Paper 2564-2018

SAS Windowing environment – Tips and Tricks

Ravi Venkata and Mahesh Minnakanti, The EMMES Corporation;

ABSTRACT

If you work with SAS, then you probably find yourself repeating steps and tasks when developing a program. SAS windowing environment provides customizable toolbar and other options to help reduce time taken to perform few repetitive tasks. In this paper, we will discuss how to utilize the SAS windowing environment to perform a recurring task that's a click or a word away and will provide few tips and tricks to help increase productivity. This paper is targeted at new SAS users who might not be aware of all the wonderful options that SAS windowing environment options available to help increase productivity.

INTRODUCTION

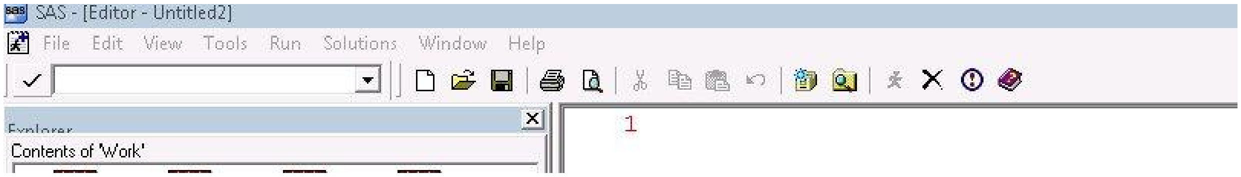
When you develop a huge SAS macro or write a huge code to accomplish a task and submit it with SAS system options like MLOGIC MPRINT. Inevitably, the log output will be huge - you might have created many macro variables, you might want to check the macro variable values or delete all the macro variables or just few of them, it might have errors and you might have to clear the work library, clear the log and re-run the code. To perform all the above tasks, on my pc - I just click few buttons. SAS windowing environment provides us an easy way to perform few of the repetitive tasks with– Just a click or a word! This paper focuses on the way to do that with few examples, discusses the ways to share your taskbar buttons and shortcuts. There are also few additional tips which might be helpful.

CLICKS AND KEYS MAKE PROGRAMMING EASY

THE CLICKS

This section discusses the methods to customize SAS toolbar on the main window to make repetitive tasks easier. SAS procedures, data steps, programs can be added to a toolbar button using the SAS function "**GSUBMIT**".

SAS main window looks like below prior to any customizations to it.



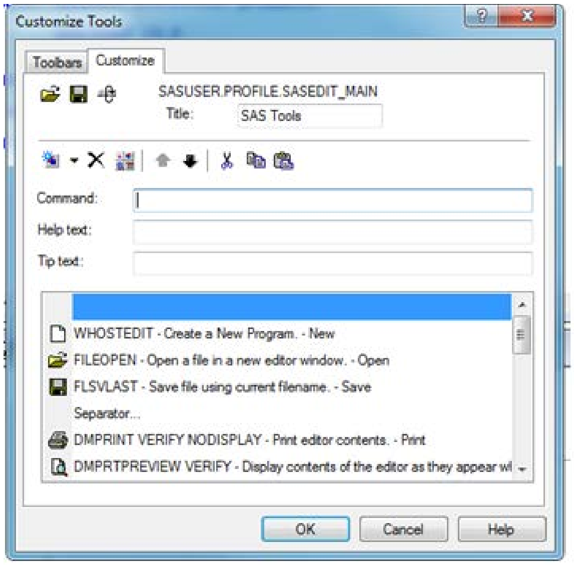
Display 1: SAS Window prior to Customization

We first look at the generic method to customize the toolbar on the SAS main window and then we look specific examples which you will find very useful in day to day programming activities. Toolbar buttons can be added specific to program editor window or specific to log window. There might be instances where adding a button specific to log window useful E.g.: Finding errors in the log.

To add a toolbar button, we do the following,

i. Make the program editor or log window active by clicking on it.

ii. Navigate to Tools>Customize and click Customize tab.



SASUSER.PROFILE.SASEDIT_MAIN

Title: SAS Tools

Command:

Help text:

Tip text:

WHOSTEDIT - Create a New Program. - New

FILEOPEN - Open a file in a new editor window. - Open

FLSVLAST - Save file using current filename. - Save

Separator...

DMPRINT VERIFY NODISPLAY - Print editor contents. - Print

DMPRTPREVIEW VERIFY - Display contents of the editor as they appear wi

Display 2: Customize Toolbar Window

iii. Click the button to get a new field and enter the command, the optional Help text, Tip text and click Save button. Click the button to assign an icon from the list shown.

Hit the arrows Down and move it after one of items in the list named SEPARATOR**,** so, that it appears on the toolbar.

Let's look at the important fields from Display 2,

Command is where you enter your SAS commands with semicolon to perform a task. The task may be a SAS command, SAS procedure or a path to SAS program.

Tip Text displays the ScreenTip that appears under the button when the pointer is hovered over the button in the toolbar. The field is very useful to a which button does what. This field can be blank.

Help Text displays the Help text that appears in the status bar message area when the pointer is placed over the button in the toolbar. This field can be blank.

Handy Buttons to Add

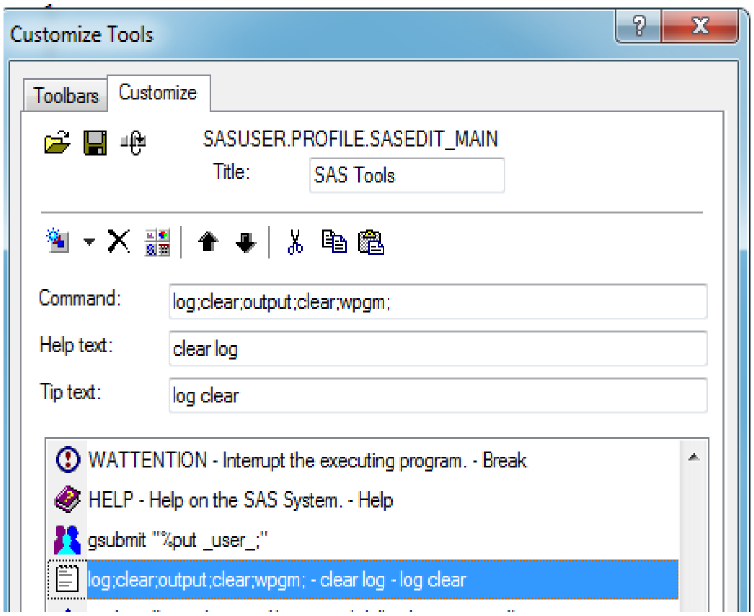
Let's look at few specific examples which we think will be helpful to SAS users. We follow the procedure described above in all the examples that follow.

Example 1: Clearing the Log

Instead of the default shortcut key, we can add the command "log;clear;output;clear;wpgm;" without quotes in the command field. It clears the log window and output window with just a click. Make sure to make the program editor window active before adding this button.

In the Help text and Tip text fields, we entered optional text clear log and log clear which could have been left blank which you'll see in other examples below.

Display 3 shows the command and other optional fields. The highlighted also shows the icon assigned to this button added after clicking button.



