PhUSE EU Connect 2018

Paper SM05

A free tool to generate automatically user documentation for SAS macro libraries

Renos Kakoullis, ClinBAY, Limassol, Cyprus
Julien Vanwinsberghe, ClinBAY, Limassol, Cyprus
Loizos Nicolaou, ClinBAY, Limassol, Cyprus

ABSTRACT

When programming teams share SAS® macro libraries with little to no documentation, it is not uncommon for inexperienced programmers to experience difficulties in understanding and using the macros. Furthermore, as these shared macro libraries may be constantly updated and expanded, updating the accompanying documentation (if any) usually becomes an afterthought.

This paper presents a tool with the ability to automatically generate this macro user documentation from SAS macro code. The tool generates a convenient and straightforward document which includes a list of macros, input parameters, and a general description of the macros (written directly in the SAS code in the form of comments) in a specified macro library. The document can also include the full macro code in Annex with hyperlinks from the documentation.

From our experience, this is a very good method to maintain an up-to-date macro package documentation which is automatically generated entirely from the SAS programs.

INTRODUCTION

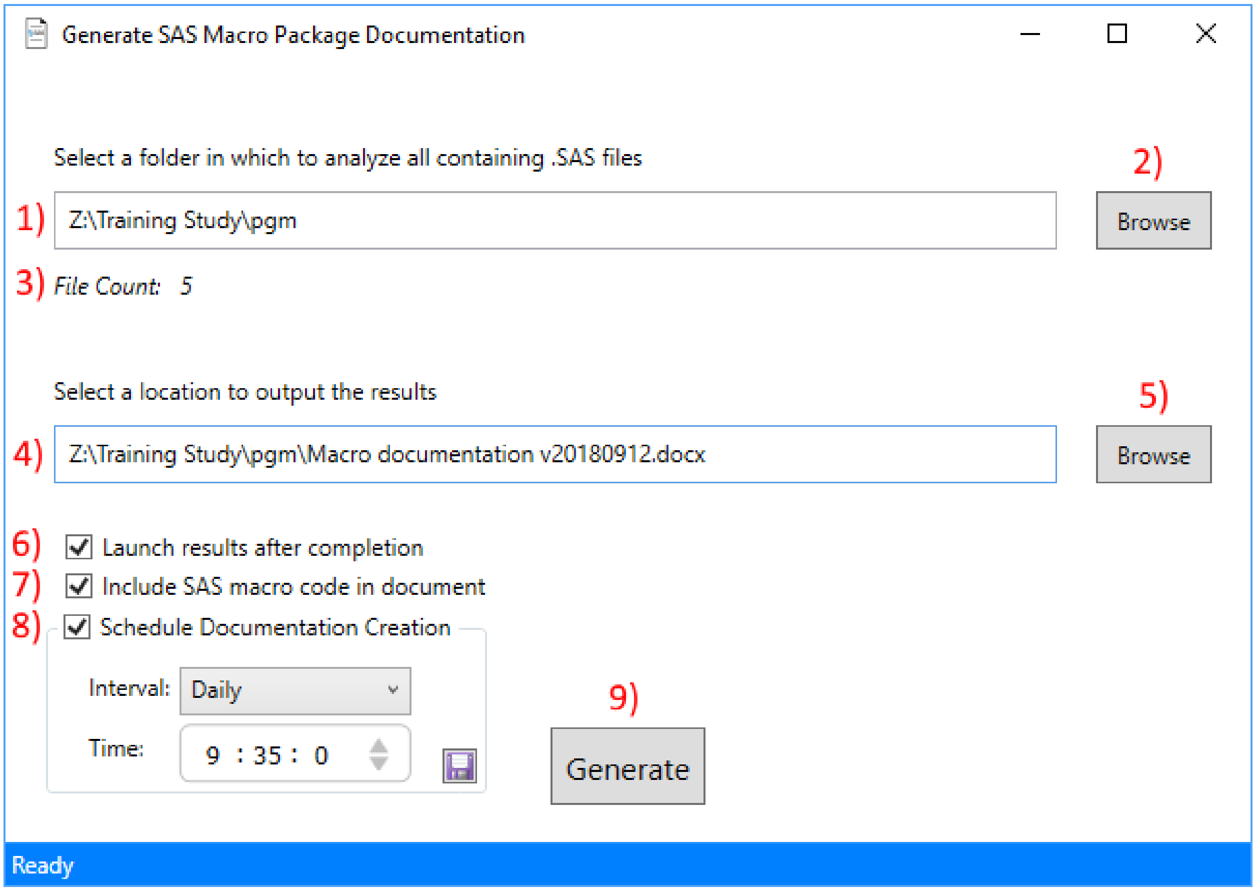
Owing to the need for an always up-to-date macro package documentation, ClinBAY has developed a free tool which meets our needs for automatically creating such documentation. This paper will discuss the technicalities of the tool, touching upon the user interface and how to use the tool, the ability to automatically update documentation using task scheduling, the ability to automatically import an introduction or user guide, the resulting documentation, as well as how the tool can be accessed.

CONCEPT

After experiencing the difficulties of having incomplete or outdated documentation for SAS macro libraries, ClinBAY decided to develop a tool to help mitigate these issues and allow their programming teams to work together more efficiently.

With SAS and .NET being considered as potential languages to build the tool, we concluded that .NET would be the language of choice for such a project as a result of its ease of use compared to SAS for developing the particular functionality required for this project. .NET would provide less complications in parsing of .sas files because of a range of built-in libraries, and as a result less code would be needed to analyze the contents of each .sas file. Furthermore, SAS does not possess the robustness of .NET for Microsoft Word automation which is necessary for producing a well formatted Microsoft Word documentation file with different fonts, styles, hyperlinks and bookmarks. SAS also contains no native ability to create scheduled tasks. Lastly, the user experience is of great importance, and using .NET would facilitate the ability to produce a simple yet powerful user interface for allowing the above mentioned features to be easily accessible. Ultimately we would have a more flexible tool which does not rely on the installation of SAS.

