Paper 240-28
Practical Tips to Customize a SAS® Session

Beilei Xu, Merck & Co., Inc., Rahway, New Jersey
Xiaohui Wang Merck & Co., Inc., Rahway, New Jersey

ABSTRACT

SAS Programmers often find that a nicely customized SAS session can greatly facilitate their routine jobs and save time spent on repetitive tedious tasks. This paper discusses some easy ways to customize a SAS session, including toolbar customization, shortcut keys, the use of the SAS configuration file and the autoexec.sas file. All the examples work on SAS V8 Windows platforms.

INTRODUCTION

You can customize your SAS session at various stages. You can customize the toolbars and shortcut keys during an interactive SAS session, and you can modify SAS configuration file to set SAS system options when SAS starts up. Also you can put options and SAS statements in an autoexec.sas file that SAS runs right after initialization. Several of these options are discussed below.

CUSTOMIZATION METHODS

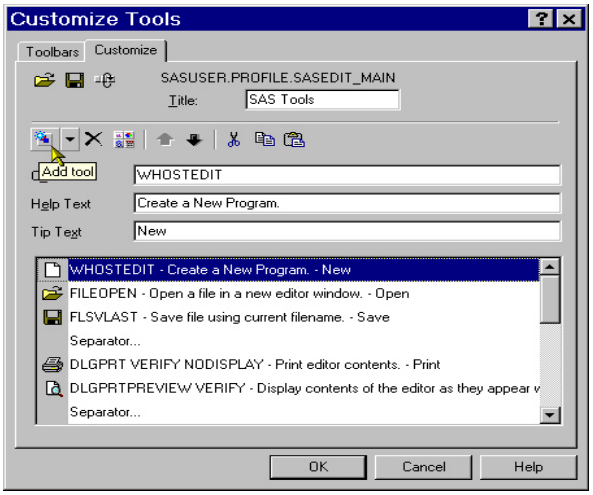
I. Customizing the Toolbar:

Toolbars offer great opportunities for customization. Each SAS Window has a default toolbar. You can add new buttons or organize the existing buttons according to your preference.

For example, you might need to include your standard program header template in every program you develop. To create the button for that, you will need to use the "Customize Tools" dialog box. With the Enhanced Editor active, use either of the following methods to open this dialog box:

- enter **tooledit** in the "command bar", or
- select the menu Tools → Customize…, or
- right-click anywhere in the toolbar area and choose Customize… from the popup menu.

The "Customize Tools" dialog box will appear:



SASUSER.PROFILE.SASEDIT_MAIN
Title: SAS Tools
WHOSTEDIT
Help Text: Create a New Program.
Tip Text: New

WHOSTEDIT - Create a New Program. - New
FILEOPEN - Open a file in a new editor window. - Open
FLSVLAST - Save file using current filename. - Save
Separator...
DLGPRT VERIFY NODISPLAY - Print editor contents. - Print
DLGPRTPREVIEW VERIFY - Display contents of the editor as they appear v
Separator...

The **Toolbars** tab is for general toolbar settings and the **Customize** tab is for defining tools on the toolbar.

To add the button that will insert the header template text in your program, follow these steps:

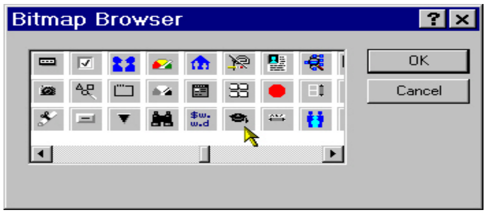
1. Click on the **Add tool** button, , or the down arrow ▼ right to it and select **Blank Tool**. This creates a template for a new tool button in the list box.

2. In the **Command** field, type **include "C:\My Documents\header.txt"**. This command copies the content of the header.txt file to the Enhanced Editor window.

   In the **Help Text** field, type **create header**. This text will show up in the message area of the status bar when the mouse pointer is moved over the created button.