SASUSER.PROFILE.SASEDIT_MAIN

Title: SAS Tools

Command: log;clear;output;clear;wpgm;

Help text: clear log

Tip text: log clear

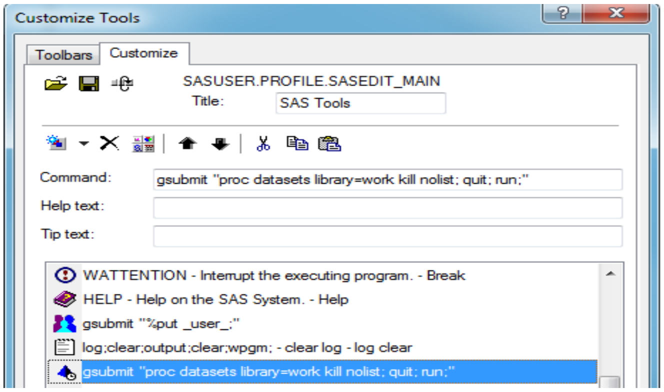
- WATTENTION - Interrupt the executing program. - Break
- HELP - Help on the SAS System. - Help
- gsubmit "%put _user_;"
- log;clear;output;clear;wpgm; - clear log - log clear

Display 3: Clear the log

Example 2: Clearing the Work Directory

The most common way to clear a library is by using PROC DATASETS procedure with KILL option. We can add such SAS procedures to a button so that it will perform the same task without typing it again. We use the SAS command GSUBMIT to pass the command. In the command field type the following: - gsubmit" Proc datasets library=work kill nolist;quit;run;"

Display 4 shows the command and other optional fields. If you observe below, help text and Tip text fields are empty but an icon has been assigned. Make sure to make the program editor window active before adding this button.



SASUSER.PROFILE.SASEDIT_MAIN

Title: SAS Tools

Command: gsubmit "proc datasets library=work kill nolist; quit; run;"

Help text:

Tip text:

- WATTENTION - Interrupt the executing program. - Break
- HELP - Help on the SAS System. - Help
- gsubmit "%put _user_;"
- log;clear;output;clear;wpgm; - clear log - log clear
- gsubmit "proc datasets library=work kill nolist; quit; run;"

Display 4: Clearing the Work Directory

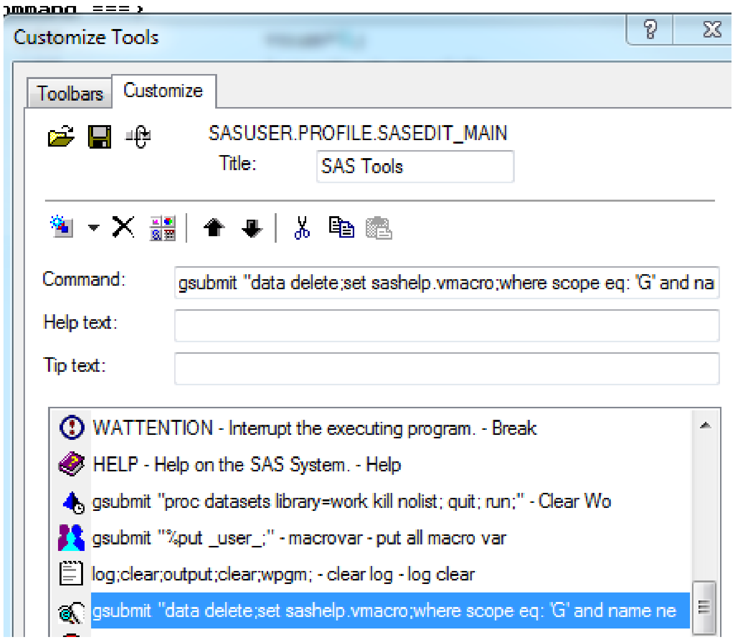
Example 3: Delete all user-defined macro variables

Whenever you develop a program with macros, either due to conflict between locally define macro variable and globally defined macro variable or running multiple different programs with same macro variables or for some other reason issues like macro variable values getting retained or values not getting assigned properly arise. Though, it is good practice to reset values before using them, it may not be feasible sometimes. For such scenarios, you delete the macro variable or assign them to blank value. Instead of manually deleted all the values, we can add a toolbar button to perform that task!

To add the button, type the following data steps in the command field using GSUBMIT function – gsubmit "**data** delete;set sashelp.vmacro;where scope eq: 'G' and name ne: 'SYS;**run**;**data** _null_;set delete;call symdel(name);**run**;"

```
data delete;
set sashelp.vmacro;
where scope eq: 'G' and name ne: 'SYS;
run;
data _null_;
set delete;
call symdel(name);
run;
```

Display 5 shows the command and the icon we assigned to it. Make sure to make the program editor window active before adding this button.



SASUSER.PROFILE.SASEDIT_MAIN

Title: SAS Tools

Command: gsubmit "data delete;set sashelp.vmacro;where scope eq: 'G' and na

Help text:

Tip text:

- WATTENTION - Interrupt the executing program. - Break
- HELP - Help on the SAS System. - Help
- gsubmit "proc datasets library=work kill nolist; quit; run;" - Clear Wo
- gsubmit "%put _user_;" - macrovar - put all macro var
- log;clear;output;clear;wpgm; - clear log - log clear
- gsubmit "data delete;set sashelp.vmacro;where scope eq: 'G' and name ne

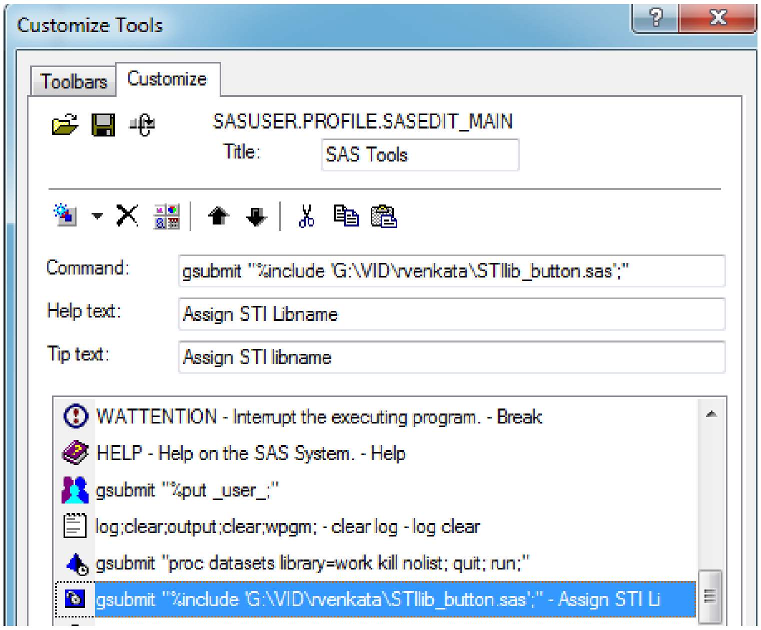
Display 5: Delete all user-defined macro variables

Example 4: Include a SAS program

i. We can also run a SAS program using toolbar button by using %INCLUDE statement and GSUBMIT command.

ii. Type the following in the command field – gsubmit "%include G:\VID\rvenkata\STIlib_buttonw.sas;" The program which I included just has a libname statement.

iii. Display 6 below shows the command and the optional field and assigned icon. Make sure to make the program editor window active before adding this button.