A GENERAL BRIEFING ON HOW THE TOOL WORKS


Generate SAS Macro Package Documentation

Select a folder in which to analyze all containing .SAS files

1) Z:\Training Study\pgm

3) *File Count:* *5*

Select a location to output the results

4) Z:\Training Study\pgm\Macro documentation v20180912.docx

6) ☑ Launch results after completion

7) ☑ Include SAS macro code in document

8) ☑ Schedule Documentation Creation

Interval: Daily

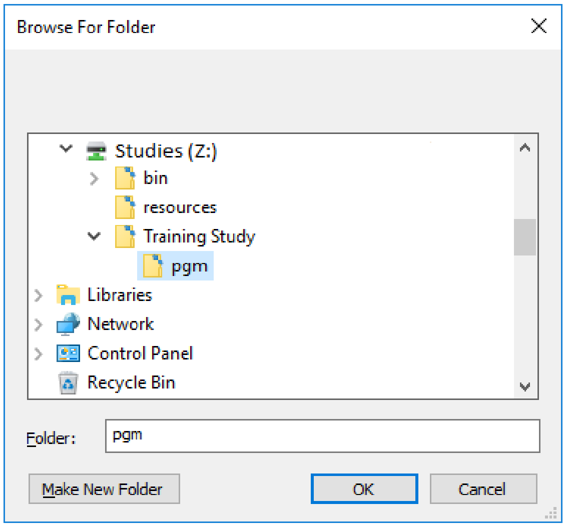
Time: 9 : 35 : 0

Ready

Figure 1: Screenshot of the tool main window.

The tool consists of a single window interface containing the following controls, as can be seen in Figure 1:

1. A text box displaying the selected folder.
2. A browse button to select a folder containing .sas files. Pressing this button will launch a folder browser where the user can navigate to any folder or paste a path directly. See Figure 2.

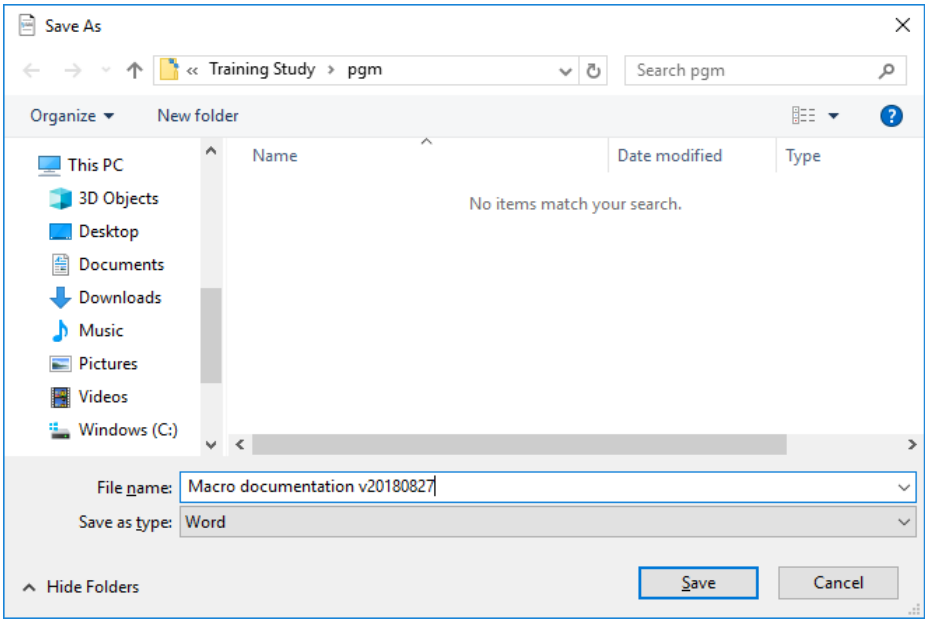

Browse For Folder

Folder: pgm

Figure 2: Screenshot of the folder selection dialog.

3. A label showing the count of .sas files in the chosen folder.
4. A text box displaying the selected output path.

5. A browse button to select the output location path. Pressing this button will launch a file-save dialog where the user can browse a location to save the file. A default filename is created using the format ‘Macro documentation vYYYYMMdd’, but this can of course be changed.



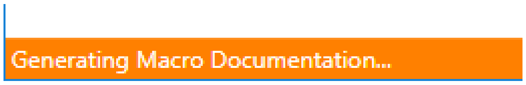
No items match your search.

File name: Macro documentation v20180827

Save as type: Word

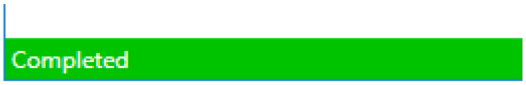
Figure 3: Screenshot of the file save dialog.

6. A checkbox allowing to launch the generated documentation file after completion.
7. A checkbox allowing to display the full SAS code of each macro within the generated documentation file. Choosing this option will create a ‘SAS Program Code’ section in the documentation with each macros full code.
8. A checkbox allowing to schedule a task, followed by task parameters Interval and Time, and a save button. Choosing this option will allow the user to schedule the creation of the documentation on a daily, weekly, or monthly basis. The time to perform the task can also be chosen. Doing so will ensure that the documentation is updated without any intervention by the user after initial setup. It is however required that the Windows user who configured the tool to perform the scheduled task be logged in during the scheduled time.
9. A Generate button. This button will only be enabled if a library and output folder have been selected. After clicking this button, the status of the procedure can be seen in the status bar at the bottom of the window.



