PharmaSUG CHINA

FREQUENTLY USED SAS OPTIONS IN CLINICAL TRIAL PROGRAMMING

Wanqing Li, Senior Programmer, Biostatistics, Amgen

Outline

- Overall insight on SAS Options

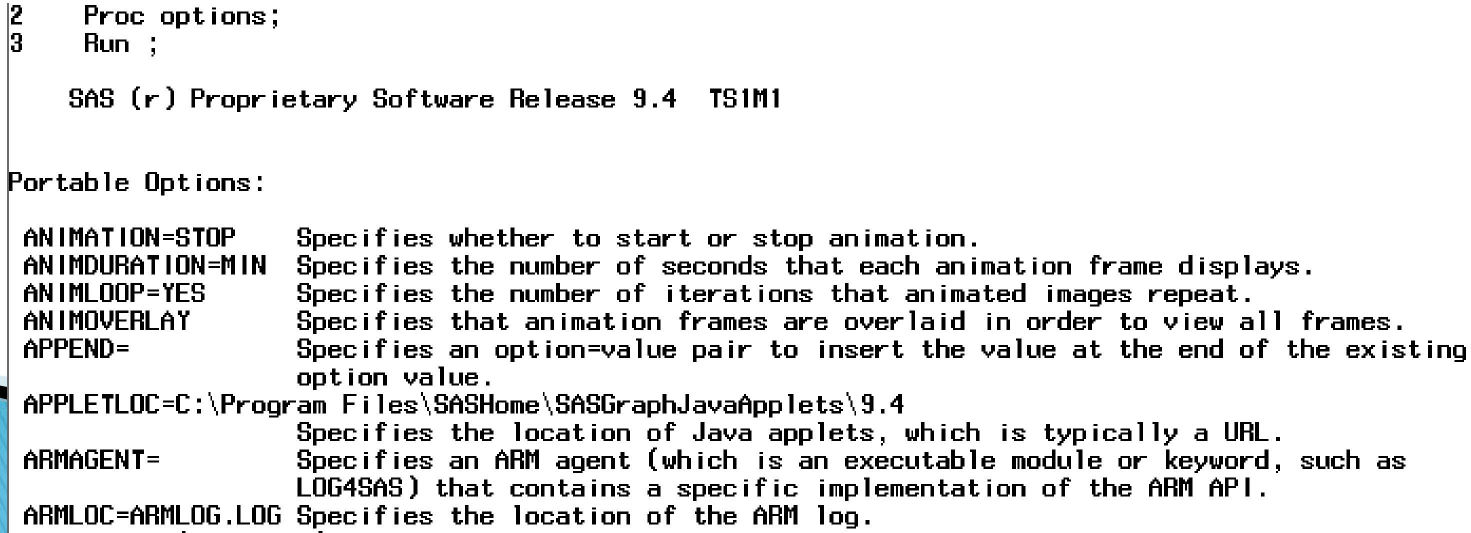
- Frequently used SAS options
  - Macro debug
  - Macro manage
  - Output control
  - Log control
  - X command

Overall insight on SAS Options

- System options are instructions that affect your SAS session. They control the way that SAS performs operations such as SAS System initialization, hardware and software interfacing, and the input, processing, and output of jobs and SAS files.
- 464 Options in SAS9.4
- 31 Groups
  - Listcontrol
  - Logcontrol
  - Macro
  - Performance
  - Communications
  - …