SASUSER.PROFILE.SASEDIT_MAIN

Title: SAS Tools

Command: gsubmit "%include 'G:\VID\rvenkata\STIlib_button.sas';"

Help text: Assign STI Libname

Tip text: Assign STI libname

```
WATTENTION - Interrupt the executing program. - Break
HELP - Help on the SAS System. - Help
gsubmit "%put _user_;"
log;clear;output;clear;wpgm; - clear log - log clear
gsubmit "proc datasets library=work kill nolist; quit; run;"
gsubmit "%include 'G:\VID\rvenkata\STIlib_button.sas';" - Assign STI Li
```

Display 6: Delete all user-defined macro variables

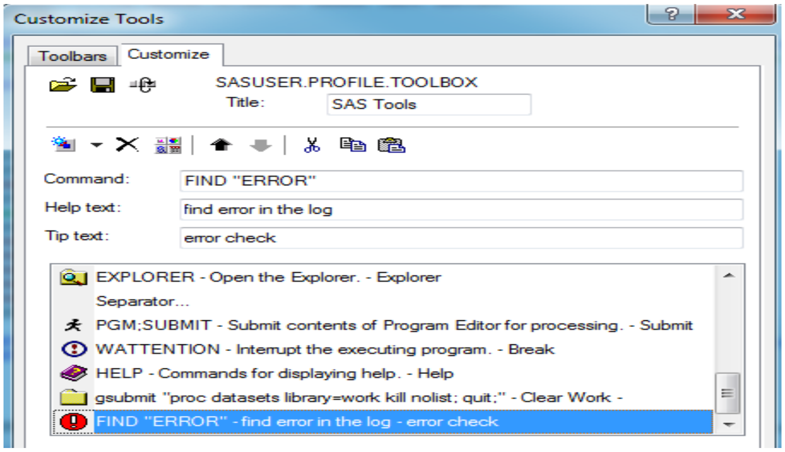
Example 5: Find an ERROR

i. We can also add a toolbar button to find an error. The usual procedure would be go to the log window, hit ctrl+f and type “error” and click “Find Next”.

ii. To add the button, make the log window active. Follow the same generic procedure. Type the following text FIND “ERROR”.

iii. Display 7 shows the command field, Help and Tip field and the icon added to the button.

iv. Once added, we can hit CTRL+L to take you to the log window, then hit CTRL+Home to take you the start of log window then you can click the button.



SASUSER.PROFILE.TOOLBOX

Title: SAS Tools

Command: FIND "ERROR"

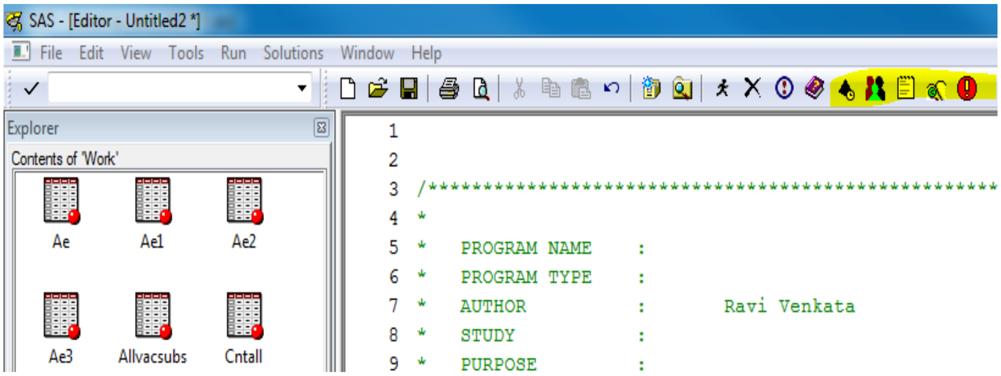
Help text: find error in the log

Tip text: error check

```
EXPLORER - Open the Explorer. - Explorer
Separator...
PGM;SUBMIT - Submit contents of Program Editor for processing. - Submit
WATTENTION - Interrupt the executing program. - Break
HELP - Commands for displaying help. - Help
gsubmit "proc datasets library=work kill nolist; quit;" - Clear Work -
FIND "ERROR" - find error in the log - error check
```

Display 7: Find Error

After adding all the buttons mentioned in examples 1-5, SAS window after customization looks like below,



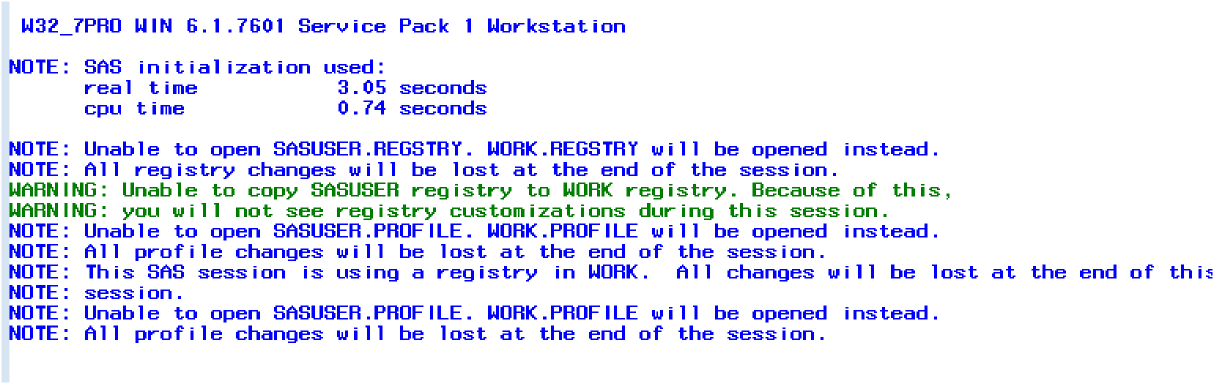
```
/***************************************************
*
*    PROGRAM NAME    :
*    PROGRAM TYPE    :
*    AUTHOR          :         Ravi Venkata
*    STUDY           :
*    PURPOSE         :
```

Display 8: SAS Main Window after customization

As shown in the examples 1-5, you can also add a %put statement or add OPTIONS SYMBOLGEN, MLOGIC etc., to the buttons using the GSUBMIT function as mentioned above.

NON-AVAILIBLITY OF TOOLBAR BUTTONS WHEN MULTIPLE SIMULATENOUS SESSIONS ARE OPENED

If you open a new SAS session without closing the previous session, open multiple sessions simultaneously the toolbar buttons won't be available. You'll see the below warning in Display 9.



```
 W32_7PRO WIN 6.1.7601 Service Pack 1 Workstation

NOTE: SAS initialization used:
      real time           3.05 seconds
      cpu time            0.74 seconds

NOTE: Unable to open SASUSER.REGSTRY. WORK.REGSTRY will be opened instead.
NOTE: All registry changes will be lost at the end of the session.
WARNING: Unable to copy SASUSER registry to WORK registry. Because of this,
WARNING: you will not see registry customizations during this session.
NOTE: Unable to open SASUSER.PROFILE. WORK.PROFILE will be opened instead.
NOTE: All profile changes will be lost at the end of the session.
NOTE: This SAS session is using a registry in WORK.  All changes will be lost at the end of this
NOTE: session.
NOTE: Unable to open SASUSER.PROFILE. WORK.PROFILE will be opened instead.
NOTE: All profile changes will be lost at the end of the session.
```

Display 9: Warning note when you open multiple sessions simultaneously

We can follow the below steps to avoid the issue.