Generating Macro Documentation...

Figure 4: Screenshot of the statusbar while generating documentation.

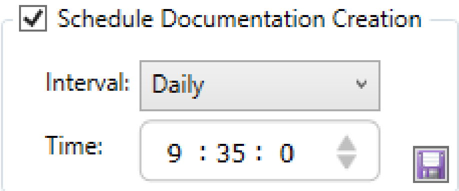


Completed

Figure 5: Screenshot of the statusbar after completing documentation.

TASK SCHEDULING FOR FULL AUTOMATION

A basic scheduling module is included in the tool, allowing for complete automation of creating and updating the SAS macro package documentation without user intervention. Figure 6 shows this module, which consists of a checkbox to enable/disable the feature, an interval drop-down list, a time selection box, and a save button.



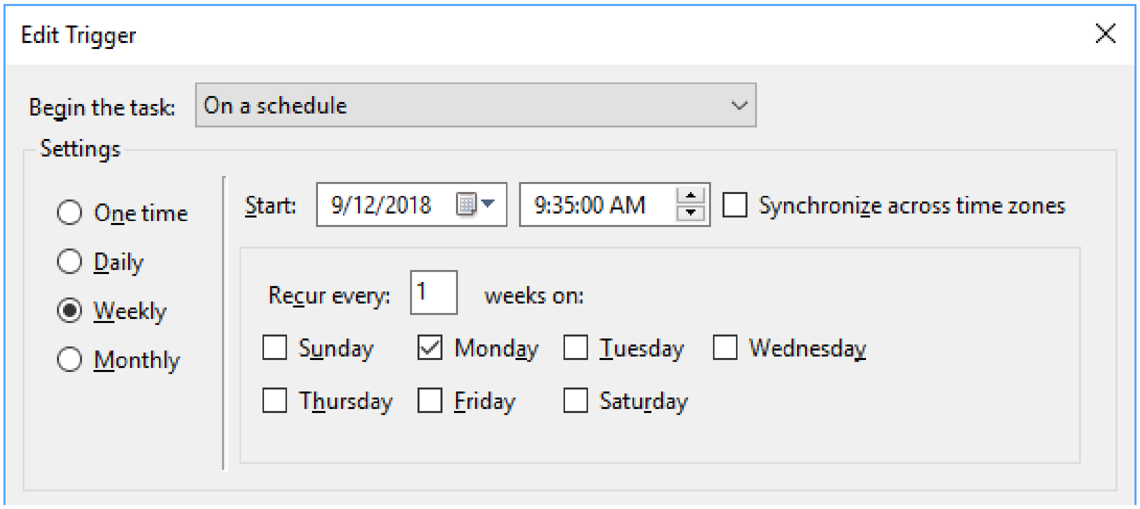
☑ Schedule Documentation Creation

Interval: Daily

Time: 9 : 35 : 0

Figure 6: Screenshot of the scheduling options.

The user can choose to perform the scheduled task on a daily, weekly, or monthly basis. Choosing to perform the task on a weekly basis will produce the document on each Monday, at the chosen time, while choosing to perform the task on a monthly basis will produce the document on the 1st of each month, at the chosen time. These parameters can be further fine-tuned in the Windows Task Scheduler with even more parameters, by locating the "UpdateSASDoc" task and editing the Trigger settings, as shown in Figure 7 and 8. Unchecking the 'Schedule Documentation Creation' checkbox will delete any task that was previously set.



Edit Trigger

Begin the task: On a schedule

Settings

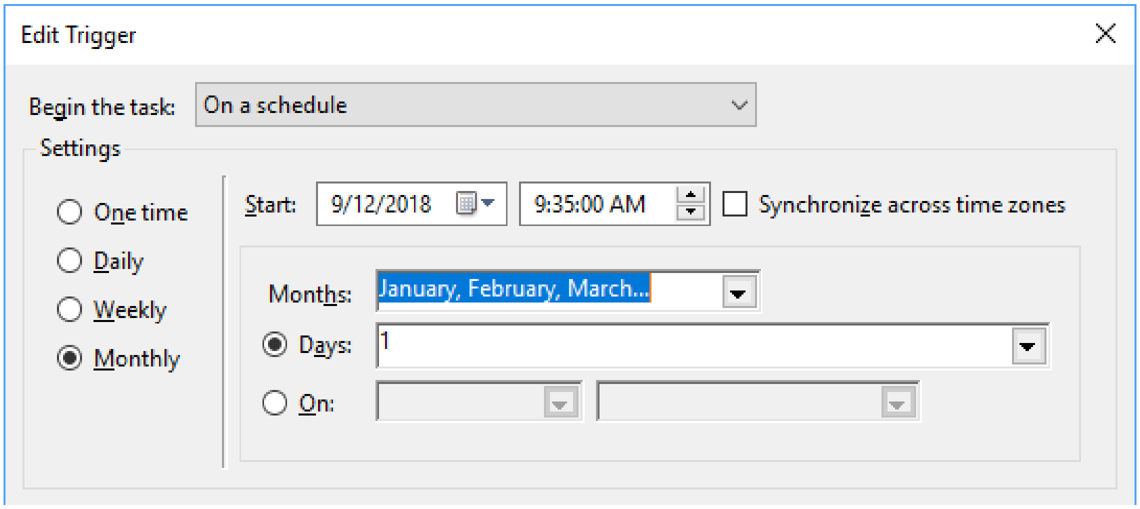
○ One time
○ Daily
● Weekly
○ Monthly

Start: 9/12/2018 9:35:00 AM □ Synchronize across time zones

Recur every: 1 weeks on:

□ Sunday ☑ Monday □ Tuesday □ Wednesday
□ Thursday □ Friday □ Saturday

Figure 7: Screenshot of the Trigger edit screen from the Windows Task Scheduler for a weekly task.



