



PLUSPROJECT
PARTNERSHIP

EMPLOYEE
OWNED
BIostatISTICS
CRO



13-16 November
ICC Belfast

EU 2022

The Clinical Data
Science Conference

Standardised SAS Macro Considerations for Submission-ready Code

Will Greenway



BETTER BUSINESS TOGETHER

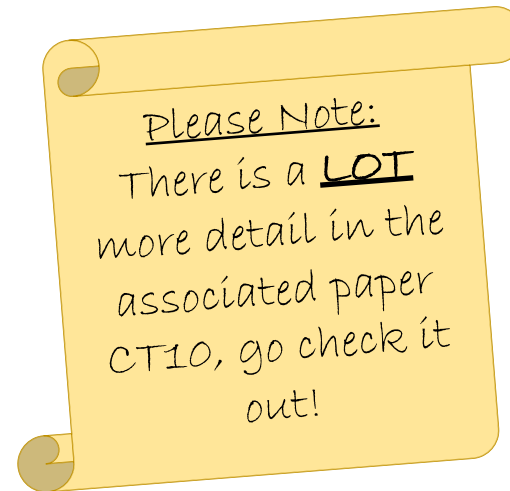
www.plus-project.co.uk

<https://www.linkedin.com/company/plus-project-ltd>



Agenda

- Submission-ready Code
- Creation
- Tidying Up
- SAS Options
 - Re-setting options
 - NOTES
 - MPRINT
- Commenting
- Debugging
- Conclusion



Submission-ready Code

“Sponsors should provide the software programs used to create all ADaM datasets and generate tables and figures associated with primary and secondary efficacy analyses.”

(FDA’s Study Data Technical Conformance Guide v4.9, March 2022)

“I used loads of macros but... I don’t want to share them”



Creation

Solution	Pro's	Con's
Submit macro code	✓ Source code, closest to the truth	✗ Often overly complex ✗ Difficult to understand ✗ Proprietary (large investment)
Submit macro catalogues	✓ Executable without giving out code	✗ Can't see code to understand it ✗ Transfer issues between systems common
Create "Submission-ready Code"	✓ Complex macro processing hidden ✓ Only shows lines executed ✓ Result is standalone executable code ✓ Transferable across systems	✗ Requires extra steps to produce ✗ Not ideal to do post-processing of created code

Creation - MFILE

4 STEP PROCESS:

1.

```
filename mprint "&progloc.\submission_ready_code\adsl_sas.txt";
```

2.

```
options mprint mfile;
```

3.

```
%macro adsl;  
    ... (Wrap entire program in a macro/series of macros, no "open code")  
%mend adsl;  
%adsl;
```

Code outside a macro
will not appear in
resolved code

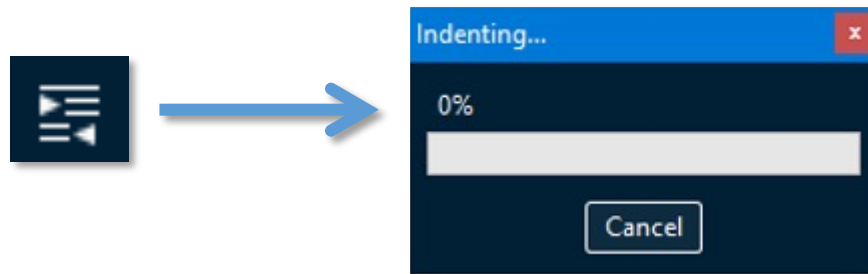
4.



Tidying Up

Using `mfile` / `mprint` by default everything aligns to the left...

- Add post-processing step to re-indent
- Quick solution is to use SAS® Enterprise Guide's "Format Code" option



Otherwise, everything else output **should** be fine...

SAS Options

Various SAS `options` may need to be changed multiple times
(why? I'll explain later...)



But doing so can impact other programs/later macros

Generally a macro should leave SAS `options` with the same settings that it found them. This means it is good practice to:

- **store** options settings at the start of a macro, and then
- **revert** them at the end to the same settings as at the start

SAS Options – Re-setting

Start of Program

```
%let optSettings=%sysfunc(getoption(notes)) %sysfunc(getoption(mprint));  
%let optLS=%sysfunc(getoption(ls,keyword));
```

NOTE: Some options like LS might need keyword to work correctly this way

End of Program (or part-way through if you want to toggle specific options)

```
options &optSettings;  
options &optLS;
```

E.G. May resolve to:

```
options NOTES NOMPRINT;  
options LS=200;
```

Alternative method using
proc optsave/optload
covered in associated paper
CT10

SAS Options – NOTES

- Set `options nonotes;` to hide unimportant messages from **SAS Log** e.g. internal macro processing
- Set `options notes;` to show users messages of interest in the **SAS Log**
- Combine with `%put` statements to improve output

```
%readxlsx(xlsxfile=&labloc.\lab.xlsx, outdset=lab);
```

SAS Log:

```
NOTE: [READXLSX] Reading: \\projects\project\study\data\labs\lab.xlsx.  
NOTE: [READXLSX] The data set WORK.LAB has 370 observations and 22 variables.
```

using `nonotes`
will not hide SAS
WARNINGS or
ERRORS!