SAS Options introduction

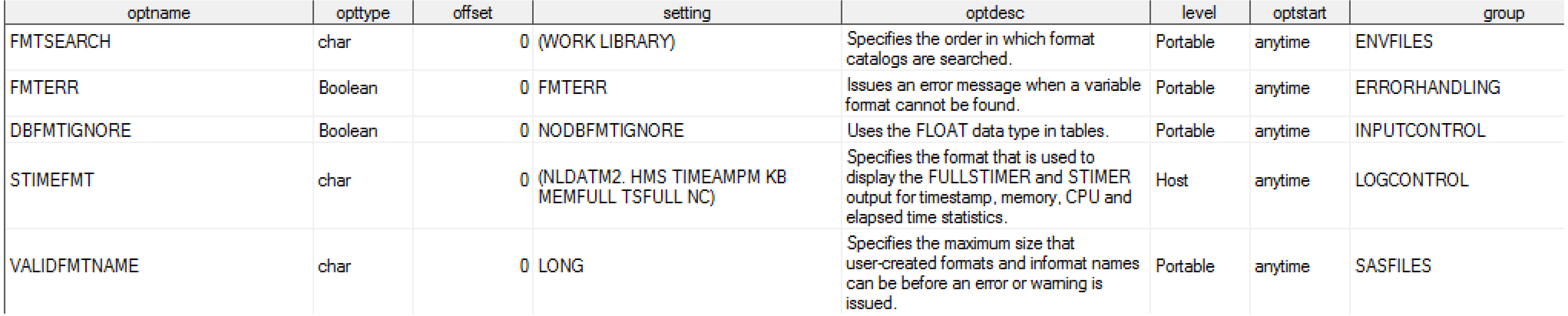
- Way to check SAS Options
  - OPTIONS Procedure.
    - Proc options;run;
      - The log will show all the options setting and description in current SAS environment.
    - Proc options group=macro;run;
      - The log only show macro related options setting and description in current SAS environment.



```
2     Proc options;
3     Run ;

    SAS (r) Proprietary Software Release 9.4  TS1M1


Portable Options:

 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.
 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.
```

SAS Options introduction

- Way to check SAS Options
  - Use VOPTION Dictionary table in SASHELP.
    - dataset VOPTION has variables of OPTNAME, OPTTYPE, OFFSET, SETTING, OPTDESC, LEVEL, OPTSTART, and GROUP to describe options in detail.



optname	opttype	offset	setting	optdesc	level	optstart	group
FMTSEARCH	char	0	(WORK LIBRARY)	Specifies the order in which format catalogs are searched.	Portable	anytime	ENVFILES
FMTERR	Boolean	0	FMTERR	Issues an error message when a variable format cannot be found.	Portable	anytime	ERRORHANDLING
DBFMTIGNORE	Boolean	0	NODBFMTIGNORE	Uses the FLOAT data type in tables.	Portable	anytime	INPUTCONTROL
STIMEFMT	char	0	(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.	Host	anytime	LOGCONTROL
VALIDFMTNAME	char	0	LONG	Specifies the maximum size that user-created formats and informat names can be before an error or warning is issued.	Portable	anytime	SASFILES

Macro debug options

- MPRINT – Displays the SAS statements that are generated by macro execution.
- MLOGIC – Traces macro execution and writes the results to the SAS log.
- SYMBOLGEN – Displays the results of resolving macro variable references in the SAS log.

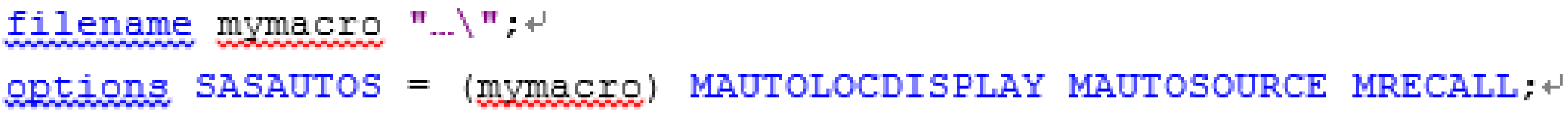
Macro manage options

- Simple way to call macro
  - %include “…\xxx.sas”;

- Use options to manage macros
  - SASAUTOS
  - SASMSTORE/ MSTORED

Macro manage options

- SASAUTOS – Specifies the location of one or more autocall libraries.
  - Firstly we can store the SAS code file in some location, then use below code to autocall the stored SAS code.



```
filename mymacro "…\";
options SASAUTOS = (mymacro) MAUTOLOCDISPLAY MAUTOSOURCE MRECALL;
```

MAUTOLOCDISPLAY – Enables the macro facility to display the autocall macro source location in the log when the autocall macro is invoked.

