Paper CT03

SAS Intellisense - Making sense of SAS's Intelligence

Deepak Ananthan, Zifo RnD Solutions, Chennai, India
Sachin Bharadia, Zifo RnD Solutions, Chennai, India

ABSTRACT

SAS® provides a lot of options in the default ToolBar like a single button click for running the programs, saving the program or opening a program. This paper will look at creating more such buttons and also look at using the command bar provided by SAS® to accelerate development time. There are a lot of routine tasks that the developer performs and if those can be provided as customized options that would greatly improve the productivity. This paper will look into creating some of the shortcuts: 1. Most of the times developers will want to open the last dataset that was created. 2. Clear the log every time we submit a program 3. Create shortcut to clear the WORK library 4. Use the command bar to open a dataset, edit it, hide unwanted columns, sort the data and do a lot more things by just typing basic commands.

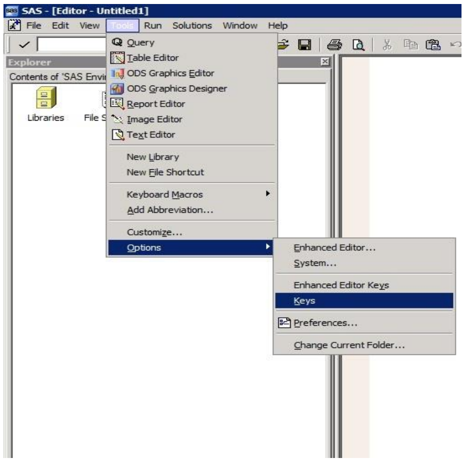
INTRODUCTION

SAS® is an ocean. It is vast, it is tranquil, it is mysterious to the uninitiated and like most oceans of our world a major portion of SAS still remains uncharted. It provides a mind boggling number of options, features, uses, most of which are yet to be charted even by the most skillful sailors of the industry. It is natural even for the masters to stick to the known, always sailing through charted territory even if it means to take a circuitous route. We get caught up in writing macros, learning concepts, implementing algorithms all of which are aimed at making us elegant, efficient enterprises we disregard the mundane tasks involved around programming with SAS as trivial.

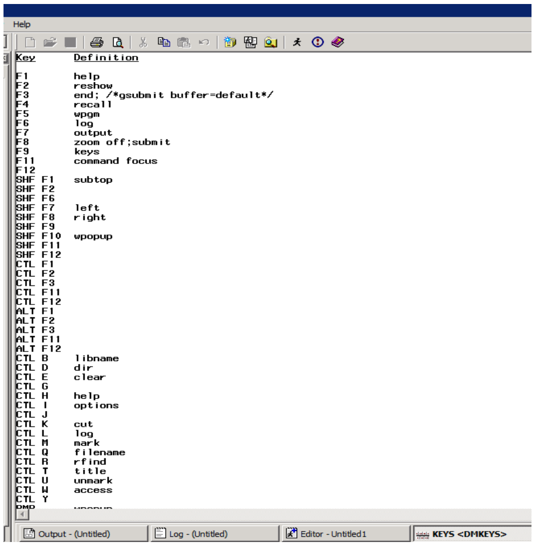
We wanted to find out what was the most frustrating part of learning to program in SAS among the members of our organization, and we were surprised to find out, it wasn't understanding macros, or transposing datasets, rather it was all the basic, repetitive tasks that really made them want to cut and run. This paper sought to give relief to all those people who've been desperately wanting to release their sails and set out and do the thing they've always wanted to do, program without pain.

ONE TOUCH RESPONSE

Have you ever written a long stretch of code, executed it and felt the excitement of looking at your output, only to go through a labyrinth of libraries, struggling to find the dataset you just created and losing the afore-found excitement in the process? This is where we see the help of the "KEYS" customization in full glory. What if you could, at the press of a single key execute your program and open the last created dataset? How about we raise the stakes higher and say you can clear your existing log, results window, submit the program, execute it AND open the last created dataset? If there was anything worse about programming in SAS, it is the need to clear the log or get sucked into blades of syntax errors and uninitialized notes. We went ahead and created a customization just for this.
The KEYS option in the SAS windowing environment enables us to achieve this feat.