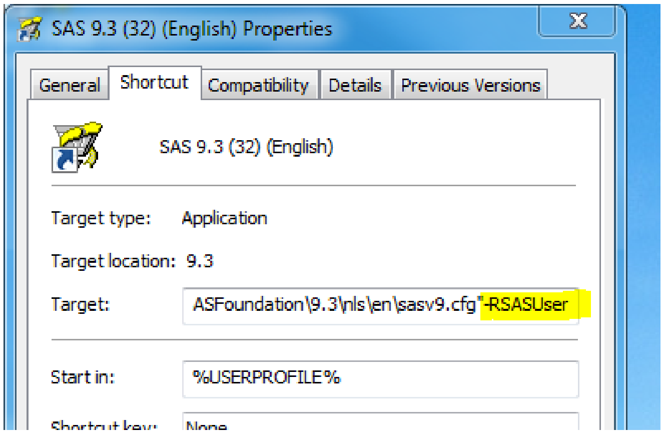
Find and open SASv9.cfg file,if SAS is installed locally located at C:\ProgramFiles\SASHome2\x86\SASFoundation\9.4\nls\en\sasv9.cfg and add the RSASUSER option below anywhere in the file, then save the changes.

-RSASUSER

The above is the ideal option if you have write access to the location. Unfortunately, most of the time IT wouldn't give us write access.

Alternative

Create a shortcut for SAS on your desktop, right click, click properties and add the above option (Display 10),



SAS 9.3 (32) (English)

Target type: Application

Target location: 9.3

Target: ASFoundation\9.3\nls\en\sasv9.cfg"-RSASUser

Start in: %USERPROFILE%

Shortcut key: None

Display 10: Fix the warning issue of multiple sessions simultaneously

If SAS is installed remotely, you might have to mention the same fix, to your IT team.

BECAUSE SHARING IS CARING

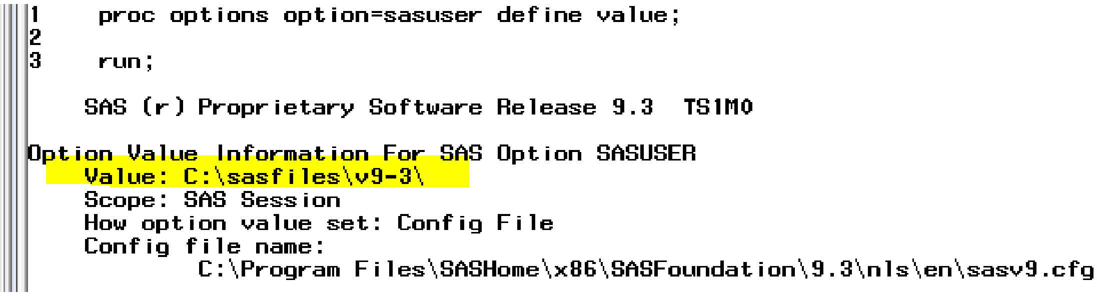
Because sharing is caring, we would like to discuss the way to share it. It might be useful if you would like standardize in your team or group or just sharing it with your co-worker. Follow the steps below,

Type the SAS procedure below,

proc options option=sasuser define value;

run;

Check the log and find out the following highlighted path in Display 11.



```
     proc options option=sasuser define value;

     run;

    SAS (r) Proprietary Software Release 9.3  TS1M0

Option Value Information For SAS Option SASUSER
    Value: C:\sasfiles\v9-3\
    Scope: SAS Session
    How option value set: Config File
    Config file name:
            C:\Program Files\SASHome\x86\SASFoundation\9.3\nls\en\sasv9.cfg
```

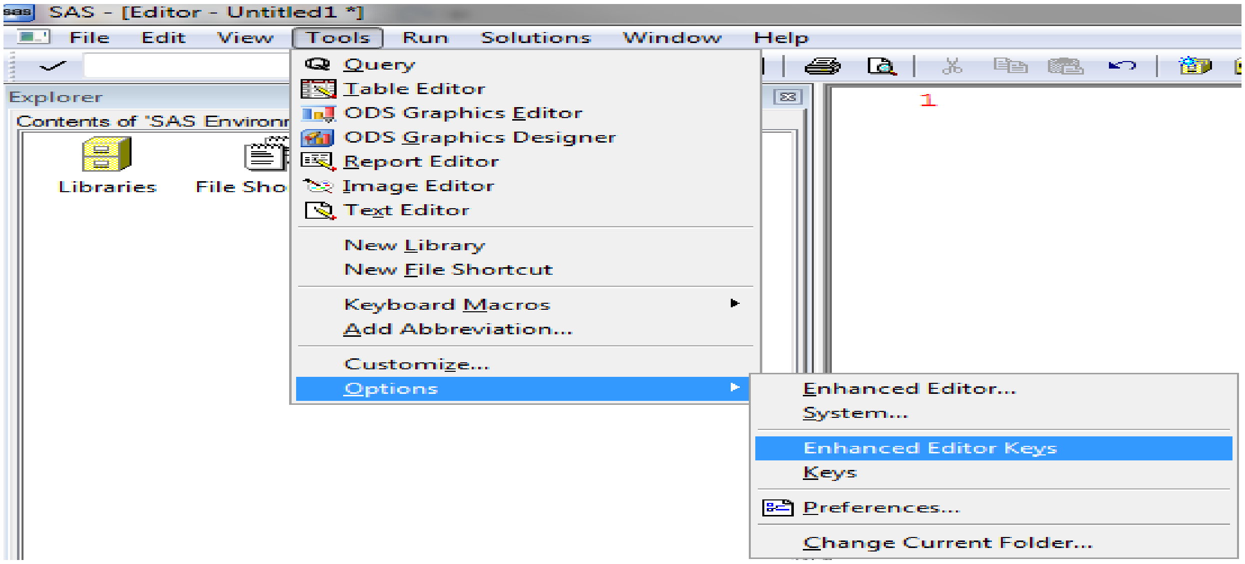
Display 11: Sharing your buttons

Go to the above path and copy the profile.sas7bcat file to your desktop.

This file will have all your toolbar buttons. You can share this file and they can replace their profile.sas7bcat with the file you send. We would advise backing up your profile. sas7bcat before replacing it though!

