Creating and using code snippets with SAS Enterprise Guide

Mark McLean, RTI International

ABSTRACT

When programming in SAS there can be a lot of repetitive code, such as merges, proc prints, proc sorts, etc. This paper will demonstrate an easy way to automate writing code so you can program faster and with less syntax errors. It will show how to create and utilize snippets of code in SAS Enterprise Guide by recording and saving macros. For example, instead of typing out the code every time you want to merge two datasets you can create and run a tailored macro, thereby increasing efficiency and accuracy. No more missing parenthesis or semicolons!

INTRODUCTION

Have you ever realized how much of the code in your programs is different variations of similar things? It is just rehashed with different dataset names or slightly different customizations? In a typical program most of it could consist of infiles, sorts, merges, macros, prints and exports. There is a way in SAS Enterprise Guide (EG) to create a code snippet that will give you the "bones" of all these steps. Then all you would need to do is run the snippet and just fill in the few parameters that you need to complete it. This paper will take you through the steps of creating these and how you can utilize them in order to save yourself time and effort. Not only would new programmers find this very beneficial, but also more experienced ones and even teams.

HOW TO CREATE SNIPPETS OF CODE IN SAS ENTERPRISE GUIDE

The first thing you need to decide before creating the snippet in SAS Enterprise Guide is what code would be helpful to replicate? The most helpful are ones that are not too complex and do not need a lot of customization. An example of this, and will be used as an example throughout this paper, is merging two datasets.

CREATE NEW CODE SNIPPET

Step 1 is to open SAS EG then go to: Program → New Snippet (see Figure 1). You can also type Ctrl+Shift+A to bring up that window.
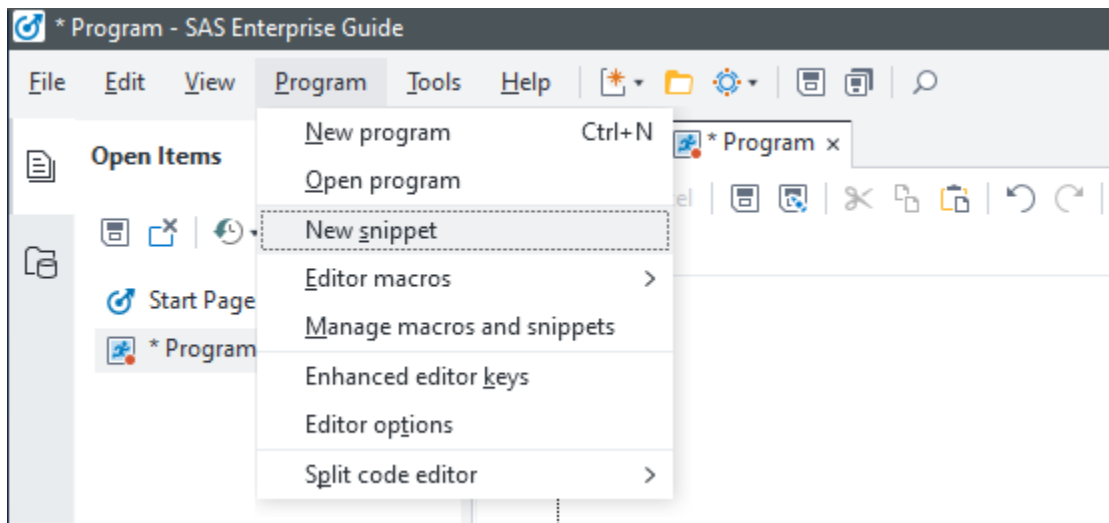