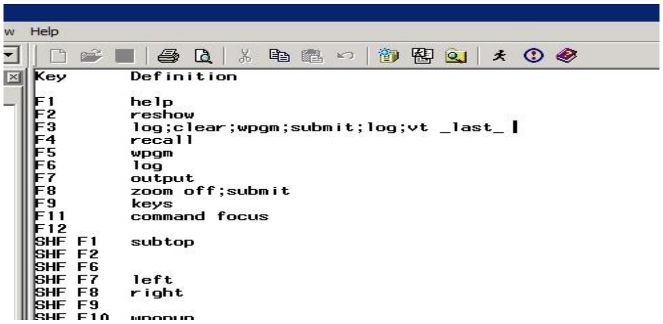
The KEYS option helps in assigning a set of functions for each of the Function Keys on the Keyboard. The SAS default KEYS settings can be found by going to the TOOLS->OPTIONS->KEYS. KEYS can also be accessed by pressing the F9 key from any of the various Windowing environments that SAS provides. This is the Default setting by SAS and can be changed to any of the allowed keys in the KEYS Option screen.



```
Key        Definition

F1         help
F2         reshow
F3         end; /*gsubmit buffer=default*/
F4         recall
F5         wpgm
F6         log
F7         output
F8         zoom off;submit
F9         keys
F11        command focus
F12
SHF F1     subtop
SHF F2
SHF F6
SHF F7     left
SHF F8     right
SHF F9
SHF F10    wpopup
SHF F11
SHF F12
CTL F1
CTL F2
CTL F3
CTL F11
CTL F12
ALT F1
ALT F2
ALT F3
ALT F11
ALT F12
CTL B      libname
CTL D      dir
CTL E      clear
CTL G
CTL H      help
CTL I      options
CTL J
CTL K      cut
CTL L      log
CTL M      mark
CTL Q      filename
CTL R      rfind
CTL T      title
CTL U      unmark
CTL W      access
CTL Y
RMB        wpopup
```

We came across a simple piece of code to reassign the submit key (F3) in the KEYS function to clear the log, submit and execute the program, print the log and open the last dataset created.



```
Key        Definition

F1         help
F2         reshow
F3         log;clear;wpgm;submit;log;vt _last_
F4         recall
F5         wpgm
F6         log
F7         output
F8         zoom off;submit
F9         keys
F11        command focus
F12
SHF F1     subtop
SHF F2
SHF F6
SHF F7     left
SHF F8     right
SHF F9
SHF F10    wpopup
```

Another minor modification that can be made to the KEYS option is to include `ODSRESULTS;CLEAR;OUTPUT;CLEAR;` which would clear the results and output window of the windowing environment.

There are multiple ways to redefine the default keys in SAS. To be precise there are 4 ways by which one can redefine a key definition in SAS. One way is what we've seen above to go to the KEYS option from the windowing environment and editing the definitions here.
For people aversive to the use of a mouse in their daily activities, SAS provides the option of using the ***KEYDEF*** command. The Syntax goes like:

```
        keydef keyname <command|~text-string>
```

The KEYDEF command must be given inside the Display Manager(DM) statement. Display Manager statements are an ocean within themselves and are wonderful tools that aid in customizations and ease of access in doing repetitive tasks in SAS.

```
        dm "keydef F5 'submit'";
```

This statement will redefine the KEYS mapping for the F5 key from the Default WPGM (most recent Enhanced Editor window) command to the SUBMIT command

There are two other ways to change the definition of a defined KEY, one involves RESOURCEHELPER to generate SAS resource specifications that can be used to redefine the keys. Another involves defining SAS.keyboardTranslations and SAS.keysWindowsLabels in the resources files. These options are beyond the scope of our context.

Keys are an essential part of our programming and the fact that minor tweaks to the KEYS function can improve our programming experience vastly was a profound discovery. The possibilities of what can be given in KEYS command are only confined to the imagination of the programmer and the 80 character restriction per key.

SENSE AND SIMPLICITY

Have you ever executed your program only to find the following message stare at you in the log?

> ***"You cannot open WORK.TEMP.DATA for output access with member-level control because WORK.TEMP.DATA is in use by you in resource environment ViewTable Window."***

It would probably drive even the most sane of us to the brinks of our sanity.

This happens every time we execute a code that uses/creates a dataset that is already open in our ViewTable Environment.