The Keys

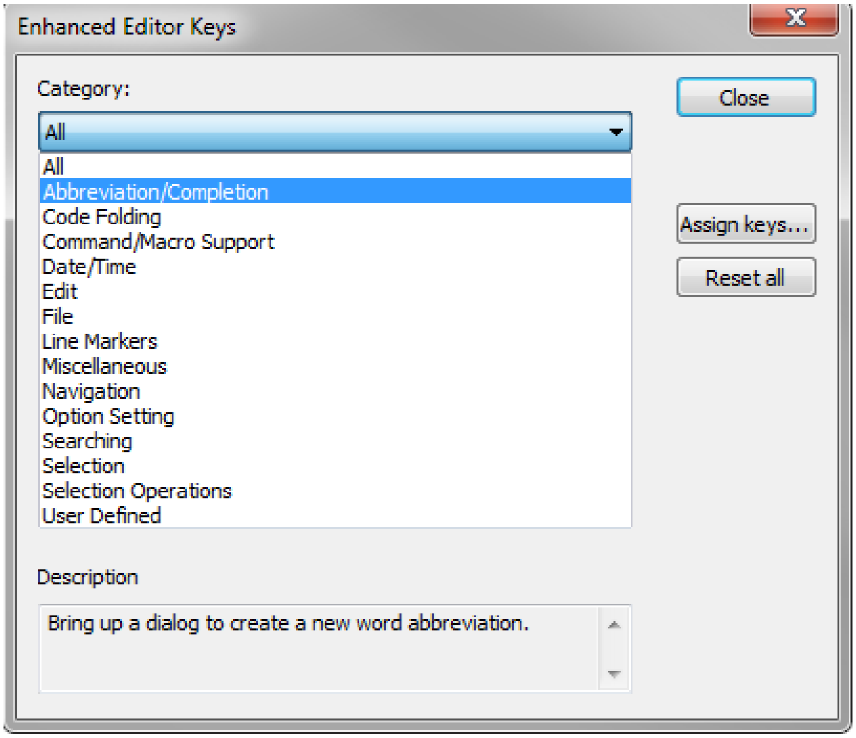
Sometimes, it feels good not to take your hands of the keyboard when you are programming so, SAS, has some inbuilt default hotkeys that can make programming more fun and make things simpler depending on the kind of work we do. We can also customize the hot keys as we want. These keys can be found under **TOOLS<OPTIONS<ENHANCED EDITOR KEYS** as shown in Display 12 and 13.



Display 12: How to navigate to Enhanced Editor Keys.

Example of the default hot keys that might be very useful in programming are CTRL+/ to mark code as comments, CTRL+SHIFT+/ to make comments as code, CTRL+F2 to mark a line in a program, F2 goes to the next marked line, etc.

There are variety of category hot keys that are available in Enhanced Editor Keys window, one among them that we think is helpful is the Abbreviation/Completion which can be opened easily by pressing CTRL+SHIFT+A.



Bring up a dialog to create a new word abbreviation.

Display 13: Enhanced Editor Keys window

ADDING AN ABBREVIATION

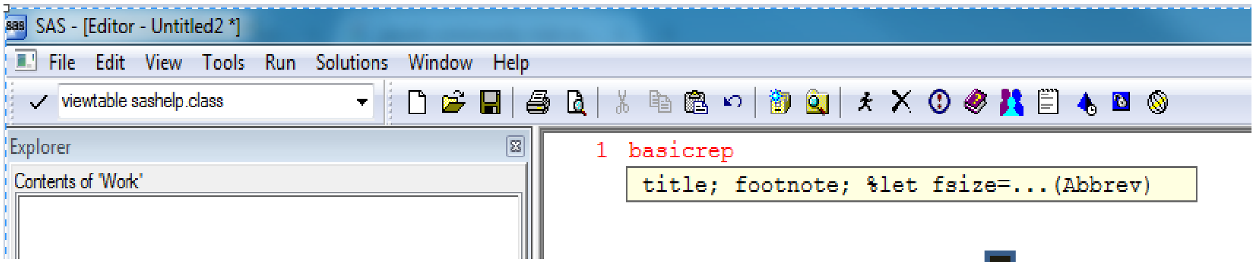
Not all the code we write every time is new. Most of the code can be repetitive including SAS functions. So, it is good to have common repetitive code saved somewhere in our SAS session. To create an abbreviation, do the following,

i. Copy the code you want to call every time with a key word you can remember for that code and press **CTRL+SHIFT+A** a pop-up window appears where we past the code under Text to insert for abbreviation.

ii. Add the key word to call your code and press ok to save the abbreviation.

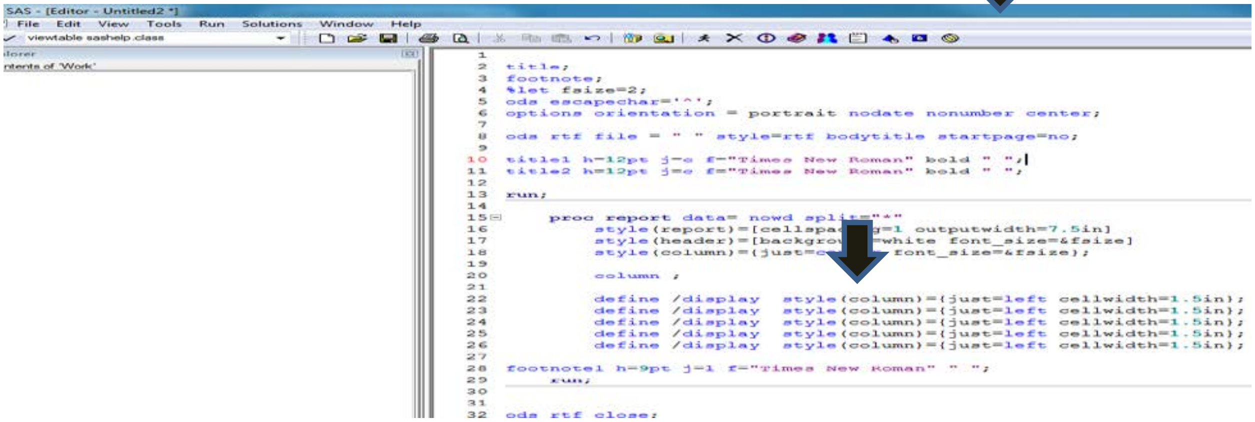
iii. Display 14 shows a predefined abbreviation '**BASICREP'** saved for the proc report template so, when we type the word '**BASICREP'** in the SAS windows editor, there is a suggestion showed in yellow. When we press the key "Enter" we can see the code printed in the SAS editor window (Display 15).

Note: These abbreviations are case sensitive.