   In the **Tip Text** field, type **header**. This text will show up under the button when the mouse pointer is moved over the created button.

3. Click on the **Change icon** button to open the Bitmap Browser and select a new image for your button. Select a bitmap that you think is appropriate for the action and click on OK.



4. Use the **Move Up** and **Move Down** arrows, which are next to the **Change icon** button, to move the button to the desired location. You can add a "separator" before the button by clicking on the down arrow right to the **Add a tool** button and selecting the **Separator**.

5. To delete a tool button, select that button in the list box and click the **Delete** button.

6. Click on the **Save the toolbar** button , which will bring up the "Save Tool" dialog box. Clik on OK without changing anything in the **Save tools** dialog box to save the new button to the active window's toolbar. In this example, it saves as sasuser.profile.sasedit_main, the defined toolbar for the Enhanced Editor.

To summarize, the key step is specifying the **Command** field, which tells SAS what to do when you click the button. It can be just a single SAS command, or a series of SAS commands separated with semicolons, or a macro call containing SAS commands. See SAS OnlineDoc for SAS commands available under Windows.

Here are some examples of other useful shortcuts that can be placed on a toolbar as buttons:

1. **recall**
   Use the SAS command **recall** to recall the last submitted block of statements. This is equivalent to selecting Recall

Last Submit from the Run menu or pressing the shortcut key F4.

2. **submit; wnextedit; clear**
In this example, several SAS commands will be executed in one click. These commands submit the content of the Enhanced Editor and clear the editor window. In SAS V6, the editor window is cleared by default on submit, but in SAS V8, Enhanced Editor window is not cleared upon submit. You can set it as default by going to Tools→Options→EnhancedEditor→General, and checking the **clear text on submit** box.

But by creating the button assigned to commands **submit; wnextedit; clear,** along with the default submit botton , you can toggle between the two options as quickly as needed. For the buttons to work, make sure the **clear text on submit** option is unchecked.

3. **Log; clear; output; clear;**
This clears both log and output windows.

4. **File "C:\MySAS\sasl_%sysfunc(today(),date9.).sas"; clear;**
This saves the contents of the active window to the file "C:\MySAS\sasl_<today's date>.sas". If today's date is August 20, 2002, the file will be saved as C:\MySAS\sasl_20AUG2002.sas. It is useful for saving programs subjected to daily updates.

5. **include** and **gsubmit**
The SAS command **include** brings the lines into the editor window without executing them. To execute those lines, another command **submit** is needed. For example, if you want to create a button to execute a program "C:\MySAS\signon.sas", the commands should be **include C:\MySAS\signon.sas; submit**.

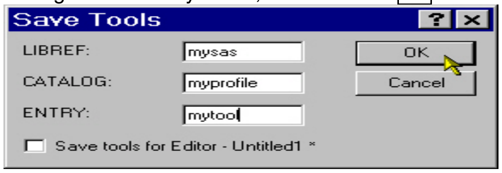
Alternatively, you can specify **gsubmit "%include 'C:\MySAS\signon.sas';"**. This will execute the signon.sas program without bringing it to the editor window.

Gsubmit command submits the following SAS statements within the quotes. For example:
gsubmit '*);*/;/*'*//*"*/;%mend;' submits the symbol string which balances the unbalanced quotes or macro definition. Assigning this subtle string to the button, you don't need to remember it.

Gsubmit can also be used to submit SAS statements in the Windows clipboard. **gsubmit buf=default** is equivalent to the menu item Submit Clipboard in the Program Editor.

Change the toolbar back to default setting
You can restore a toolbar to its default setting by clicking on the **Restore Defaults** button, , in the "Customize Tools" dialog box.

Share the toolbar with others
After adding or modifying buttons to a toolbar, you can save it for the active window (see step 6 above). You can also save it to another libref, catalog and entry name for later use. Go to the "Customize Tools" dialog box and click the **Save the toolbar** button, . In the **Save Tools** dialog box, specify the appropriate libref, catalog and the entry name, then click the OK button.