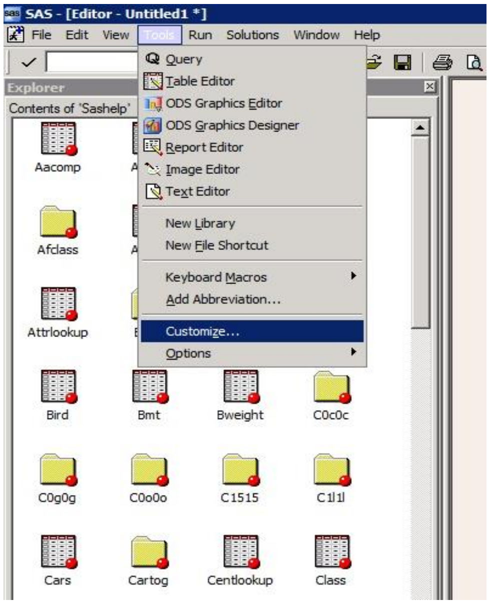
What if we told you the answer too was facing you right in the face every time you had SAS open?

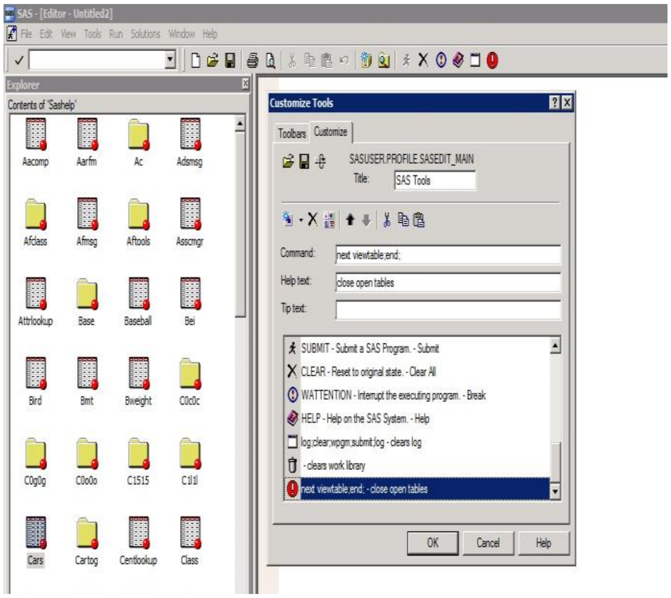
What if you could, at the click of single button close a dataset that is open?

The button can be placed right at the focus of your vision, in the Toolbar.

Toolbar customizations are one of the least explored features of SAS, mainly because of the lack of explicit literature as well as awareness on the endless possibilities within the same. Most of us like to tweak our working environment, add a little bit of our personal touch to our immediate setting, right from having a plant at our desk to having a style with which we code. An extension of this basic human need for the personal touch is the SAS toolbar customization. SAS contains an inherent fluidity in allowing freedom in the ways one can come up with customizations.

To create a custom button on the toolbar, go to the Tools->Customize option and click on the Customize Window.





SASUSER.PROFILE.SASEDIT_MAIN

Title: SAS Tools

Command: next viewtable;end;

Help text: close open tables

Tip text:

SUBMIT - Submit a SAS Program. - Submit
CLEAR - Reset to original state. - Clear All
WATTENTION - Interrupt the executing program. - Break
HELP - Help on the SAS System. - Help
log;clear;wpgm;submit;log - clears log
 - clears work library
next viewtable;end; - close open tables

In the Command line in the Customize window, enter the command for the functionality of the button that you want . The Help and the Tip Text can be used to give a basic definition and ease of use texts for the button you have created.

Coming back to our drive-to-the-brink-of-sanity problem, all we have to do is to give the following piece of code to the Command field in the Customize window:

```
        next viewtable;end;
```

Once you've created this button, it would essentially close an open dataset. Now we wanted to take this one bit further and tried to see how many datasets we could close at the same time. So we spammed the afore mentioned piece of code into the command bar until it could not take anymore. We found out that the maximum length that the field could accept was 500. We realized that we could close 25 open datasets at the same time! Of course we created this as a button.