SAS - [Editor - Untitled2 *]
File Edit View Tools Run Solutions Window Help
viewtable sashelp.class
Explorer
Contents of 'Work'

```
basicrep
title; footnote; %let fsize=...(Abbrev)
```

Display 14: SAS Window after typing a predefined Abbreviation

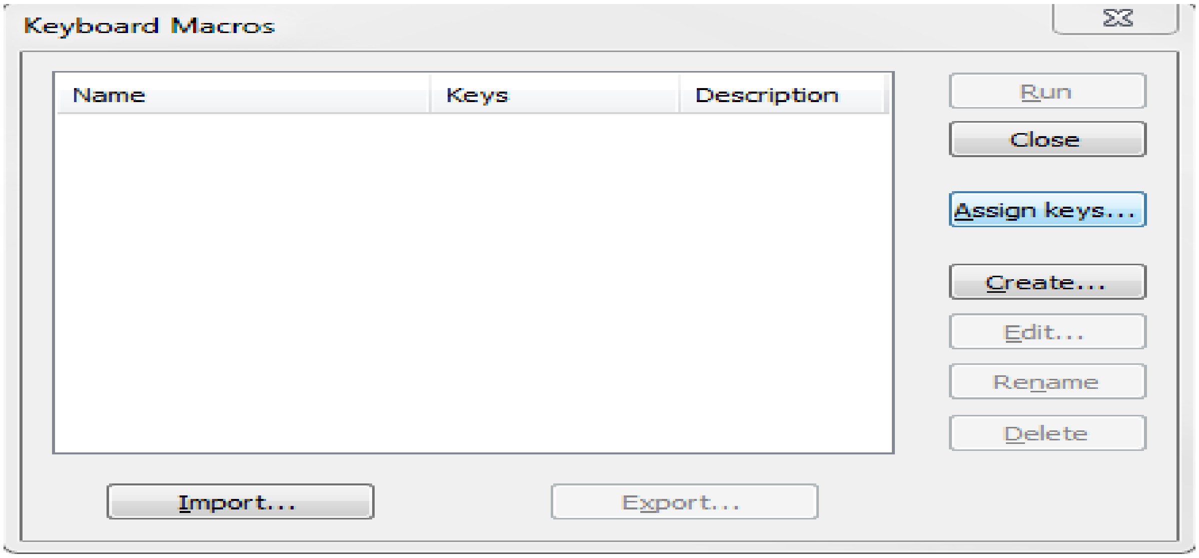

SAS - [Editor - Untitled2 *]
File Edit View Tools Run Solutions Window Help
viewtable sashelp.class
Explorer
Contents of 'Work'

```

title;
footnote;
%let fsize=2;
ods escapechar='^';
options orientation = portrait nodate nonumber center;

ods rtf file = " " style=rtf bodytitle startpage=no;

title1 h=12pt j=c f="Times New Roman" bold " ";
title2 h=12pt j=c f="Times New Roman" bold " ";

run;

      proc report data= nowd split="*"
           style(report)=[cellspacing=1 outputwidth=7.5in]
           style(header)=[background=white font_size=&fsize]
           style(column)={just=center font_size=&fsize};

           column ;

           define /display  style(column)={just=left cellwidth=1.5in};
           define /display  style(column)={just=left cellwidth=1.5in};
           define /display  style(column)={just=left cellwidth=1.5in};
           define /display  style(column)={just=left cellwidth=1.5in};
           define /display  style(column)={just=left cellwidth=1.5in};

footnote1 h=9pt j=l f="Times New Roman" " ";
      run;


ods rtf close;
```

Display 15: SAS Window with printed Abbreviation

Keyboard macros

There is another useful hotkey function where we can assign customized hot keys that are useful in our day to day programming and are called as keyboard macros. To assign such keys press CTRL+SHIFT+M and then click assign keys (Display 16).



Keyboard Macros

Name	Keys	Description

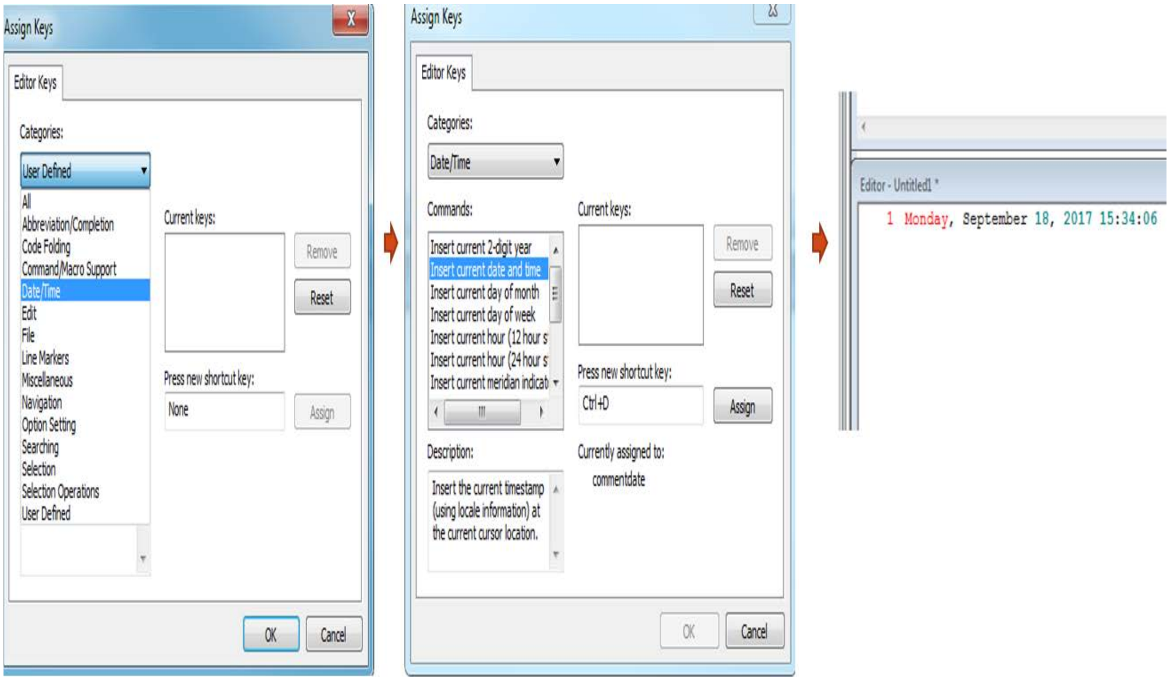
Display 16: SAS Window- Keyboard Macros

The below example shows how to assign a keyboard macro for printing date and time in SAS editor window.

i. After pressing CTRL+SHIFT+M, Select Date/Time and click ok

ii. Assign any combination of keys you prefer and click assign them and press ok to save. For this example, we assigned "Ctrl+D" key combination (Display 17).

iii. Now, make the SAS editor windows as the active window and press CTRL+D to print date and time.



Assign Keys

Categories: User Defined

Press new shortcut key: None

Assign Keys

Categories: Date/Time

Commands: Insert current date and time

Press new shortcut key: Ctrl+D

Currently assigned to: commentdate

Description: Insert the current timestamp (using locale information) at the current cursor location.

Editor - Untitled1 *

```
1 Monday, September 18, 2017 15:34:06
```

Display 17: SAS Windows- Keyboard Macros example

COMBINING THE AVAILABLE KEYBOARD MACROS.