Save Tools

LIBREF: mysas

CATALOG: myprofile

ENTRY: mytool

□ Save tools for Editor - Untitled1 *

This will save the toolbar to myprofile.mytool in the SAS library "mysas". Now, we can share "mytool" with others.

To apply "mytool" toolbar to other user's window, make that window active first. Second, click the **Open a toolbar** button in the "Customize Tools" dialog box, say "No" to the "Do you want to save changes..." prompt. Input the libref, catalog and entry name, and click OK to close this dialog box. Mysas.myprofile.mytool is now opened. Then click the **Save the toolbar** button and check the window dependent option **Save tools for the *active window name*** in the **Save tools** dialog box and click OK. Now "mytool" overwrites the defined toolbar. Then you can enjoy the convenience of those tools.

II. Shortcut Keys and Keyboard Macros
Shortcut Keys are equivalent to toolbar buttons. It's just a matter of your preference. SAS also assigns some default key definitions. (check the following location for the default key definitions, *SAS OnlineDocV8→SAS Companion for Windows Environment→PART4 Apendices→Default Key Settings for Interactive SAS Sesssions)*.

You can customize and add your key definitions by assigning commands to keys. Type **keys** in the command bar or select Keys from menu Tools→ Options→Keys, the "Keys" window opens and allows you to assign commands to keys. Or use **keydef** command to define the shortcut. Issuing **keydef "SHF F2" dlgpref** in the command box, pressing the "Shift +F2" key will open the Preferences dialog box.

Some shortcut keys don't have commands equivalents. For example, in the Enhanced Editor, the shortcut keys Ctrl+Shft+L convert the selected text to lowercase; Ctrl+Shft+U convert the selected text to uppercase. There are no SAS commands to convert texts' cases.

In SAS V8, the Enhanced Editor has a keyboard macro feature. You can create a keyboard macro to group a series of Enhanced Editor commands and instructions together as one command. You can run it either from Tools menu, or by using a keyboard shortcut assigned to that keyboard macro. But, before creating your own shortcut, do check the shortcuts that already exist to avoid duplicating effort.

There are two ways to create keyboard macros. The easiest way is to use the keyboard macro recorder. Simply go to Tools→Keyboard Macros→Record New macro to start recording the sequence of your actions in the Enhanced Editor window. The other way is to select series of commands you want SAS to perform from the **Commands** list box in "Create Keyboard Macro" dialog box invoked by going to Tools→Keyboard Macros→Macros…→Create.

The keyboard macro can be shared by going to Tools→ Keyboard Macros→ Macros…→ Import… or Export… to import or export it to or from a folder.

To assign a shortcut to the macro, go to Tools→ Keyboard Macros→ Macros… , select the keyboard macro and click on Assign keys… to get the "Assign Keys" dialog box. Then place the cursor in the **Press new shortcut key:** field, and press the keys you want to assign to the macro on the keyboard. Make sure the **Current Assigned to** field in the "Assign Keys" dialog box has a value of **None**, if you don't want to overwrite the existing key assignment. Then click on OK to save the assignment.

Creating keyboard macros can help writing and editing your programs. Here are some practical ideas to use keyboard macros:

1. record a comment line to a keyboard macro to insert a comment line in the program when it is needed. It not only saves

your time on typing the "***"s, but also makes your programs consistant with same length of comment lines.

2. record often used SAS statements or standard program header to keyboard macros. For example, SAS statements like "proc contents; run;", "proc print; run;", often used label assignments, etc can all be recorded into keyboard macros.

III. Modifying Configuration File

The SAS configuration file enables you to specify SAS system options that are used to establish your SAS session. A default SAS configuration file named sasv8.cfg is created in the !SASROOT folder when SAS V8 is installed. You can create your own configuration file to override the default one. Refer to the SAS OnlineDoc for help on how to create and use an alternate configuration file.