(that's a good
thing by the
way...)

SAS Options – MPRINT

- Set `options nomprint;` to stop generating code into the referenced **mprint** file
- Set `options mprint;` to continue generating code into the referenced **mprint** file



Does every generated line really need to be output?

Are they all useful?

Will some cause confusion / wasted time trying to work out purpose?

Does it represent final processing, or help later macro processing (which will be already resolved in **mprint**)?

SAS Options – MPRINT

UPDATED

```
%macro sdtm2wrk;
```

```
proc sql noprint;
  select memname into :dset1-
  from dictionary.tables
  where libname='SDTM';
  %let lastsdtm=&sqllobs;
quit;
```

```
%do d=1 %to &lastsdtm;
  data &&dset&d;
    set sdtm.&&dset&d;
  run;
%end;
```

```
%mend sdtm2wrk;
%sdtm2wrk;
```

Original Code: Read in all SDTM datasets into WORK library

*ALL code is output to the referenced **mprint** file*

mprint:

```
proc sql noprint;
select memname into :dset1- from dictionary.tables where libname='SDTM';
quit;
```

```
data AE;
set sdtm.AE;
run;
data CM;
set sdtm.CM;
run;
...
```

But what is
this bit doing?!

Reviewer

Answer:
NOTHING (anymore)

```
%macro sdtm2wrk;
```

```
%let optmprint=%sysfunc(getoption(mprint));
options nomprint;
```

```
proc sql noprint;
  select memname into :dset1-
  from dictionary.tables
  where libname='SDTM';
  %let lastsdtm=&sqllobs;
quit;
```

```
options mprint;
```

```
%do d=1 %to &lastsdtm;
  data &&dset&d;
    set sdtm.&&dset&d;
  run;
%end;
```

```
options &optmprint;
```

```
%mend sdtm2wrk;
%sdtm2wrk;
```

*Creates macro variable for use in %do loop
Not important in resolved code*

***THIS** is the important bit, the actual reading in of datasets*

mprint:

```
options nomprint;
data AE;
set sdtm.AE;
run;
data CM;
set sdtm.CM;
run;
...
options MPRINT;
```

***Result:** Less code that needs to be reviewed; let's the reviewer concentrate on the lines that matter*

SAS Options – MPRINT



Where you add `nomprint` / `mprint` on the `options` line matters!

```
%macro mprint;  
  ** Toggle to hide processing **;  
  options nomprint nonotes;  
  
%mend mprint;  
%mprint;
```

mprint:

```
** Toggle to hide processing **;  
options nomprint nonotes
```

ERROR: Statement is not valid or
it is used out of proper order.

```
%macro mprint;  
  ** Toggle to show processing **;  
  options mprint notes source;  
  
%mend mprint;  
%mprint;
```

mprint:

```
source;
```

```
%macro mprint;  
  ** Toggle to hide processing **;  
  options nonotes nomprint;  
  
%mend mprint;  
%mprint;
```

mprint:

```
** Toggle to hide processing **;  
options nonotes nomprint;
```

NOTE: Statement is valid and
would correctly execute if run.

```
%macro mprint;  
  ** Toggle to show processing **;  
  options notes source mprint;  
  
%mend mprint;  
%mprint;
```

mprint:

Commenting

```
%macro comments;  
  
/* Block-comment */  
* Statement Comment;  
%* Macro comment;  
  
%mend comments;  
%comments;  
  
mprint:  
  
* Statement Comment;
```

Only Statement Comments made it through

P.S. We don't talk about ~~Block~~ COMMENT;

Comment Type	Commenting Macro Processing (e.g. %do loop or %if condition)	Commenting Resolved Code (code output to the mprint file)
/* Block-comment */	✓	✗
* Statement Comment;	✗	✓
%* Macro comment;	✓	✗

Debugging

Recommendations to help debugging:

- Check **pre-requisites**
*(You don't have to go over the top, just **key** checks)*
- Program proper **error handling** and clean-up
(Consider what might still be good to keep, e.g. interim datasets)
- Add a **debug macro parameter** and use this to:
 - Add extra **NOTE :** messages to help determine macro variable states
 - If debug mode active, prevent **nonotes/nomprint** deactivating during the code. Use macro variables to keep their state

Conclusion

Controlling what is displayed to the user in the log can help with debugging, and also avoid overloading them!

