A Little Goes a Long Way: Habits of the Efficient Project-Juggling SAS Programmer

Jonathan L. Moscovici, Gino Colavita, Jessica Chouinard
Justin Grondines, Patrick Pelletier
Quintiles Canada Inc., 100 Blv. Alexis-Nihon, St-Laurent, Montréal 20724

Abstract
When projects pile up and deadlines draw near, an effective SAS programmer often needs to become, among other things, an efficient SAS programmer. One point of view is that the adequate juggling of several projects can be broken down into main ideas such as clear and effective communication amongst collaborators, task automation and proper organization. These keys concepts will be outlined and exemplified using real code and techniques, facilitating the quest for high quality deliverables at breakneck speeds.

Key Words: SAS, Project Management, Statistical Programming, Efficient

1. Communication - Recipe For Success

A successful SAS programming project can be viewed as being a combination of two main elements. The first would be effective communication through proper email protocol, and the use of tracking documents & programming plans. The second consists of task automation through the use of bath scripting, SAS Macro and taskbar customizations. These two elements result in better organization and time management, leading to increased efficiency, cost reduction and an optimistic outlook on the handling of multiple projects.

1.1 Communication – Proper Email Protocol
Time is often a scarce resource, so it helps to learn to be frugal with it when possible. Many times, efficiency can just be the sum of small actions. Good habits don't need to take any longer than otherwise, so it's worth adopting them when possible. Since email is a popular communication tool amongst collaborators, it's a natural starting point for efficiency analysis. Some useful suggestions include always starting email subject lines with the project code/ID, to facilitate searching for correspondence later, and giving immediate context to a message (very useful when working on multiple projects), as well as including a meaningful subject afterwards. Furthermore, it pays to attempt to be succinct and organized with information in a message (e.g. bullet points, highlights, bold fonts, etc). Lastly, ensuring that all team members are copied in the message ensures that information is properly shared (proper labelling and organization of the messages can offset the inconvenience of having too much email).

1.2 Communication – Tracking Documents

A very useful tool for communication amongst collaborators is the tracking document. This technique enables task sequencing, better communication, and proper bookkeeping for larger projects through the creation of a shared spreadsheet containing all project elements to be contributed (data sets, tables, analyses, documents, etc.) along with their currently assigned team members and completion status.

	A	B	C	D	E	F	G	H	I	J	K	L
1						Macro				QC		
2	Run Order	Type	Derived Dataset	Overall Status	Spec Init	Name (if applicable)	Program Name	Init	Final Date Dataset	Init	Status	Comments
3	1	Derived	ADSL		SV		ADSL.sas	ML	27-Jun-2013	JLM	Completed (no issues)	
4	2	Derived	ADAE		SV		ADAE.sas	MPF	04-Jul-2013	JLM	Completed (no issues)	
5	2	Derived	ADEX		SV		ADEX.sas	JG	04-Jul-2013	JLM	Completed (no issues)	
6	2	Derived	ADDTH		SV		ADDTH.sas	JC	04-Jul-2013	JLM	Completed (no issues)	
7	3	Derived	ADOR		SV		ADORC.sas	CLN	04-Jul-2013	MB	Completed (no issues)	

Figure 1: Tracking Document Template

The tracking document is a live and dynamic way for organizational information to be centralized and easy to access. This leads to less time spent catching up on project status, while ensuring proper recording of major project milestones (less digging through email).

2. Automation

2.1 Automation - SAS Icon Customization

The SAS system allows users to customize buttons on the toolbar. Most users are aware of this, but make surprisingly little use of it. One great suggestion is to add icons capable of searching for specific keywords in the program log.

Figure 2: Custom SAS Toolbar Icons

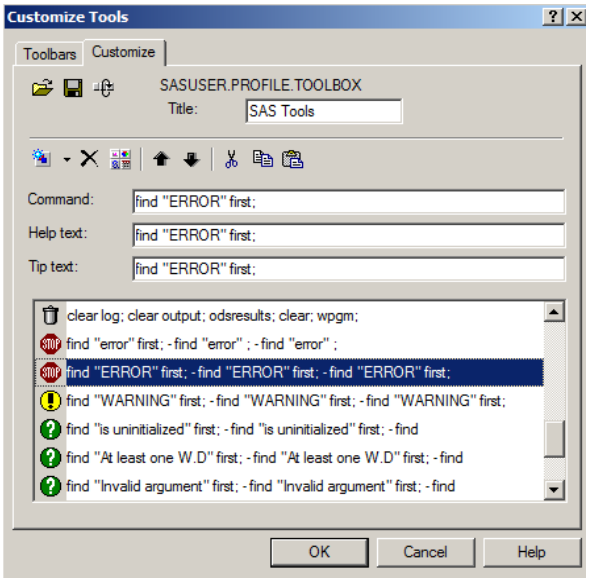

Figure 3: Custom SAS Toolbar Icon Manager

The above examples in Figure 3 tell SAS to begin searching from the top of the file, in order to catch the first instances or keywords such as "error" or "warning" that programmers will definitely not want to miss. Making a habit of always checking program logs using these customized buttons can ensure clean code.

