

## Paper 098-2011

## How to Drive Your SAS® Programs on Cruise Control

Teo Gamishev, SC Budget and Control Board, Columbia, SC

### ABSTRACT

This paper introduces a simple, robust and efficient system to automate almost any SAS® program or project, requiring submitting series of SAS programs. The reader will be exposed to the system's big picture and technical details on how to build and apply it.

When created and applied on the first project, the new system provided the following results:

- Programmer's time: reduced from 13 hours to 13 minutes.  
That is 60 times or 6,000% improvement!
- Programmer's efforts: from 9 to 3, on a scale from zero to 10.  
Three times or 300% improvement!
- Programmer's errors: reduced to zero.  
Chances for human errors are eliminated!
- Programmer's experience: from stressful to joyful.  
Immeasurable improvement in the quality of work and life!

The system continues to provide similar results on all of its subsequent applications to different complex projects and single programs.

### INTRODUCTION OF THE PROBLEM

The usual workload of a SAS programmer frequently includes resubmitting a previously created program or series of programs to generate an outcome and deliver it to the customer. This work usually requires changes of certain parameters like dates, dataset names, variables names, titles and footnotes. Most of the time these tedious duties are done under pressure to deliver high quality results ASAP. Surfing through the program(s) to update them properly, being under pressure, very often ends in frustrating results due to increased number of human errors like: skipping a line for an update, typos, wrong dates, title errors, not reported in the SAS Logs and the list goes on.

The system, described here, eliminates all of the above problems.

One of its numerous advantages is that there is no need to have a very detailed knowledge of the SAS programs to which the system is applied. The first application of this system happened during the very beginning of a new contract. At that time, only general impression of what the programs are doing was available.

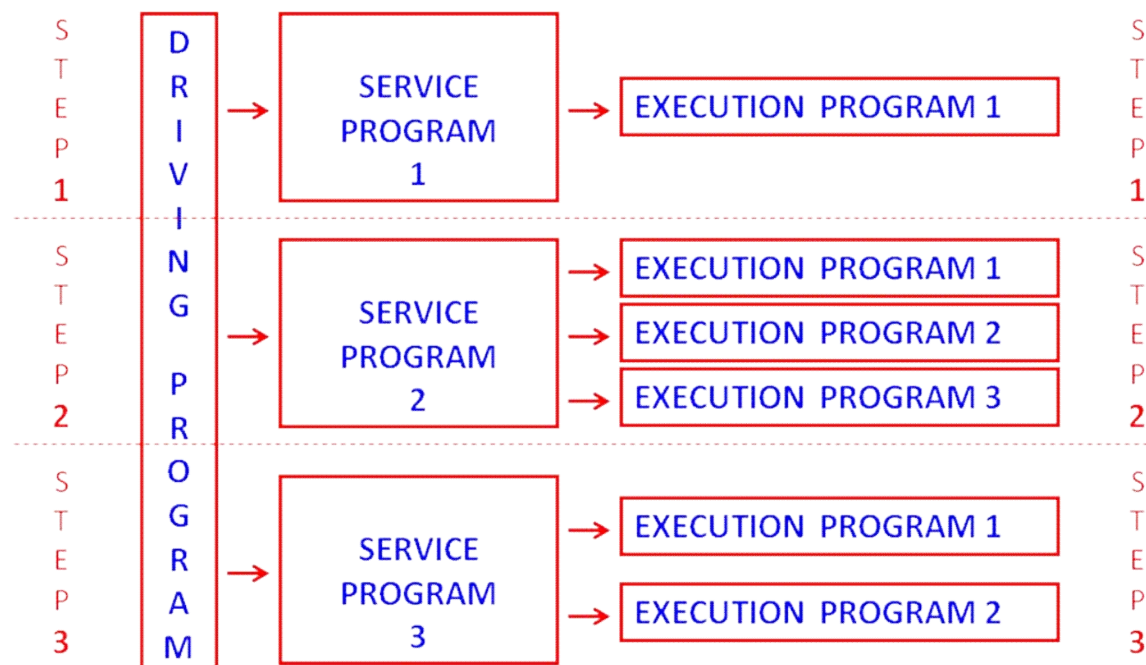
### SOLUTION

#### SAS ON CRUISE CONTROL SYSTEM'S BIG PICTURE

is described in the following four steps:

1. Find out the Main Steps in the process/project
2. Create a DRIVING Program, containing those steps
3. Create a SERVICE Program for each step
4. Create EXECUTION Programs for each step

Here is a flow-chart explaining how the system works:



**Figure 1**

Explanation of the Four Steps:

1. Find out the Main Steps in the Process.

A typical SAS project will contain some if not all of the following operations:

- Data Downloading
- Data Manipulations
- Generating Outcomes(Analyses, Modeling, Forecasting)
- Delivering Outcomes
- Data Archiving
- Email Management

2. Create a DRIVING Program, containing steps relevant to your project:

The Driving program is the WHEEL in the SAS on Cruise Control system

This is a SAS code where the programmer:

- Pushes the button "to select the desired speed" by entering one or several macro variables,
- Controls the entire process by triggering the Project' Steps, one by one or all at once, when applicable.