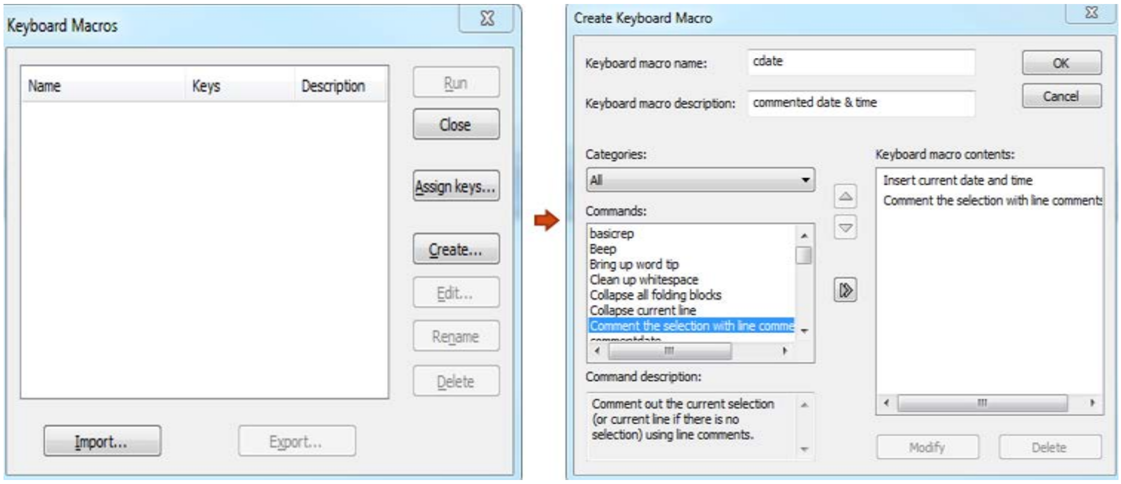
We can also combine the existing available keyboard macros to create new macro hotkeys.

Below is one of the example to show how we can do that (Display 18).

i. Press **CTRL+SHIFT+M** then click **CREATE**

ii. From the list of Commands, select the commands you want to combine into Keyboard macro contents (Display 18).

iii. Assign a name for the selected combination in the field keyboard macro name. In this example, we named it as “cdate” and add an optional description in the keyboard macro description field (Display 18). Press ok to save.



Keyboard macro name: cdate

Keyboard macro description: commented date & time

Keyboard macro contents:
Insert current date and time
Comment the selection with line comments

Command description: Comment out the current selection (or current line if there is no selection) using line comments.

Display 18: Combining Keyboard macros

ASSIGNING A HOTKEY TO A COMBINED KEYBOARD MACRO

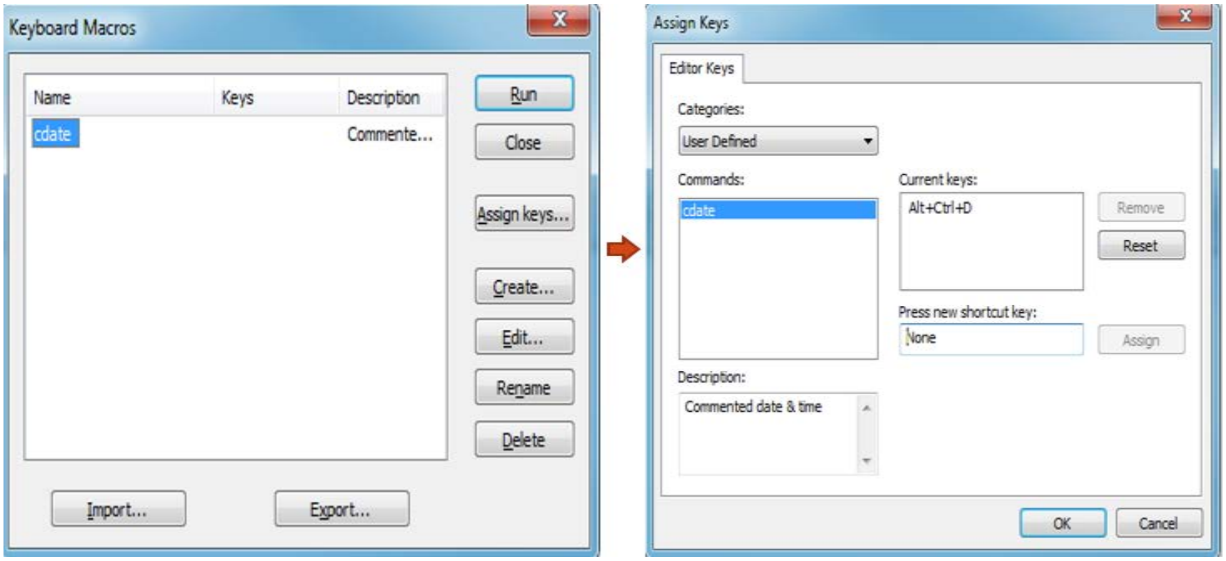
To assign hotkeys to the keyboard macro, press **CTRL+SHIFT+M** where we can find the predefined macros.

i. Select predefined keyboard macro – “**cdate”** (Display 18) and press **Assign Keys**.

ii. Keeping the current keys field as active, press the keys that you want to assign and press ok to save.

iii. In our example, we assigned **ALT+CTRL+D** as the commented date hotkey command (Display 19).

iv. Keeping SAS Editor Window active, when you press “**ALT+CTRL+D”,** SAS will print a commented date and time in the editor window!



Display 19: Assigning hotkey to Keyboard macro

AGAIN, SHARING IS CARING

Sharing your macros with your team might help and save time as well. Yes, there is an easy way to do that and we discuss that below (Display 20).

Exporting

i. Press **CTRL+SHIFT+M** to open Keyboard Macros window.

ii. Select all the macros that you want to export into a single file, click **Export.**

iii. All the selected macros can be stored in a single “. kmf” file.