MAUTOSOURCE – Enables the macro autocall feature.

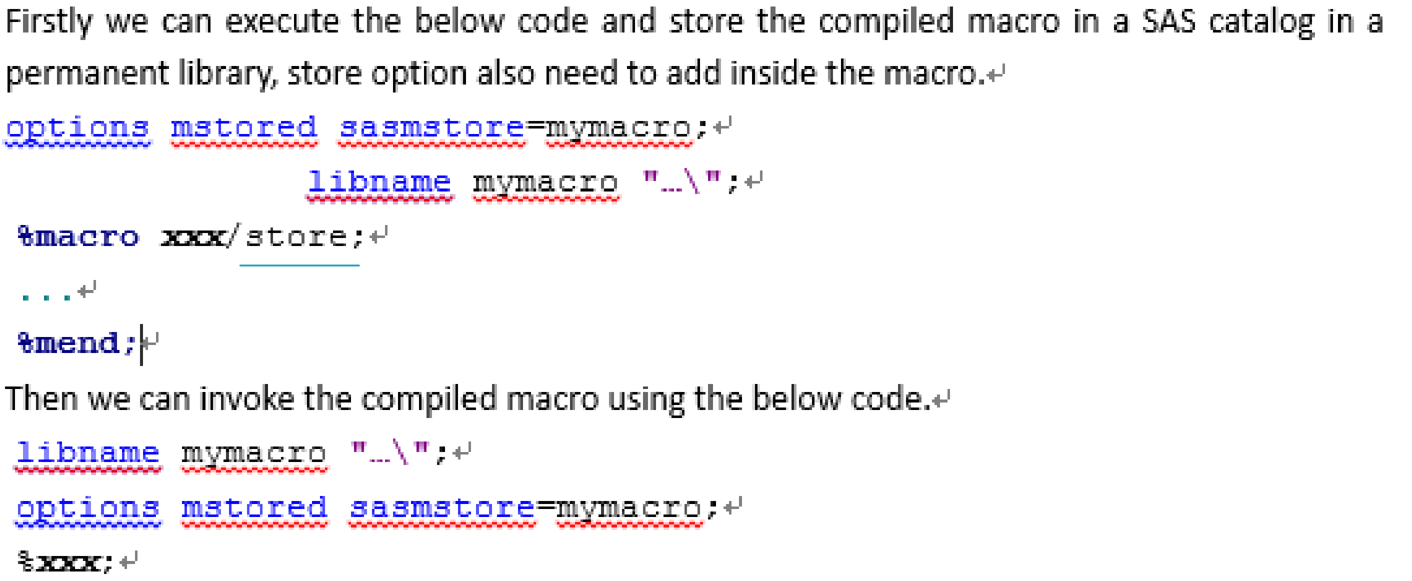
MRECALL – Searches autocall libraries for an undefined macro each time an attempt is made to invoke the macro.

Macro manage options

- SASAUTOS
  - In this way, we can invoke all the SAS files in the defined filename, don’t need to %include one by one.
  - Here I recommend to save these settings in AUTOEXEC.sas at the same location of sas.exe in your computer, then AUTOEXEC.sas will be executed automatically when SAS opened, and all of these macros can be directly invoked.

Macro manage options

- SASMSTORE – Specifies the libref of a SAS catalog for stored compiled SAS macros.
- MSTORED – Searches for stored compiled macros that are in the library specified by the SASMSTORE= option.



Firstly we can execute the below code and store the compiled macro in a SAS catalog in a permanent library, store option also need to add inside the macro.