Edit Trigger

Begin the task: On a schedule

Settings

○ One time
○ Daily
○ Weekly
● Monthly

Start: 9/12/2018 9:35:00 AM □ Synchronize across time zones

Months: January, February, March...

● Days: 1

○ On:

Figure 8: Screenshot of the Trigger edit screen from the Windows Task Scheduler for a monthly task.

Note that enabling this feature will remove the date portion (e.g. v20180815) from the file name as the file will be replaced with each trigger of the task, rather than creating new files with different file name with each trigger.

If the ‘Schedule Documentation Creation” feature has been setup, the settings will be loaded each time the tool is launched.

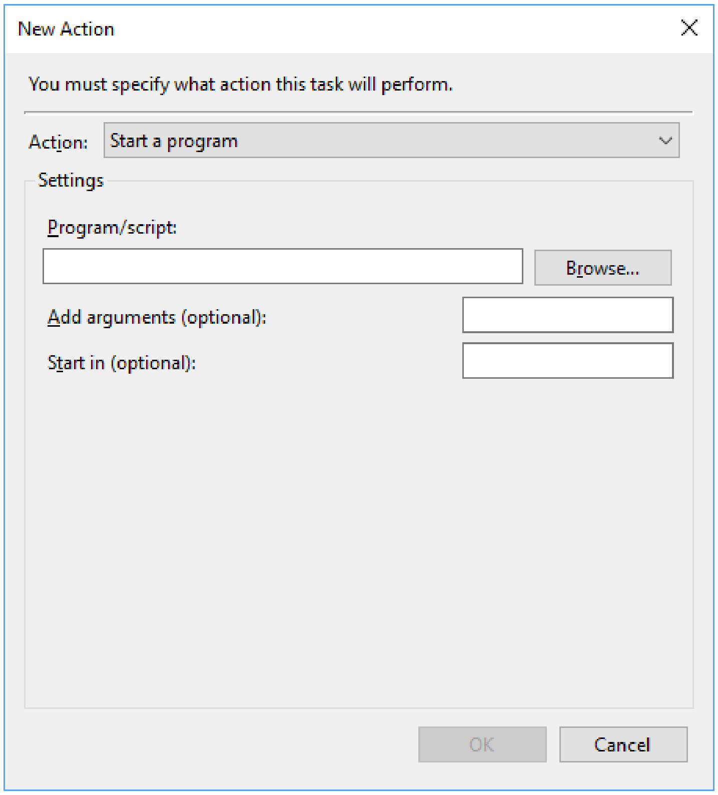
SCHEDULE MULTIPLE UPDATES WITH THE WINDOWS TASK SCHEDULER

If, and only if, the user wishes to schedule *multiple* macro SAS package documentation updates, this can be configured in the Windows Task Scheduler. This section is not required for the user if they use and update only one SAS macro library documentation as that is handled in the tool itself.

In the Windows Task Scheduler dialog, in order to create multiple tasks to update various macro package documentations, you can choose to create more Actions for the already existing UpdateSASDoc task as each Action correlates to one macro package. Each action is comprised of the following:

- A path to the application executable, e.g.
  C:\Program Files (x86)\ClinBAY\SASMacroDoc\SASMacroDoc.exe
- A string of arguments containing the macro package location, the output folder location, and a boolean value indicating whether to include the full macro code in the documentation, e.g.
  -"C:\Users\User\Desktop\Training study\pgm" -"C:\Users\User\Desktop\Macro documentation.docx" -True

Actions can be added to the existing UpdateSASDoc task from within the Windows Task Scheduler window by locating the “UpdateSASDoc” task and launching its properties. Select Actions and press New to add a new action. Refer to Figure 9 for the “New Action” window.



You must specify what action this task will perform.

Action: Start a program

Figure 9: Screenshot of the Windows Task Scheduler New Action window

To aid in this process, you may open the existing Action by clicking the action and pressing Edit. You can then copy its “Program/script” path, as well as the “Add arguments (optional)” string and paste them in the New Action window. Make sure to alter the library and output path in the arguments string for the new package.

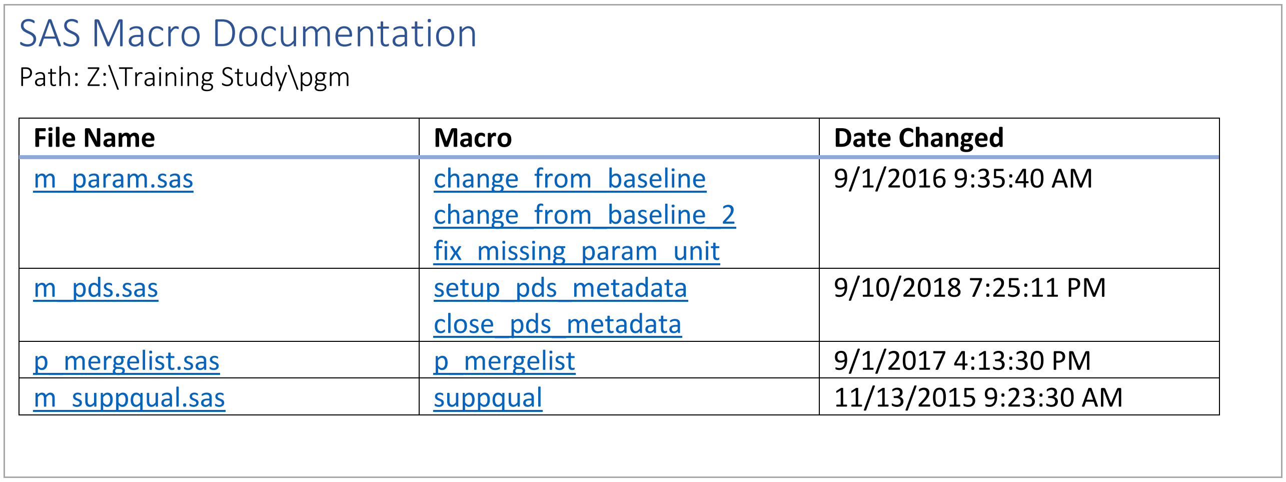
OUTPUT USER DOCUMENTATION

Following the operational steps above, a Microsoft Word .docx file is produced, structured such that it is easy for the user to see all contained macros in each .sas file, and to navigate to their macro defnintion lines or full macro code using bookmarks and hyperlinks.