There are many ways to specify values for SAS system options. However some of the system options can be specified only when a SAS session is initialized.

The syntax for setting SAS system options in SAS configuration file is as follows:
- SAS system option *value*.

The hyphen is required to specify the system option value in the configuration file.

The following examples illustrate how to modify and specify system options in the configuration file. In the configuration file, there are warnings on modifying the file. It is highly recommended to always save a copy of the original configuration file before modifying it.

1. **-work "C:\MySAS\SAS\"**
   This will change your default work folder to C:\MySAS\SAS.

2. **-register "EXCEL" "excel.exe"**
   This will add the Microsoft Excel application to the Tools menu in the main SAS window. It is useful when you want to open Excel from SAS. Programatically, you can use: **x 'start excel'** or use system function: **system('start excel')** within a data step.

3. **-altlog C:\MySAS\logs\mylog.log**
   This will capture a copy of log to "C:\MySAS\logs\mylog.log".

4. **-sortsize 8m**
   By default, this option is set to 2 megabytes. If your machine has more than 12 MB of physical memory and you are sorting large data sets, setting this option to a value between 2 MB and 8 MB may improve performance.

   SAS Technical Support Note SN-002527 also provides a formula to calculate the optimal sorting memory based on your PC RAM.

5. **-set sasautos** ("C:\MySas\MyMacros"
   "!sasroot\core\sasmacro"
   "!sasext0\assist\sasmacro"
   "!sasext0\eis\sasmacro"
   "!sasext0\ets\sasmacro"
   "!sasext0\graph\sasmacro"
   "!sasext0\or\sasmacro"
   "!sasext0\qc\sasmacro"
   "!sasext0\stat\sasmacro")
   This adds your own macro library path "C:\MySAS\MyMacros" to "sasautos" which enables you to call your macros directly in any session.

6. **-sasinitialfolder c:\MyProject\project1\**
   This specifies the path "C:\MyProject\project1\" as the current working folder and the default folder for the **Open** and **Save As** dialog boxes when SAS initialized.

   Omitting the path and using -sasinitialfolder ".", specifies the default folder for the **Open** and **Save As** dialog boxes to the current working folder which, by default, is the folder designated by the **sasuser** system option in the SAS configuration file. It is displayed in the status line at the bottom of the main SAS window.

IV. Using Autoexec.sas

Compared to the configuration file, autoexec.sas differs in two ways:

1. Autoexec.sas is not required for SAS to run. If there is a copy of autoexec.sas in the same folder as sas.exe, SAS will automatically execute the autoexec.sas right after SAS initialized. You can see the execution information in the log after the initialization information.

2. Autoexec.sas uses SAS statements just like any other SAS program. It is useful to have some pre-setup SAS statements in the autoexec.sas, e.g., assigning a commonly used library, formats or macro variables, setting up your default options for reporting, or running a sign on program at start up etc.

CONCLUSION

There are many ways to customize a SAS session to your preference. Using toolbars and keys, and modifying the configuration file and autoexec.sas are basic and feasible ways to customize. You can use different methods to accomplish the same task. Without any advanced technique, these basic customization tools can meet most of your customization needs.

REFERENCE

SAS Online Doc V8

TRADEMARKS

SAS[®] and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

Other brand and product names are registered trademarks or trademarks of their respective companies in the USA and other countries.

CONTACT INFORMATION

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

Xiaohui Wang
sas Certified Professional Version 8
Merck Co. & Inc.
RY34-A320
P.O. Box 2000
Rahway, NJ 07065
(732)-594-2311
xiaohui_wang@merck.com

Beilei Xu
Merck Co. & Inc.
RY34-A320
P.O. Box 2000
Rahway, NJ 07065
(732)-594-9980
beilei_xu@merck.com