Figure 1

In the 'Abbreviation:' field, type in what you want the code snippet to be named. For this example it will be called 'Merge'. In the 'Text to insert' section either type or paste the code you want to use for the snippet. (see Figure 2)
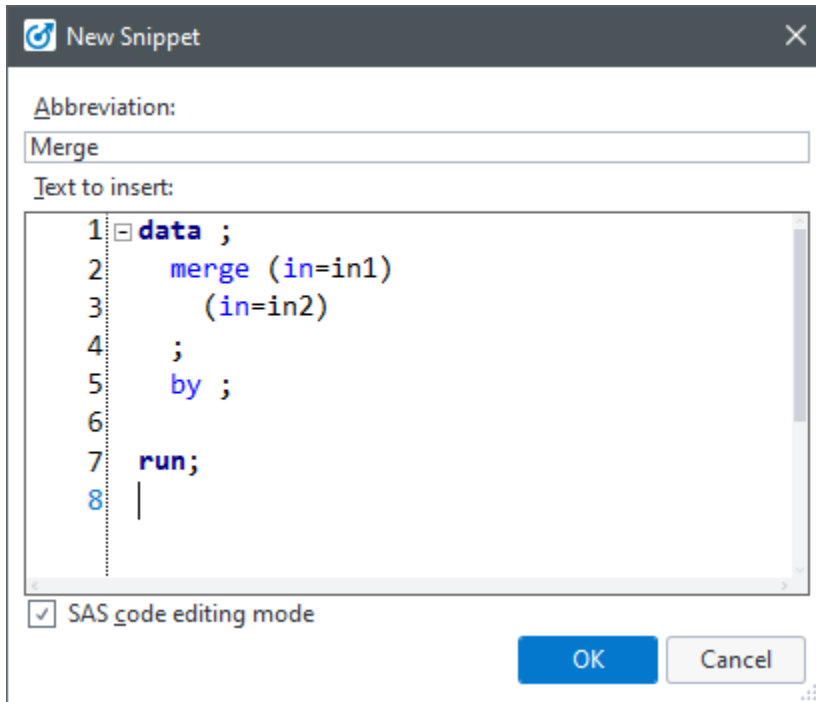

Figure 2

After the code snippet has been created, you can edit it to add a description and shortcut keys. To add a description, which is optional, go to Program → Manage macros and snippets (see Figure 3).
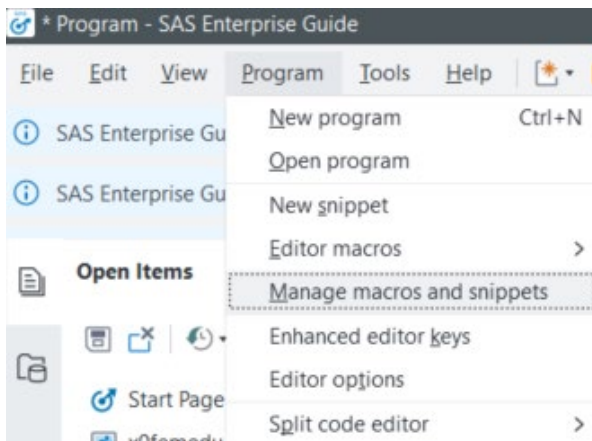