The document contains two standard sections and potentially a further two optional ones. The first contains the full path to the library and a summary view table. The second section is optional and displays an introduction which could include information on how the macro package works, the best practices to use, what are the main macros, etc. The third section displays the macro definition line including input parameters along with the macro and input parameters description, and the final optional section displays the full SAS macro code for each macro. Below we describe the documentation in more detail, with sample hyperlinks to simulate navigation through the user guide.

MACRO SUMMARY

The Macro Summary section displays the path to the SAS library, followed by a table below it with File Name, Macro, and Date Changed columns. The rows of the File Name and Macro columns contains links which navigate to the Macro Information and SAS Program Code sections of the document respectively. As a result, this section of the document acts as the “Home”, allowing the user to navigate to all parts of the document easily. Figure 10 shows a sample Macro Summary, with live hyperlinks ‘m_param.sas’ and ‘change_from_baseline’ navigating to respective parts of this document as a demonstration of how the documentation functions.



SAS Macro Documentation

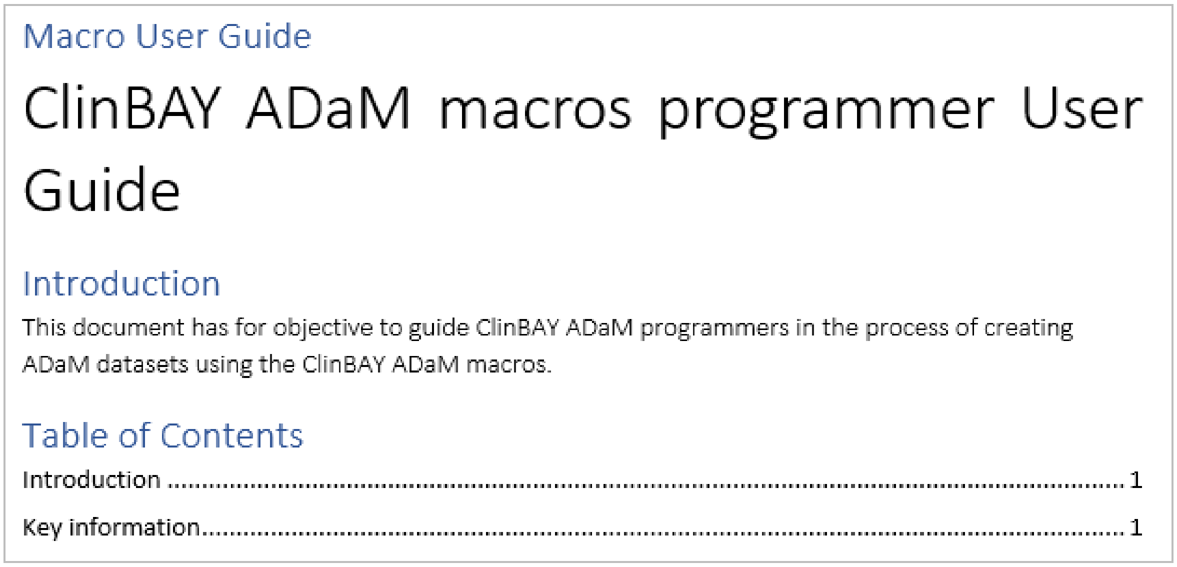
Path: Z:\Training Study\pgm

File Name	Macro	Date Changed
m_param.sas	change_from_baseline change_from_baseline_2 fix_missing_param_unit	9/1/2016 9:35:40 AM
m_pds.sas	setup_pds_metadata close_pds_metadata	9/10/2018 7:25:11 PM
p_mergelist.sas	p_mergelist	9/1/2017 4:13:30 PM
m_suppqual.sas	suppqual	11/13/2015 9:23:30 AM

Figure 10: Screenshot of the Macro Summary

MACRO PACKAGE INTRODUCTION / USER GUIDE

The Macro Package Introduction section can display information such as macro usage, best practices, and other information the user wishes to provide. The presence of this section is dependant on whether the user has placed a ‘macro_user_guide.docx’ document within the macro library folder, as this will be imported into the documentation after the above table of contents. It is important that the introduction file have the exact file name specified. The file is effectively copied and pasted within the documentation, as it retains all formatting such as titles, pictures, sections, links, etc. Omitting a ‘macro_user_guide.docx’ file from the SAS library folder, or having the file with a different name, will cause the tool to skip this section when generating the output. Refer to Figure 11 for a sample Macro User Guide.