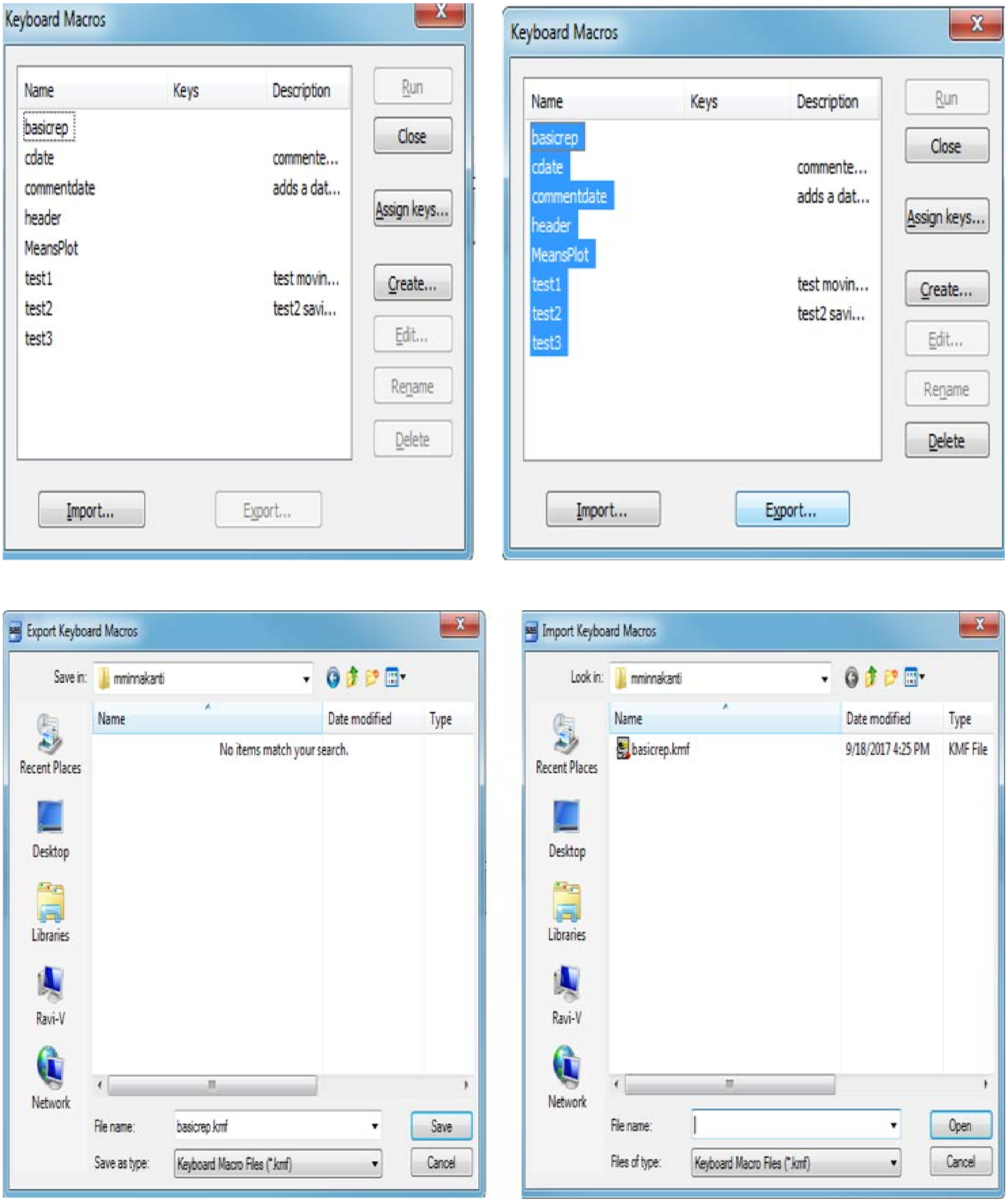
Importing

i. Press **CTRL+SHIFT+M** to open Keyboard Macros window.

ii. Click Import and select the “.kmf” file which has your keyboard macros and click Open.

iii. All the selected macros will be imported to your SAS environment.

NOTE: Macros with the same name will get overwritten.



Name	Keys	Description
basicrep		
cdate		commente...
commentdate		adds a dat...
header		
MeansPlot		
test1		test movin...
test2		test2 savi...
test3		

Save in: mminnakanti

Name	Date modified	Type

No items match your search.

File name: basicrep.kmf

Save as type: Keyboard Macro Files (*.kmf)

Look in: mminnakanti

Name	Date modified	Type
basicrep.kmf	9/18/2017 4:25 PM	KMF File

File name:

Files of type: Keyboard Macro Files (*.kmf)

Display 20: Exporting/Importing Keyboard Macros

QUICK-TIPS

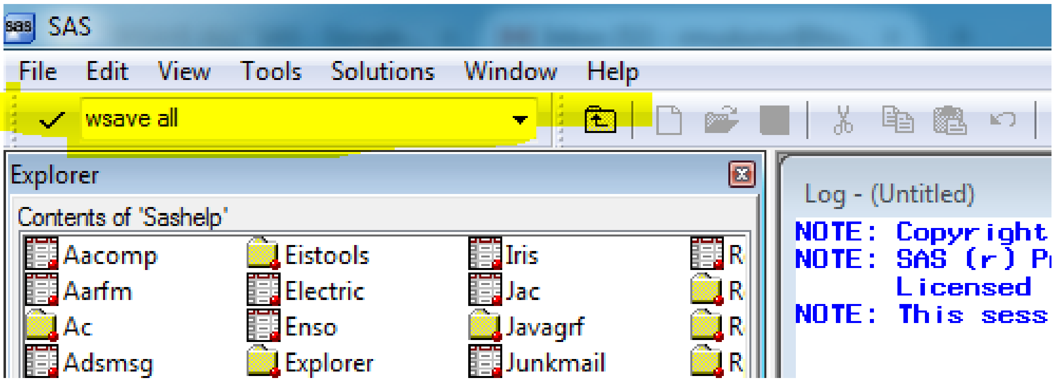
Formatting your code

If you also have SAS Enterprise guide and SAS 9.2/9.3/9.4 installed, it is easy to format your code. Open SAS EG, paste your code and hit ctrl+I. It will format your code!

Change default way of display datasets in the explorer window

i) Click View, select your preferred view- List, Details, Small Icons or Large icons.

ii) Type “wsave all” as shown in Display 21.



```
wsave all
```

```
NOTE: Copyright
NOTE: SAS (r) P
      Licensed
NOTE: This sess
```

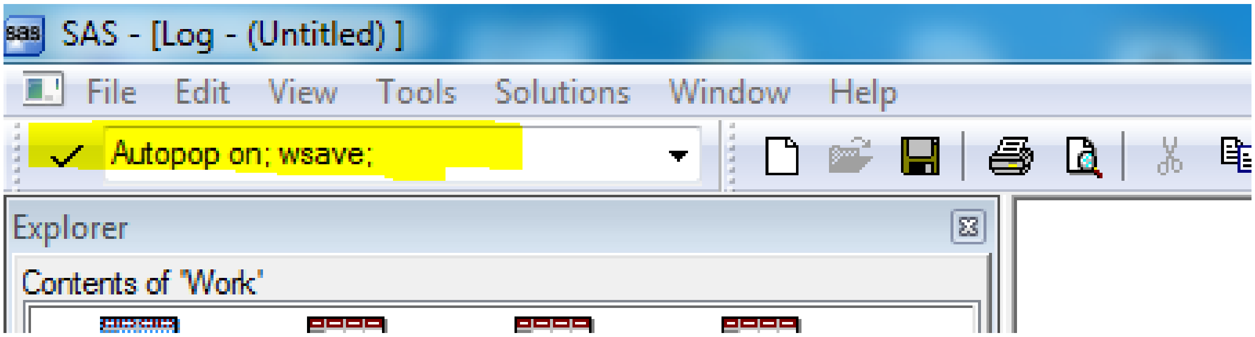
Display 21: Change default way to view the datasets.

Show me the log first!

i. Click and make the log window your active window.

ii. Type “Autopop on; wsave;” in the command window and hit enter as shown in Display 22.

iii. Whenever you submit the program, it will directly take you to the log.


SAS - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Explorer
Contents of 'Work'

```
Autopop on; wsave;
```

Display 22: Show the log first.

CONCLUSION

There are many inbuilt SAS options that can save time and increase efficiency for a programmer which can also be customized and shared among your team. This was an effort to show how some of the SAS Windowing Environment options work, that, we thought are useful to other SAS users.

REFERENCES

http://support.sas.com/documentation/cdl/en/hostwin/69955/HTML/default/viewer.htm#n0ygxckxoqkz7vn115sybx7ihf48.htm

http://support.sas.com/kb/32/774.html

ACKNOWLEDGMENTS

Thanks to SAS Tech Support

CONTACT INFORMATION

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

Raviteja Venkata

raviiteja@gmail.com

Mahesh Minnakanti

minnakantimahesh@gmail.com