Figure 3

Click on the code snippet you created then hit the 'Edit' button. Write the description in 'Keyboard macro description' then click OK. For this example, I put 'merge 2 datasets' (see Figure 4).
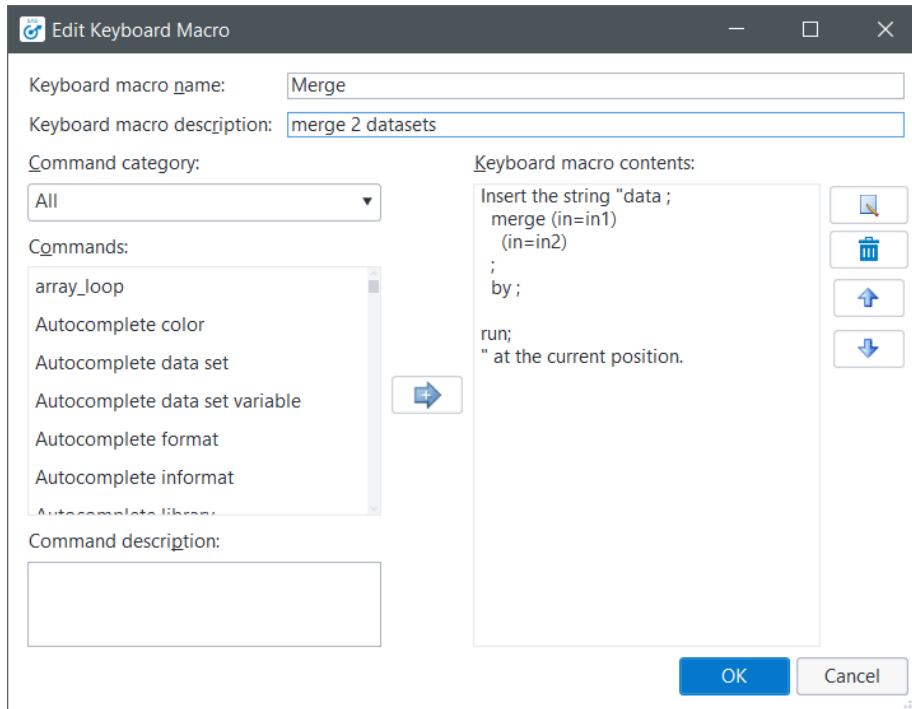
2

Figure 4

ASSIGN SHORTCUT KEYS

Now that the code snippet has been created, it needs the shortcut keys assigned. Think about what would be easy to remember for when you want to run the snippet. For example, with the Merge snippet example I will assign keys Ctrl+Shift+M. Here are the steps to set that up:

1. Open the 'Manage Macros and Snippets' window by going to Program → Manage macros and snippets (see Figure 3)

2. From there click 'Assign Keys'.

3. Under the 'Commands' list find the snippet you saved and click on it.

4. Click on 'None' under 'Press new shortcut key:'.

5. Now you need to hit the shortcut keys that you want to assign, click 'Assign', then click the OK button.

You can see in Figure 5 that the 'Current keys' assigned for the Merge snippet are now Ctrl+Shift+M.
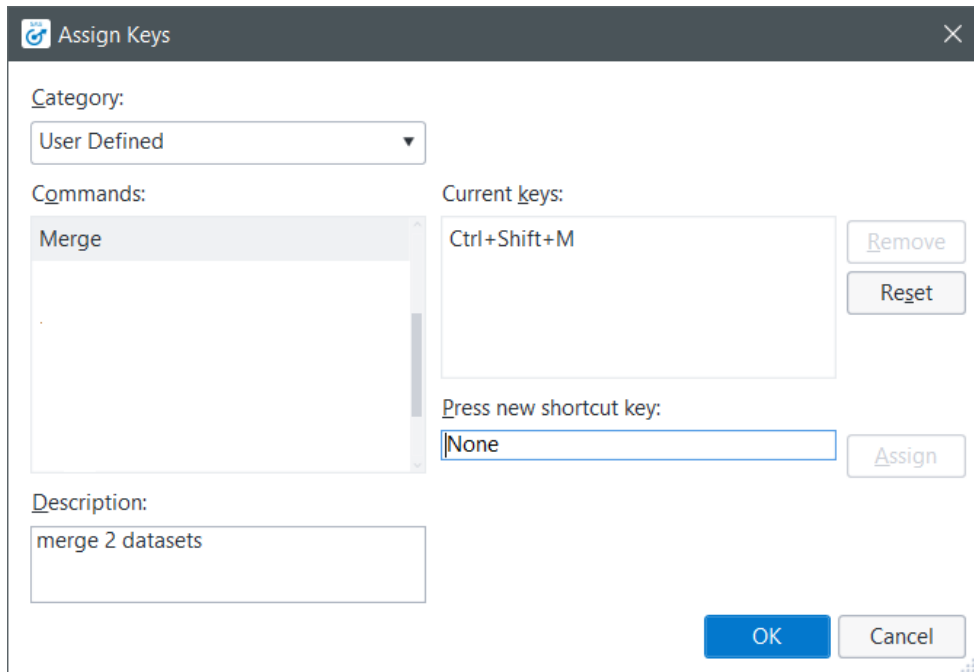
Figure 5

One thing to note is that it is not required that a shortcut key be assigned. Section "Using code snippets in your programs" will go over in more detail how to run them without shortcut keys.