Macro User Guide

ClinBAY ADaM macros programmer User Guide

Introduction

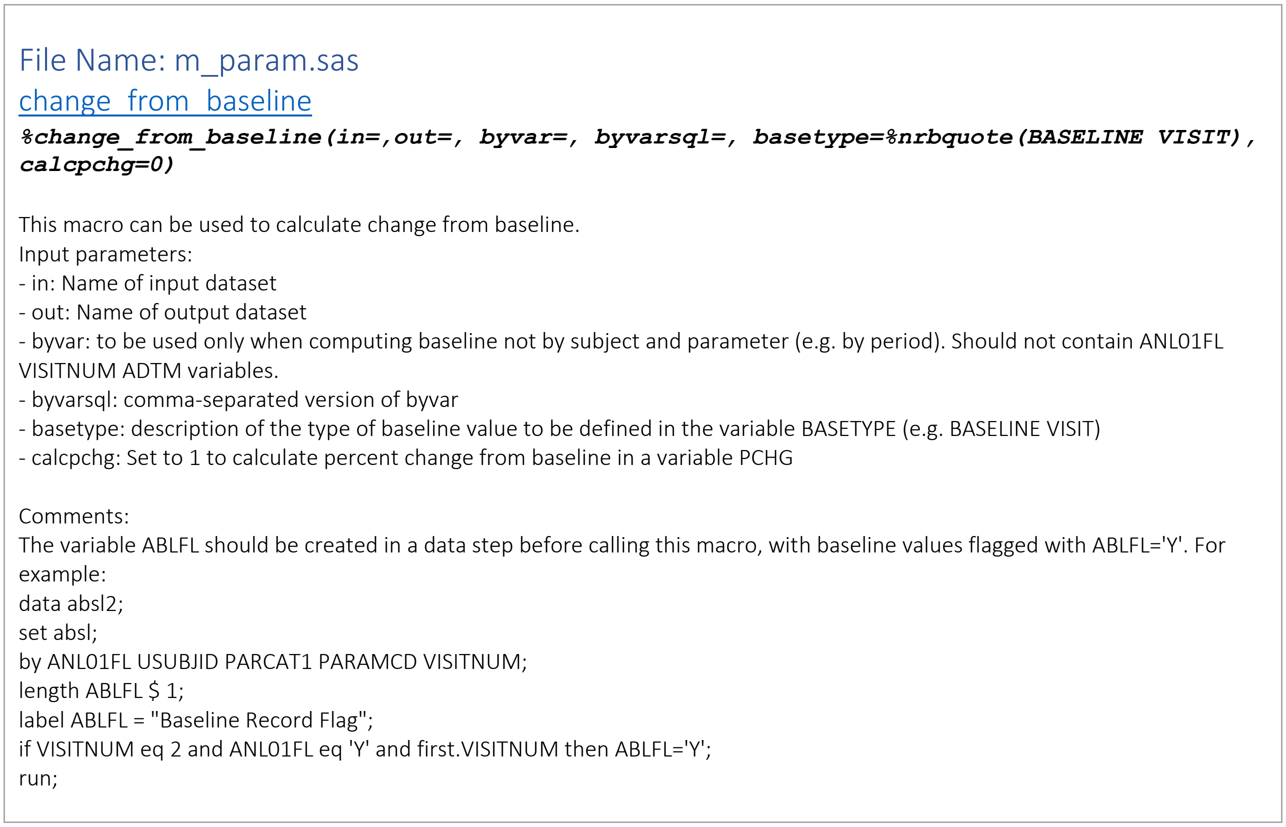
This document has for objective to guide ClinBAY ADaM programmers in the process of creating ADaM datasets using the ClinBAY ADaM macros.

Table of Contents

Figure 11: Screenshot of a sample of the imported Macro User Guide

MACRO INFORMATION

The Macro Information section shows each macro with the structure: File Name > Macro Name > Macro Definition > Macro Description / Input Parameter Description (if exists), as can be seen in Figure 12. The macro name can be clicked to navigate to the full macro code if 'Add program code to output' has been ticked on the main screen of the program before generating the document. This is demonstrated below by clicking the link 'change_from_baseline'.



File Name: m_param.sas

change_from_baseline

```
%change_from_baseline(in=,out=, byvar=, byvarsql=, basetype=%nrbquote(BASELINE VISIT), calcpchg=0)
```

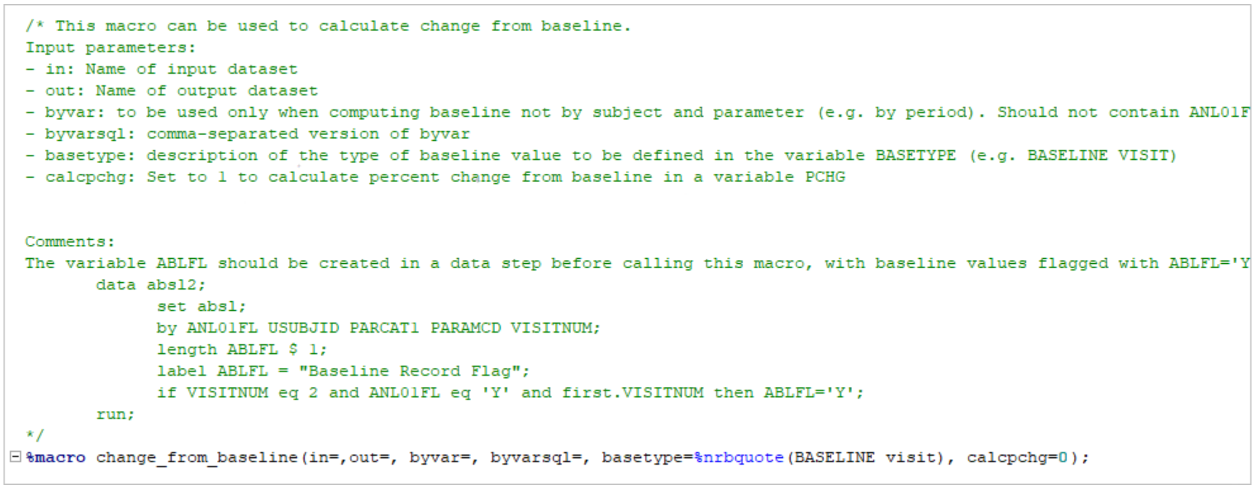
This macro can be used to calculate change from baseline.
Input parameters:
- in: Name of input dataset
- out: Name of output dataset
- byvar: to be used only when computing baseline not by subject and parameter (e.g. by period). Should not contain ANL01FL VISITNUM ADTM variables.
- byvarsql: comma-separated version of byvar
- basetype: description of the type of baseline value to be defined in the variable BASETYPE (e.g. BASELINE VISIT)
- calcpchg: Set to 1 to calculate percent change from baseline in a variable PCHG