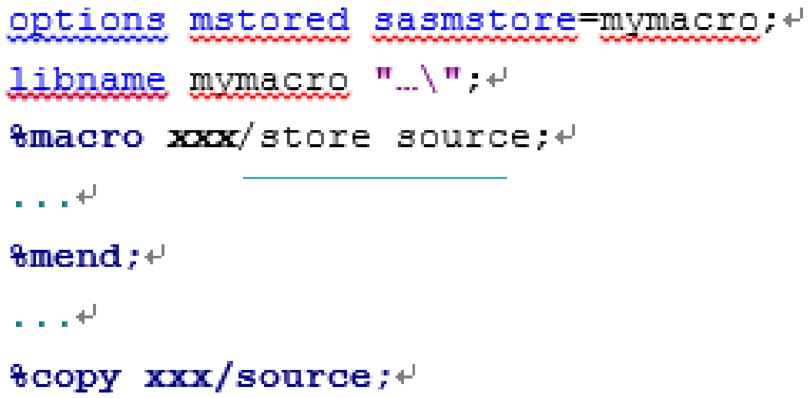
```
options mstored sasmstore=mymacro;
                    libname mymacro "…\";
 %macro xxx/store;
 ...
 %mend;
```

Then we can invoke the compiled macro using the below code.

```
 libname mymacro "…\";
 options mstored sasmstore=mymacro;
 %xxx;
```

Macro manage options

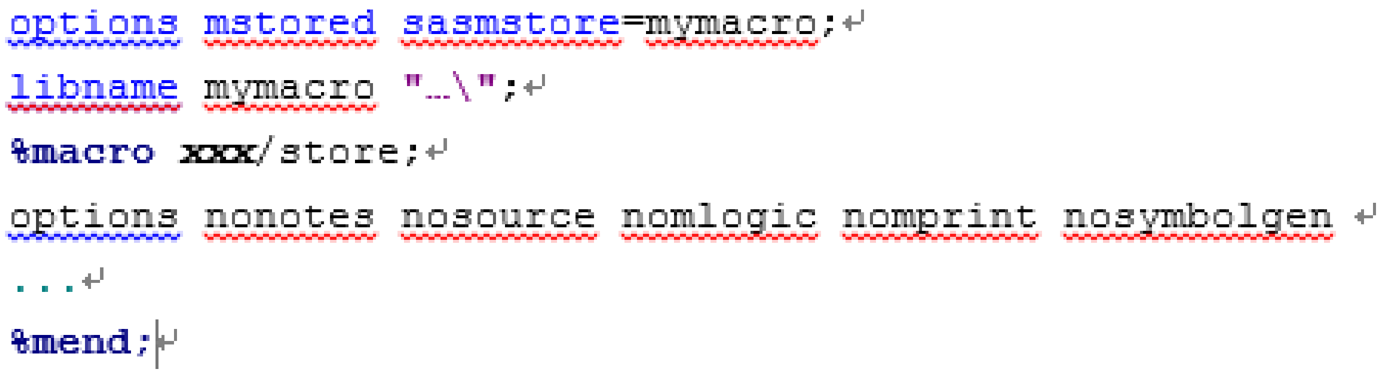
- SASMSTORE/MSTORED
  - In this way, you can’t see the source code when you use it, but if you set source option inside the macro like below when you store the macro, then you can use %copy xxx/source to show the source code.



```
options mstored sasmstore=mymacro;
libname mymacro "…\";
%macro xxx/store source;
...;
%mend;
...;
%copy xxx/source;
```

Macro manage options

- SASMSTORE/MSTORED
  - Also you can use options MLOGIC/SYMBOLGEN/MPRINT through log to check the code. But if you don’t want someone to check your source code when you store the compiled macro, you can use options nonotes nosource nomlogic nomprint nosymbolgen inside the macro like below, it is a way to protect your macro copyright.