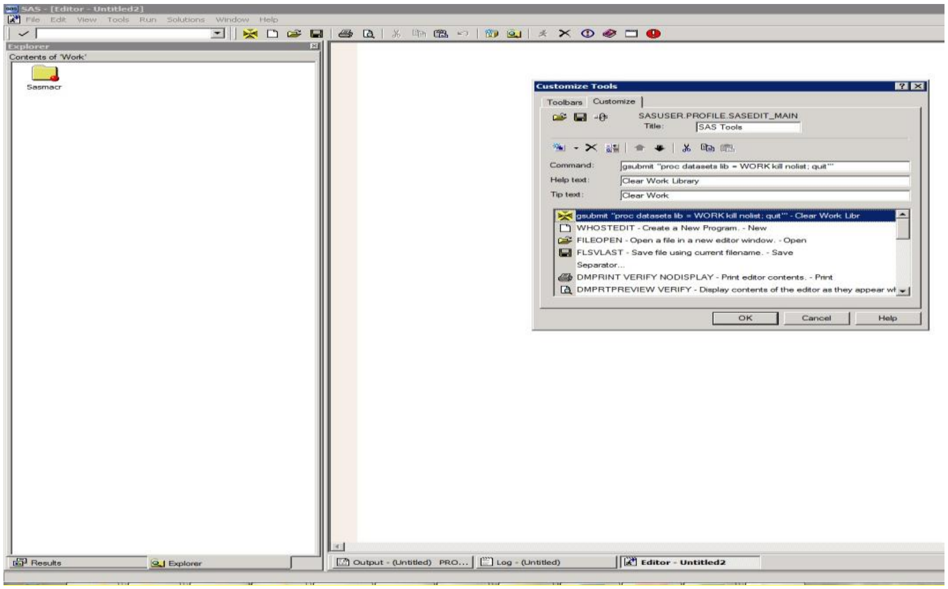
Clearing Entire Libraries

An obvious question that stemmed out of being able to clear 25 open datasets at the same time was, is it possible to clear an entire library?

Well, has SAS ever said "NO" to us? It only ever says "wait for it in the next version". Lucky for us, the current version of SAS by itself is able to solve our problem.

Repeat the same process mentioned above to create a button and in the command field, give the following piece of code:

```
        gsubmit "proc datasets lib = WORK kill nolist;quit;"
```



SASUSER.PROFILE.SASEDIT_MAIN
Title: SAS Tools

Command: gsubmit "proc datasets lib = WORK kill nolist; quit""
Help text: Clear Work Library
Tip text: Clear Work

gsubmit "proc datasets lib = WORK kill nolist; quit"" - Clear Work Libr
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 wh

The WORK library faces the brunt of our programming wrath, storing largely unrefined datasets that pushes the boundaries of our computational capabilities and the need to call a piece of code that clears the library is essential for efficiency and basic computational decency. Now if this can be achieved at the click of a single button who wouldn't want that.

Even though the default SAS WORKTERM option deletes the datasets present in the WORK library at the end of each SAS session, this button gives us control to have a clean house at our whim.

These customizations are simple, effective, easy to use and accessible to all. If you could identify common problems across your organization that require customization and come up with buttons on your own, it would be highly beneficial to the members of the organization in saving time and improving efficiency.

Command Bar

We’ve looked at closing open datasets and opening created datasets, now how about seeing an added bonus of looking at datasets like how they were intended to be viewed?

The most common feedback from members of our organization was that it takes just as long to look at the data that they just programmed for, as it takes to actually program. Upon further investigation we came to know that using the where filter option and using the variable drop down menu to filter the dataset, an essential feature of SAS created to aid us in looking at data instead of writing where clauses inside DATA steps or using PROC SQL to filter data actually becomes one of the key areas of loss of enthusiasm. Instead of right-clicking on the datasets, using the where option and filtering based on your needs, the entire process can be done by without ever touching the mouse but also without writing actual programs in the Editor Window.

This is where the COMMAND bar in SAS comes to the rescue.

The COMMAND bar can be found at the top left of your Windowing Environment on your Toolbar. The COMMAND bar accepts a large number of functions as input and aids in a myriad of functionalities within the Windowing environment, right from viewing, filtering and looking at data to printing a file.

The COMMAND option can also be accessed from the COMMAND line in the various SAS windows by typing the “PMENU OFF” option in the SAS COMMAND bar. This option enables the COMMAND line as the first line of various windows including editor, results, output, log windows.