2.2 Automation - SAS Keyboard Macros

The SAS system allows users to create keyboard macros that can save a great deal of code. By storing usual procedure keywords into macros, it is possible to use a shortcut key to call them. To create such macros, enter **Tools→Keyboard Macros→Record New Macro** and type the desired text. For example, if one wanted to assign a keyword for Proc Freq, one could type:

```
Proc Freq data=;
Tables / list missing;
Run;
```

Then select "Stop Recording". Once recorded, these macros can be renamed, modified and easily managed.

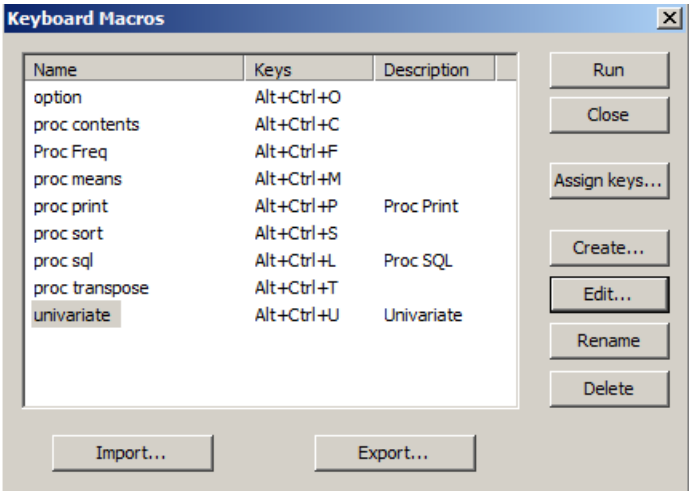

Figure 4: SAS Keyboard Macro Manager

2.3 Automation - Project Setup Program

A good way to avoid repetitive coding, such as formatting and attributes common to all programs in a project, is to create a central setup program that sets global options, macro variables, formats and library names for a project. This program can then be referenced in all other programs, avoiding the need to re-code all of these specifications. Here are some examples of tasks that can be performed globally:

2.3.1 Set Global Options

```
Option ls = 250 ps = 68 nodate Nonumber validvarname = v6
VALIDFMTNAME=FAIL formchar      = '|_---|+|---+=|-/\<>*' bufno=500
sortsize=50M compress=NO lrecl=1500 nobyline;
```

2.3.2 Define Global Macro variables

```
%LET root=C:\temp; %let date=20130804; %let _client=Client 1;
```

2.3.3 Define Libnames

```
libname sdtm     "&root.\data\&date"     access=readonly;
```

2.3.4 Macros and Formats

```
/* Include Formats*/
      %inc "&root.\Formats\format.sas";
/* Include Global Compiled Macros*/
      libname mylib 'C:\SasData\SASMACRO\compiled\20130804';
       options mstored sasmstore=mylib;
/* Access Project Macros */
      options mautosource MRECALL  sasautos=("&root.\Macros\",
      sasautos);
```

2.4 Automation - DOS Batch Files

A common task for today's SAS programmer is to run a series of SAS programs in a particular order. One can save a lot of time and trouble by programming a batch file that defines that files to be run and their order.

1. Simply **create a *.BAT file (using notepad or UltraEdit, for example)**
2. You need to know where your **SAS.exe** is located.
   The command line will look like this

   C:\…\"9.2(32-bit)"\Sas.exe -SYSIN c:\temp\my_sas_pgm.sas -nologo -batch -set jobpath …\"9.2(32-bit)"\SASV9.CFG

   SYSIN file-specification: Specifies to start SAS and submit the file in batch mode. The value of file-specification must be a valid Windows filename.

   BATCH: Specifies that SAS use the batch settings of LINESIZE=, OVP, PAGESIZE=, and SOURCE.

3. Once the file is done, **simply double click on the file to launch it.**

References

SAS Institute Inc. 2011. SAS ® 9.2 Language Reference: Dictionary, Fourth Edition. Cary, NC: SAS Institute Inc

Contact

Jonathan L. Moscovici
Quintiles Canada Inc.
100 Alexis-Nihon Blvd
Suite 800
Saint-Laurent, QC H4M 2P4
(514) 855 5509
(514) 855 0992 (FAX)
Jonathan.Moscovici@quintiles.com