Comments:
The variable ABLFL should be created in a data step before calling this macro, with baseline values flagged with ABLFL='Y'. For example:

```
data absl2;
set absl;
by ANL01FL USUBJID PARCAT1 PARAMCD VISITNUM;
length ABLFL $ 1;
label ABLFL = "Baseline Record Flag";
if VISITNUM eq 2 and ANL01FL eq 'Y' and first.VISITNUM then ABLFL='Y';
run;
```

Figure 12: Screenshot of a section of the Macro Information

The macro description is defined in the SAS program code as a comment placed directly above the macro definition using the comment syntax /* for opening comment tag and */ for closing comment tag. Input parameter descriptions can also be entered here in order to be shown in the documentation. Refer to Figure 13 for a sample macro definition and macro and input parameter descriptions in a .sas program.



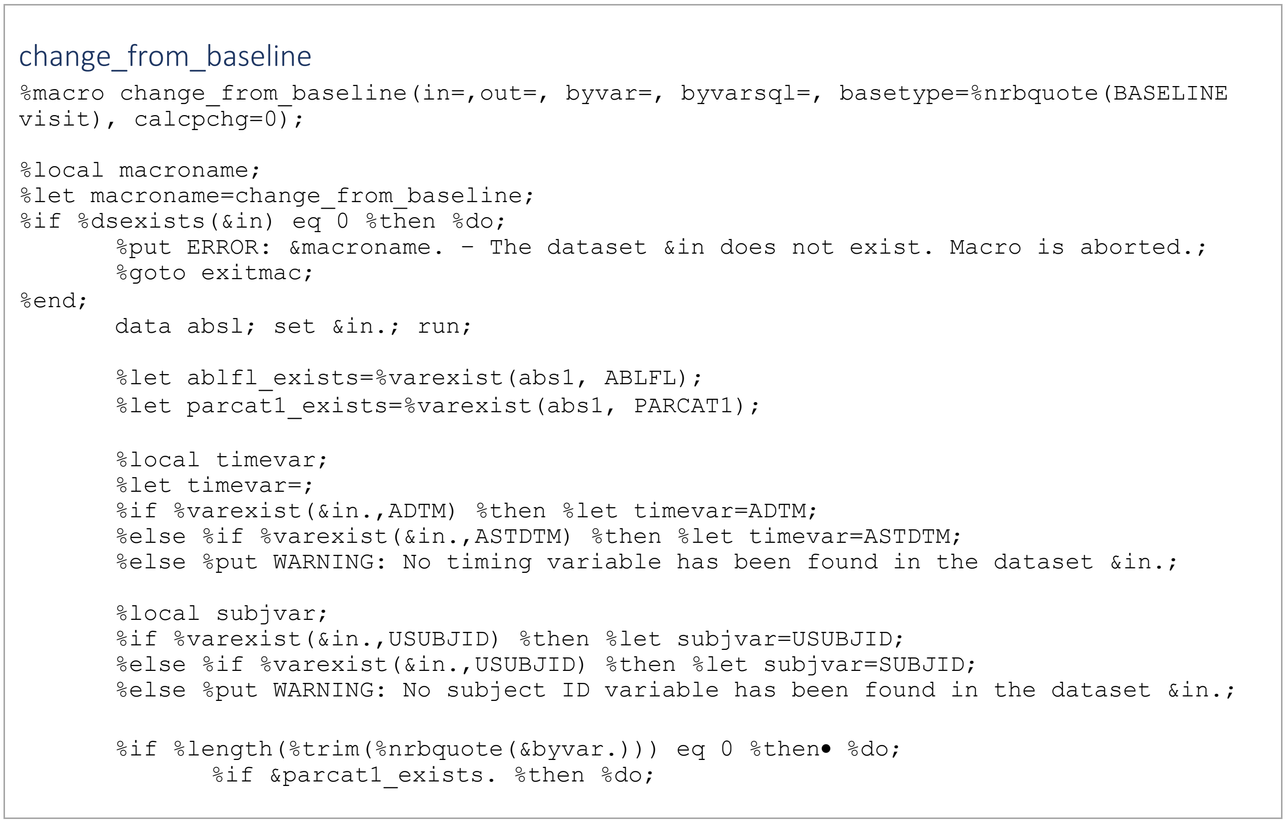
```
/* This macro can be used to calculate change from baseline.
Input parameters:
- in: Name of input dataset
- out: Name of output dataset
- byvar: to be used only when computing baseline not by subject and parameter (e.g. by period). Should not contain ANL01F
- byvarsql: comma-separated version of byvar
- basetype: description of the type of baseline value to be defined in the variable BASETYPE (e.g. BASELINE VISIT)
- calcpchg: Set to 1 to calculate percent change from baseline in a variable PCHG


Comments:
The variable ABLFL should be created in a data step before calling this macro, with baseline values flagged with ABLFL='Y
       data absl2;
             set absl;
             by ANL01FL USUBJID PARCAT1 PARAMCD VISITNUM;
             length ABLFL $ 1;
             label ABLFL = "Baseline Record Flag";
             if VISITNUM eq 2 and ANL01FL eq 'Y' and first.VISITNUM then ABLFL='Y';
       run;
*/
%macro change_from_baseline(in=,out=, byvar=, byvarsql=, basetype=%nrbquote(BASELINE visit), calcpchg=0);
```