Controlling what is output to the `mprint` file can help with generating clean, easy to understand code

Say **NO** to random semi-colons

Say **NO** to code that would not actually run!

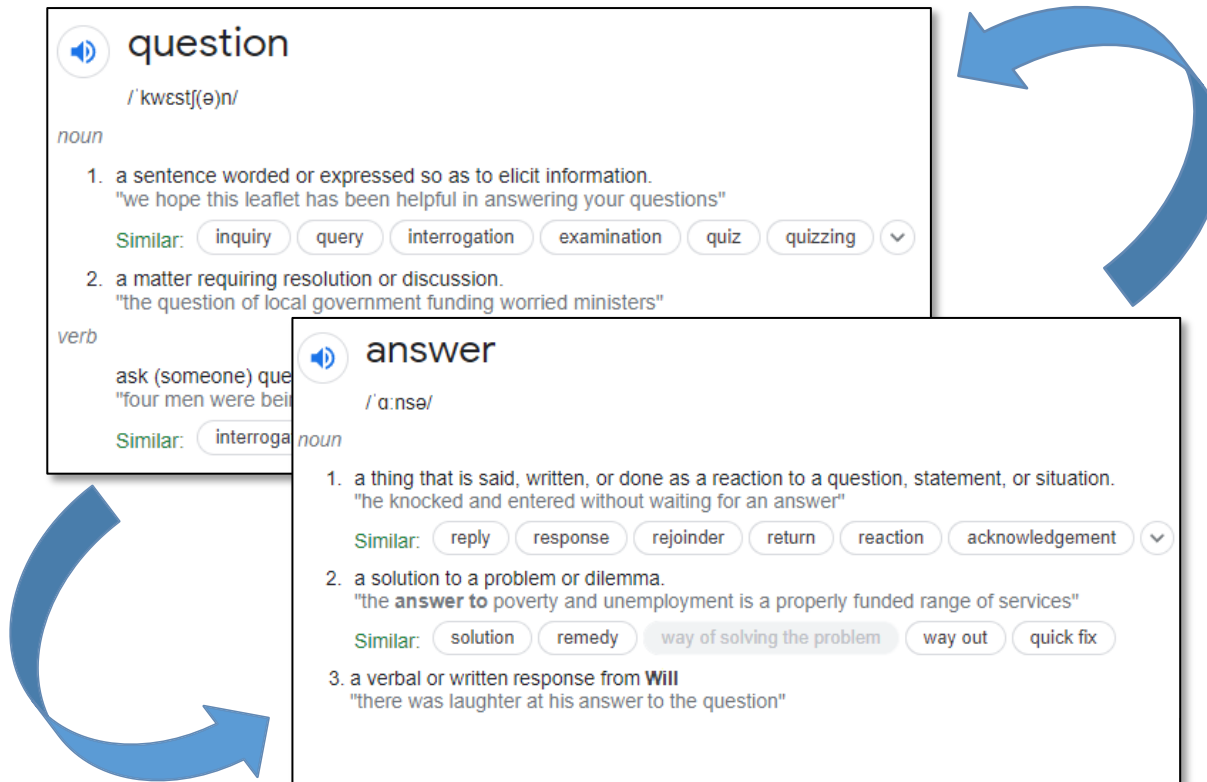
Say **NO** to meaningless comments (about macro processing...)

Say **NO** to including processing steps that have no significance

Say YES to clean, understandable, submission-ready code

Oh... and go read paper **CT10** before your next macro project! 

Questions?



question
/ˈkwɛstʃ(ə)n/
noun

1. a sentence worded or expressed so as to elicit information.
"we hope this leaflet has been helpful in answering your questions"

Similar: inquiry query interrogation examination quiz quizzing

verb

ask (someone) question
"four men were being asked questions"

Similar: interrogate

answer
/ˈɑːnsə/
noun

1. a thing that is said, written, or done as a reaction to a question, statement, or situation.
"he knocked and entered without waiting for an answer"

Similar: reply response rejoinder return reaction acknowledgement

2. a solution to a problem or dilemma.
"the **answer** to poverty and unemployment is a properly funded range of services"

Similar: solution remedy way of solving the problem way out quick fix

3. a verbal or written response from **Will**
"there was laughter at his answer to the question"



EMPLOYEE OWNED BIOSTATISTICS CRO

Will Greenway
Principal Programming Specialist

<https://www.linkedin.com/in/willgreenway>
will.greenway@plus-project.co.uk
www.plus-project.co.uk

MANCHESTER • EDINBURGH • LEEDS • WARSAW • KIEV