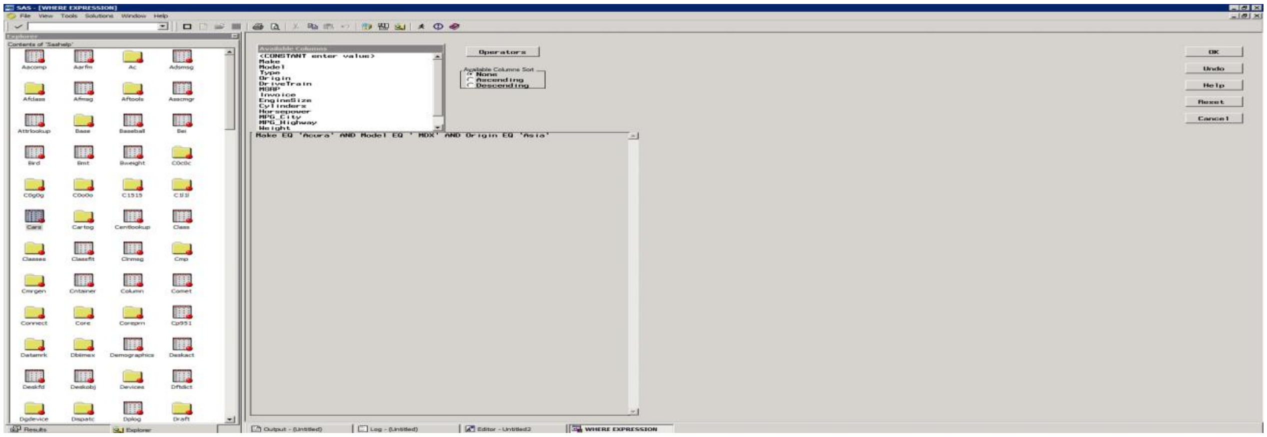
The exhaustive list of every SAS command available to us can be identified from the SAS COMMAND REFERENCE from the Help and Documentation file given with the software.

Each windowing environment has its own set of applicable commands that work only within that environment.

Care must be taken to understand which commands work within which environment before executing it as giving a inapplicable command for a given environment would not return anything and could lead to confusion.

Moving on to the exciting part of the COMMAND option, it can basically be used as a fact checker for our dataset by using the vt(viewtable) option as a command to open any dataset that is applicable in the current session. Once a dataset is available with the current session, it can be filtered using the COMMAND bar.

The usual way to go ahead with filtering a given dataset is to use the where pop up option in SAS and filter using the drop down parameter option.



```
Make EQ 'Acura' AND Model EQ ' MDX' AND Origin EQ 'Asia'
```

This function has a variety of features that include being able to have multiple filter conditions, looking up distinct values of the variables and so on. But the problem arises in dealing with the rudimentary User experience around which this whole feature has been developed. Now instead of using this to filter, we can carry out the exact operation using the COMMAND bar by giving the where condition inside the bar.

Consider the dataset SASHELP.CARS in the SAS default library. You can just type out the where condition like how you would do inside a data step. The command

```
        where MAKE eq “Acura” and MODEL eq “MDX”
```

would return the dataset filtered as specified when you give the command in the view table window.

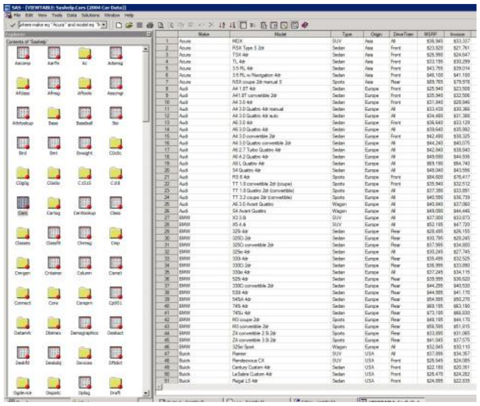
Moving forward on this option, suppose you want to only look at certain columns from the given dataset, giving the COLUMN command would display only the specified columns.

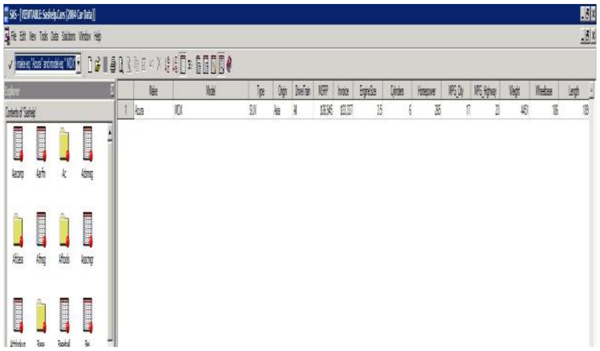
```
        COLUMN 'SEX AGE'
```

Similarly to hide a number of columns, giving

```
        Hide 'SEX AGE'
```

would hide those columns. Caution must be exercised while giving these commands on the applicable windowing environment and integrity of the command given as it does not throw any show issues explicitly.





Entire books can be written on COMMAND bar functionality but we went ahead and consolidated some of the most useful functions based on popularity and possibility.

