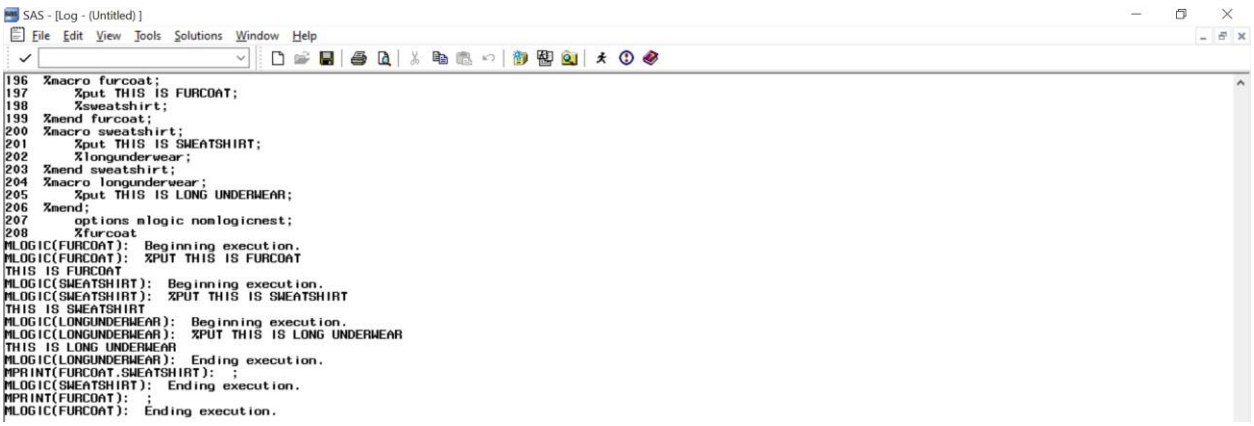


Figure 6. Log from MLOGIC and NOMLOGICNEST test

While these system options increase the size and complexity of the log, they are extremely worthwhile for debugging.

Syntax: OPTIONS MLOGIC MLOGICNEST;

ORIENTATION (NUMBER, DATE, LEFTMARGIN, RIGHTMARGIN, TOPMARGIN, BOTTOMMARGIN)

These system options are all a part of the ODSPRINT group and affect how ODS outputs are formatted. The default for output is to have the page NUMBER and DATE on the top right, the default for orientation is portrait, and the default margins vary with the destination and style template being used. You can override these defaults using the LEFTMARGIN, RIGHTMARGIN, TOPMARGIN, and BOTTOMMARGIN options. Many production reports have extremely strict reporting requirements as to margins, dates, and page numbers and it is important to be able to control these items in output. For the PDFs mentioned above, we needed to have narrow margins and control over where titles and footnotes were placed, as well as needing to switch orientation for wide tables within the PDF. All this could be accomplished with these system options. (Hadden, “Programming the Provider Previews”, 2012).

Long-Stay Quality Measures that are Included in the QM Rating

			[redacted]				AL	US
	2018Q1	2018Q2	2018Q3	2018Q4	4Q avg	Rating Points	4Q avg	4Q avg
MDS Long-Stay Measures								
Lower percentages are better.								
Percentage of residents experiencing one or more falls with major injury	2.2%	0.0%	2.3%	2.3%	1.7%	80	3.3%	3.4%
Percentage of residents who self-report moderate to severe pain[1]	14.9%	28.6%	19.0%	19.0%	20.5%	20	6.5%	7.0%
Percentage of high-risk residents with pressure sores[3]	2.8%	5.6%	0.0%	0.0%	2.2%	100	7.8%	7.4%
Percentage of residents with a urinary tract infection	10.9%	9.1%	4.7%	4.7%	7.4%	20	3.4%	2.8%
Percentage of residents with a catheter inserted and left in their bladder[1]	2.5%	2.2%	0.0%	0.0%	1.2%	80	2.5%	2.2%
Percentage of residents whose need for help with daily activities has increased	31.4%	27.6%	15.4%	15.4%	23.3%	15	12.9%	14.7%
Percentage of residents who received an antipsychotic medication	34.1%	30.2%	26.8%	26.8%	29.6%	15	19.9%	14.7%
Percentage of residents whose ability to move independently worsened[1]	27.4%	d<20	14.3%	14.3%	18.3%	75	14.9%	17.8%

[1]These measures are risk adjusted.
[2]This measure includes some imputed data because there are fewer than 20 resident assessments or stays across the four quarters. This value is used in calculating the QM points and used in the QM rating calculation but will not be displayed on Nursing Home Compare.