Figure 13: Screenshot of sample Macro definition and Macro comment/description

SAS PROGRAM CODE

The SAS Program Code section shows each macros full code with the structure: File Name > Macro Name > Macro Code, as can be seen in Figure 14. This section is optional depending on the needs of the programming team. Also it may increase the size and length of the document significantly depending on the size of the SAS library.



change_from_baseline

```
%macro change_from_baseline(in=,out=, byvar=, byvarsql=, basetype=%nrbquote(BASELINE
visit), calcpchg=0);

%local macroname;
%let macroname=change_from_baseline;
%if %dsexists(&in) eq 0 %then %do;
       %put ERROR: &macroname. - The dataset &in does not exist. Macro is aborted.;
       %goto exitmac;
%end;
       data absl; set &in.; run;

       %let ablfl_exists=%varexist(abs1, ABLFL);
       %let parcat1_exists=%varexist(abs1, PARCAT1);

       %local timevar;
       %let timevar=;
       %if %varexist(&in.,ADTM) %then %let timevar=ADTM;
       %else %if %varexist(&in.,ASTDTM) %then %let timevar=ASTDTM;
       %else %put WARNING: No timing variable has been found in the dataset &in.;

       %local subjvar;
       %if %varexist(&in.,USUBJID) %then %let subjvar=USUBJID;
       %else %if %varexist(&in.,USUBJID) %then %let subjvar=SUBJID;
       %else %put WARNING: No subject ID variable has been found in the dataset &in.;

       %if %length(%trim(%nrbquote(&byvar.))) eq 0 %then• %do;
              %if &parcat1_exists. %then %do;
```

Figure 14: Screenshot of a section of the SAS Program Code

ACCESSING THE TOOL

The tool is completely free to download and can be accessed at www.clinbay.tk/macrodoctool/setup.html. A portable version of the tool is also available which is identical to the standard installed version in terms of use and functionality. It can be downloaded at www.clinbay.tk/macrodoctool/portable.html

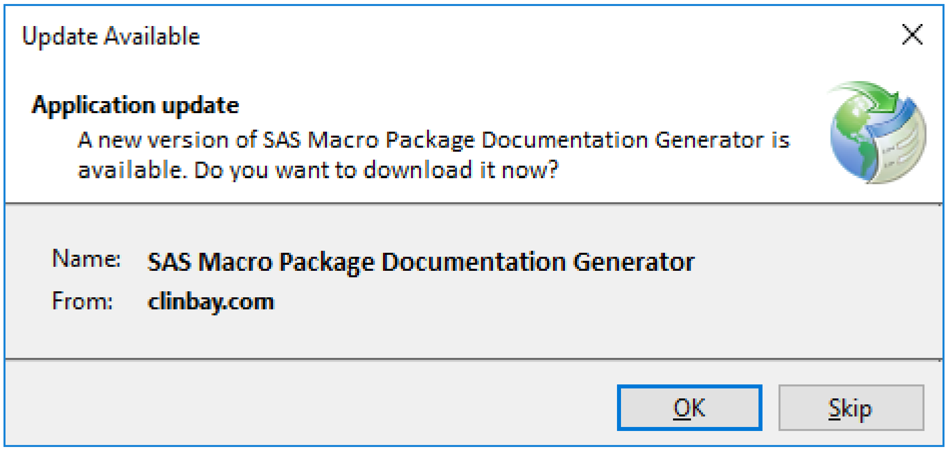
TECHNICAL REQUIREMENTS

- Windows XP and later
- .NET Framework 4.0 Client Profile and newer

Note that administrator privileges will only be needed if the installer detects that .NET Framework 4.0 client profile is missing from the computer. Otherwise administrator privileges are not needed for installation, use, and updating of the program. Also be aware that SAS and Microsoft Word are not required for functioning of the program.

UPDATES

The program contains an auto-update feature which allows the user to choose whether to perform an update when one becomes available. If chosen, the update will download and be applied after restarting the program.



Update Available

Application update

A new version of SAS Macro Package Documentation Generator is available. Do you want to download it now?

Name: **SAS Macro Package Documentation Generator**

From: **clinbay.com**

Figure 15: Screenshot of the auto-update window.

CONCLUSION

This paper shows how the free tool to generate SAS macro documentation can be beneficial to SAS programmers by improving the workflow between teams. By automating the generation of macro documentation, it both ensures an up-to-date reference for programmers to access, and saves time and effort in manually updating the documentation which can become increasingly difficult the larger the SAS macro library becomes.

The tool can be downloaded for free at www.clinbay.tk/macrodoctool/setup.html as an installed program, and at www.clinbay.tk/macrodoctool/portable.html as a portable program. For technical support you can contact ClinBAY at support@clinbay.com.

CONTACT INFORMATION

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

Renos Kakoullis
ClinBAY
182 Agias Fylaxeos street, Office 101, Kofteros Business Center
3083, Limassol, Cyprus
Email: renos@clinbay.com
Web: www.clinbay.com

Julien Vanwinsberghe
ClinBAY
182 Agias Fylaxeos street, Office 101, Kofteros Business Center
3083, Limassol, Cyprus
Work Phone: +32-67 70 90 84
Email: julien@clinbay.com
Web: www.clinbay.com

Loizos Nicolaou
ClinBAY
182 Agias Fylaxeos street, Office 101, Kofteros Business Center
3083, Limassol, Cyprus
Work Phone: +32-67 70 90 80
Email: loizos@clinbay.com
Web: www.clinbay.com