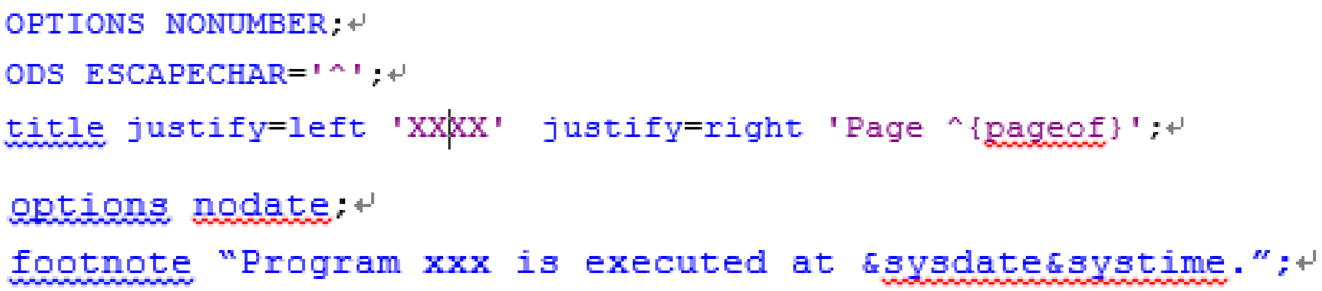
```
options mstored sasmstore=mymacro;
libname mymacro "…\";
%macro xxx/store;
options nonotes nosource nomlogic nomprint nosymbolgen
...
%mend;
```

Output control options

- MISSING – Specifies the character to print for missing numeric values.
  - Defaultly SAS output will show "." for the missing numeric values, so we need to use options missing=" " to make sure the output show it as " ".
- DATE – Prints the date and time that a SAS program started.
- NUMBER – Prints the page number on the first title line of each page of SAS output.
- BYLINE – Prints the BY line above each BY group.

Output control options

- DATE/NUMBER
  - The default data and number setting usually is not acceptable, we can change the option setting and combine other code to change the page number and datetime display style.



```
OPTIONS NONUMBER;
ODS ESCAPECHAR='^';
title justify=left 'XXXX'  justify=right 'Page ^{pageof}';

options nodate;
footnote “Program xxx is executed at &sysdate&systime.”;
```

Log Control and others options

- FMTERR – Issues an error message when a variable format cannot be found.
  - There will be an error and can’t successfully open SAS datasets when format is missing, because the default option is FMTERR, we can set it to NOFMTERR, then can check the datasets firstly when format is not available.

Log Control and others options

- FMTSEARCH – Specifies the order in which format catalogs are searched.
    - The FMT default search way is to search formats in work library, but sometimes the formats datasets is not in work library but in some other library, then we can set options fmtsearch=(fmtlib, work) to get the formats successfully.

Log Control and others options

- VARLENCHK – Specifies the type of message to write to the SAS log when the length of the variable that is being read is longer than the length that is defined for the variable.
- QUOTELENMAX – Writes a warning message to the SAS log if a quoted string exceeds the maximum length 262 allowed.
  - Sometimes we need to set it to VARLENCHK=NOWARN/NOQUOTELENMAX to avoid the warning message in log when some special situation.

Log Control and others options

- DMSLOGSIZE – Specifies the maximum number of rows that the SAS Log window can display.
- DMSOUTSIZE – Specifies the maximum number of rows that the SAS Output window can display.
  - By default, SAS only allows 99,999 lines in the log and output window, so we usually need to clear the window when it is full. We can set DMSLOGSIZE=999999 and DMSOUTSIZE=999999 to keep SAS from stalling, but still need to clear the window when lines >1million.

Log Control and others options

- COMPRESS – Specifies the type of compression to use for observations in output SAS data sets.
  - When the datasets is very large, we can set option compress=yes to decrease storage size and save a lot of executing time.

X command options

- XWAIT – The DOS shell closes and returns to SAS after EXIT is entered on the command line.
- XSYNC – Windows commands execute synchronously with SAS.
  - When the program executing need to interact with windows and use X command in SAS, it is better to set options NOXWAIT NOXSYNC.
  - Otherwise, it will stay in DOS interface and the following code will be not executed if you don't set NOXWAIT, the interaction between DOS and SAS will have a short delay when code executed if you don't set NOXSYNC.

Conclusion

- We can use flexible options to facilitate programming work and accomplish our intent.

- The summary is just some frequently used options, we can use proc options or VOPTION in SASHELP to check all the options setting and description and combine our sense to resolve some other questions.

Thanks