Figure 7. Image from PDF report

Syntax: ORIENTATION=LANDSCAPE;

Syntax: NODATE;

Syntax: NONUMBER;

Syntax: TOPMARGIN=.5in BOTTOMMARGIN=.5in LEFTMARGIN=.5in RIGHTMARGIN=.5in;

OPTLOAD AND OPTSAVE PROCEDURES

SAS provides two procedures for saving and loading specific sets of SAS System Options. This way, you can save production, validation, and debugging option sets and easily recall them for specific tasks.

```
proc optsave out=dd.options;
run;
```

Program 4. PROC OPTSAVE example

Figure 8. Log from PROC OPTSAVE procedure

You've Got Options

OPTNAME	OPTVALUE
ACCESSIBLECHECK	0
ACCESSIBLEGRAPH	0
ACCESSIBLEPDF	0
ACCESSIBLETABLE	0
ANIMATION	STOP
ANIMDURATION	MIN
ANIMLOOP	YES
ANIMOVERLAY	1
APPENDVARLENWARN	1
APPLETLOC	C:\Program Files\SASHome\SASGraphJavaApplets\9.4

Figure 9. Test Print from PROC OPTSAVE procedure

```
proc optload data=dd.options;
run;

options dlcreatedir;

proc options;
run;
```

Program 5. PROC OPTLOAD program

Figure 10. Log listing snippet from PROC OPTLOAD program

CONCLUSION

SAS provides a vast number of useful system options, as well as tools to exploit these resources. From descriptive tools to examples to tools to manage the saving and loading of system option settings on the fly, SAS system options are an integral part of the SAS programmer's toolbox.

REFERENCES

Cassidy, Deb. (2021). "Using Options to Improve Report Appearance". Proceedings of SAS Users Group International Conference, Chicago, Illinois. Retrieved from https://www.lexjansen.com/sugi/sugi21/iv/151-21.pdf

Fehd, Ronald. (2023). "A Configuration File Companion: testing and using environment variables and options; templates for startup-only options initstmt and termstmt". Proceedings of Pharmaceutical SAS Users Group Conference, San Francisco, California. Retrieved from https://www.lexjansen.com/pharmasug/2023/AP/PharmaSUG-2023-AP-009.pdf

Franklin, David. (2008). "Adding Control to Your Data Using SAS System and Dataset Options". Proceedings of Pharmaceutical SAS Users Group Conference, Atlanta, Georgia. Retrieved from https://www.lexjansen.com/pharmasug/2008/tt/TT03.pdf

Hadden, Louise S. (2024). "The Missing(ness) Piece: Building Comprehensive, Data Driven Missingness Reports and Codebooks Dynamically." *Proceedings of PharmaSUG 2024*, Baltimore, MD. Available at https://www.lexjansen.com/pharmasug/2024/DV/PharmaSUG-2024-DV-127.pdf

Hadden, Louise S. (2017). "Get Smart! Eliminate Kaos and Stay in Control – Creating a Complex Directory Structure with the DLCREATEDIR Statement, SAS® Macro Language, and Control Tables." *Proceedings of WUSS 2017*, Long Beach, CA. Available at https://www.lexjansen.com/wuss/2017/43_Final_Paper_PDF.pdf

Heaton, Edward. (2004). "SAS System Options are Your Friends". Proceedings of SouthEast SAS Users Group Conference, Nashville, Tennessee. Retrieved from https://analytics.ncsu.edu/sesug/2004/TU07-Heaton.pdf

Hadden, Louise S. (2018). "Programming the Provider Previews: Extreme SAS® Reporting." *Proceedings of SCSUG 2018*, Austin, TX. Available at https://www.lexjansen.com/scsug/2018/Hadden-SCSUG2018_Programming-the-Provider-Previews.pdf

Lafler, Kirk. (2019). "Saving and Restoring Startup (Initialized) SAS® System Options". Proceedings of Western Users of SAS Software Conference, Seattle, Washington. Retrieved from https://www.lexjansen.com/wuss/2019/25_Final_Paper_PDF.pdf

Lin, Wei, and Kang, Jiannan. (2009). "Don't Despair, You Do Have An Option!" Proceedings of NESUG 2009, Burlington, VT. Available at https://www.lexjansen.com/nesug/nesug09/po/PO03.pdf

Poll, Denise M. (2011). "SAS® Options: Versatile Players in the Game of SAS". Proceedings of SAS GLOBAL FORUM Conference, Las Vegas, Nevada. Retrieved from https://support.sas.com/resources/papers/proceedings11/240-2011.pdf

Russell, Kevin, and Tyndall, Russ. (2010). "SAS® System Options: The True Heroes of Macro Debugging". Proceedings of Northeast SAS Users Group Conference, Baltimore, Maryland. Retrieved from https://www.lexjansen.com/nesug/nesug10/ff/ff10.pdf

Thornton, Patrick, and Barbalau, Iuliana. (2012). "Working the System: Our Best SAS® Options." Proceedings of SAS Global Forum 2012, Orlando, FL. Available at https://support.sas.com/resources/papers/proceedings12/047-2012.pdf

Zender, Cynthia. (2007). "ODS Options and SAS® Stored Processes". Proceedings of SAS Global Forum Conference, Orlando, Florida. Retrieved from https://support.sas.com/resources/papers/proceedings/proceedings/forum2007/021-2007.pdf

ACKNOWLEDGMENTS

To my muses John King (aka Data _Null_), who reminded me years ago of the nesting data set option to debug data set loops, and Bart Jablonski (aka Yabwon), who got me onto the MPRINTNEST and MLOGICNEST options for debugging macro loops – many thanks for the information and the ongoing dialogue about the religion of SAS.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Louise S. Hadden
louisesquibbhadden@gmail.com