USING CODE SNIPPETS IN YOUR PROGRAMS

You have created your code snippets and they are now ready to use while you are writing your SAS program. When you get to the point where it would be helpful to call one there are two ways to do it. If shortcut keys are assigned then just hit those keys, it's that easy! If the code snippet does not have a shortcut keys assigned, then it can be called by:

1. Typing Alt+F8

2. Double-clicking the one from the list that you want to run (you can also single click then hit the Run button)

This method can be helpful if you have more code snippets than can easily be assigned to shortcut keys.

OTHER USEFUL TIPS WITH CODE SNIPPETS

EXAMPLES OF CODE SNIPPETS

Everyone will have their own set and variations of code snippets that will be the most useful to them, but here are ideas of some that could be helpful:

Name	Code Snippet
Array	ARRAY arxx[*] xx xx ; DO N=1 TO DIM(arxx) ; arxx[N] END; drop N;

| CSV Export | ```
proc export data=xx
    outfile="/xx/xx.csv"
    dbms=csv
    replace;
run;
``` |
| CSV Import | ```
proc import datafile="/xx/xx.csv"
  out=xx
  dbms=CSV replace;
  guessingrows=max;
run;
``` |
| Excel Export | ```
proc export
  data=dsnm
  dbms=xlsx
  outfile="/xx/xx.xlsx"
  replace;
run;
``` |
| Excel Import | ```
proc import  out=xx
  datafile="/xx/xx.xlsx"
  dbms=xlsx replace;
  *SHEET="Sheet1";
  *range="Sheet1$A1:Z999";
  getnames=yes;
run;
``` |
| Find Duplicates | ```
proc sort data= xx ; by xx; run;
data getdups;
  set xx ;
  by  xx;
  if first.xx  and last.xx  then delete;
run;
``` |
| Macro Template | ```
%macro macname(varnm1);
  if &varnm1.= xx then xx ;
``` |

| | ```
%mend macname;

%macname(xx);
``` |
| Merge Datasets | ```
data ;
  merge (in=in1)
    (in=in2)
  ;
  by ;

run;
``` |
| Proc Contents | ```
proc contents data= ;
run;
``` |
| Proc SQL | ```
PROC SQL;

  create table xx as
  select
  from
  ;

QUIT;
``` |
| Proc Print | ```
proc print data= (obs=10);
run;
``` |
| Proc Sort | ```
proc sort data= ; by ;
run;
``` |
| Proc Transpose | ```
proc transpose data=xx out=t_xx /*prefix=*/;
  *by xx;
  *id xx;
  *var xx;
run;
``` |

IMPORT/EXPORT CODE SNIPPETS

SAS EG lets you import and export code snippets. This can be helpful since someone already may have one setup that could be useful to you. You can also let people use ones that you have setup. Just go to "Manage macros and snippets" (see Figure 3). If importing, click on the Import button and select the location it was saved. If exporting, click the code snippet(s) you want to export, click the Export button, then select the location to save them to and click Save.

CONCLUSION

It is easy to set up code snippets and utilize them in your programming. Doing this will speed up coding since it will require less typing and reduce errors. Not only can they have a big impact for a programmer, especially a newer one, but teams as a whole. With the import/export features for SAS EG, teams could setup a repository to be shared across team members. This would make specific code readily available and have it be more standardized.

CONTACT INFORMATION

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

Mark McLean
RTI International
mrmclean@rti.org