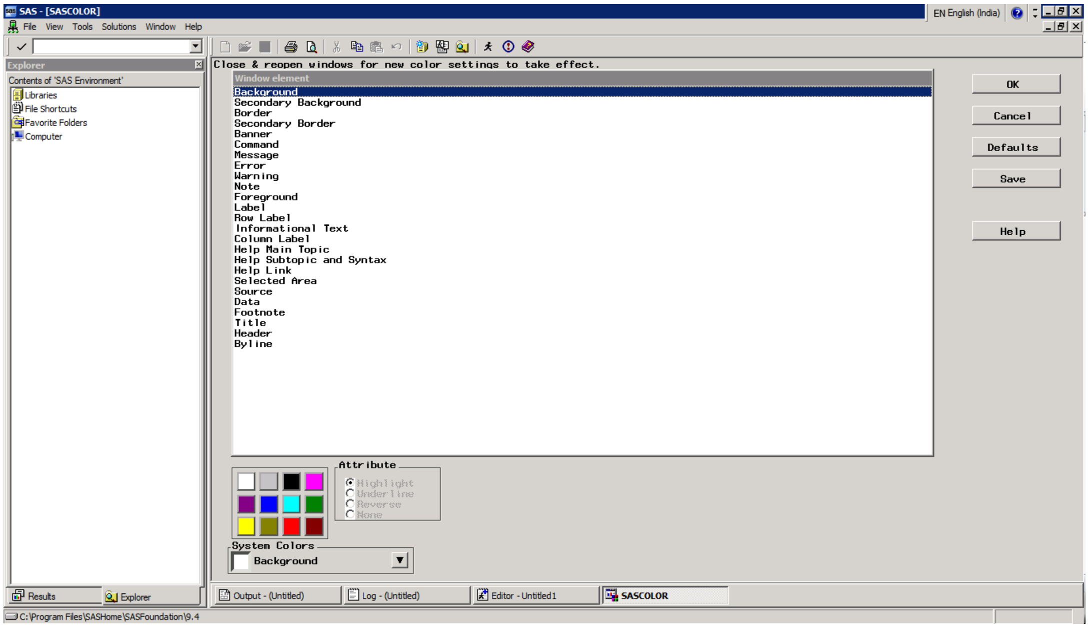
Dataset accessibility and viewing were looked at some of the most time consuming activities involved while programming with SAS outside of actually writing code.

VT, HIDE, COLUMN, UNHIDE, SORT, DMEXPOPTS are all basic commands that aid in dataset viewing and accessibility.

Options like ODSDOCUMENTS, ODSRESULTS, DMPRINT, DMPRTSETUP, ODSTEMPLATES all pertain to the outputs from SAS.

The PROGRAM, PGM, DLGFONT, SASCOLOR are all Editor Window options that are essential in pleasing the aesthetic sense of the programmer.

WDOCKVIEWRESIZE, WDOCKVIEWMINIMIZE, WDOCKVIEWRESTORE, WSCREENTIPS are all window options that can help you get a better look at how you view your SAS window.



Close & reopen windows for new color settings to take effect.

Window element
Background
Secondary Background
Border
Secondary Border
Banner
Command
Message
Error
Warning
Note
Foreground
Label
Row Label
Informational Text
Column Label
Help Main Topic
Help Subtopic and Syntax
Help Link
Selected Area
Source
Data
Footnote
Title
Header
Byline

Attribute
● Highlight
○ Underline
○ Reverse
○ None

System Colors
Background

Finally, our least and most favorite option, the BYE option that closes the current SAS session.

The COMMAND bar is one of the least explored and most useful feature of SAS that exponentially improves speed and efficiency of a programmer, instilling a sense of confidence and power in the Programmer.

CONCLUSION

We've only scratched the surface of a very deep and adventurous ocean. Customizations are only limited to the imagination of the programmer. Going by the quote "What is one man's toy is another man's tool", you can look at customizations and automation of repetition as merely an added feature of your working environment or you can look at it as an essential tool of effectiveness. Now most of the options that have been mentioned here may seem to save trickles of time at any given instance, but if you could save 20 seconds for every 10 minutes of programming it amounts to saving hours and hours for each person within the lifetime of a project. Extrapolating that even half the members of an organization gives fruition to the saying "An hour saved is an hour earned". Sensible options and intelligent features make SAS one of the most advanced yet unexplored software of our times.

REFERENCES

RECOMMENDED READING

CONTACT INFORMATION

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

Author Name: Deepak Ananthan
Company: Zifo RnD Solutions
Address: 21A, Anna Salai, Littlemount, Saidapet, Chennai, Tamil Nadu 600015
City / Postcode: Chennai / 600015
Work Phone: +91 44 43114002
Fax:
Email: Deepak.A@zifornd.com
Web: www.zifornd.com