The DRIVING program is the only program to handle with ones the system is properly built!

EXAMPLE: Driving Program

```

%LET DM=Jan2010;                                *Cruise Control PUSH BUTTON;

*****;
***          STEP ONE                          ***;
***    DOWNLOADING MONTHLY EXTRACTS           ***;
*****;

%include "C:\Programs\DataDownloading.sas"; * Service Program1;
  
```

```

*****;
*** 1. Monitor the downloading process          ***;
*** 2. Check the file size                     ***;
*****;
*****;
***                                END of STEP ONE          ***;
*****;

*****;
***                                STEP TWO                  ***;
***                                DATA MANAGEMENT, DATA MANIPULATIONS          ***;
*****;

%include "C:\Programs\DataManagement.sas";* Service Program2;

*****;
*** 1. Check the Log for ERRORS                ***;
*** 2. Check the updated file size            ***;
*****;
***                                END of STEP TWO          ***;
*****;

```

### 3. Create a SERVICE Program for each step

The SERVICE programs perform “the fine tuning of the SAS engine”

They are SAS programs where:

- All of the required parameters for that particular Project Step are generated and stored into macro variables: dates, datasets names, variable names
- All new folders and libnames are generated

#### EXAMPLE: Service Program

```

* Creating global macro variables;
data _null_;
  dmfirst=intnx('month', "01&DM"d,0);
  dmlast=intnx('month',dmfirst,0,'end');
  call symput('DMYYYYMM', put(dmfirst, YMMN6.));
  call symput('YDM', trim(left(put((dmfirst), YEAR4.))));
run;

%put _user_;      * Checking the new macro variables;

* Creating a folder for the Stat Pack Reports;
filename papka PIPE "md "&thepath\&DMYYYYMM"";

data _null_; infile papka; run;

* Triggering the programs for creating the reports;
%include "&thepath\Programs\XLAllCasesReportsNEW.sas";

```

### 4. Create EXECUTION Programs for each step

The EXECUTION Program represents “the running engine” in the SAS on Cruise Control system.

This is a SAS program where part or all of the Project’ Step outcomes are produced.

#### EXAMPLE: Execution Program

(This is just a part of the program, showing how the Service program macro variables are used.)

```

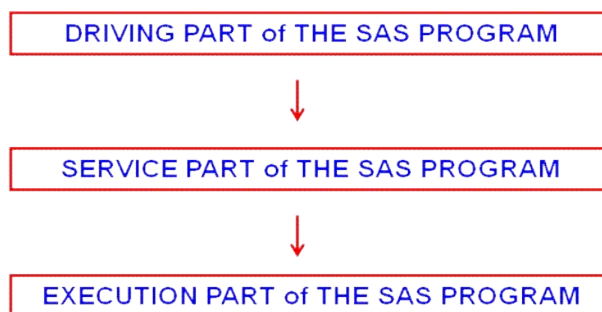
proc sql;
  create table      TMONTH&CY as
  select distinct  a.CASE, a.PGM, a.BENMONTH,
  from  chip.BENFY&BYDM a
  where a.PGM=&FI AND
        a.ISSDATE LE &DMLD  AND
        a.BENMONTH=&DMONTH ;

quit;

```

Applying the System into a small project:

Here is a flow chart showing how to apply the system to a single SAS program:



**Figure 2**

Explanation of the flow chart above:

The SAS on Cruise Control System can be applied to even a single SAS program in order to use the system's benefits.

The top part of the program should contain the elements of the DRIVING program and mostly the 'Cruise Control Push Button', the macro variable(s). They are needed to generate all of the parameters in the SERVICE part. All efforts should be made to squeeze the number of those initial macro variables to one or two. Most of our projects need one macro variable, only. However, if a specific project requires 25 macro variables, for example, they should be there.

The next part of the program should perform all of the functions of the SERVICE program. Which means all of the required parameters to run the program properly should be generated.

At the end, the EXECUTION part should produce the desired outcome, using all of the macro variables from the SERVICE program.

The above format gives a simplicity and clarity to all of the ad-hocs requests a SAS programmer often has to respond. In addition, they can be easily recycled with minimum time and efforts in the future.

## SAS ENGINE TUNING TIPS

Initial development of the system will require finding solutions of unusual problems which may pop up along. Solving those problems is the fun part of the game, which will widen your skill and give you the courage to go to the end of the task.

Here is a list of the problems we found out how to solve directly from SAS:

1. Control the flow of the SAS program with macros
2. Create custom time periods like state and fiscal year
3. Create new dataset at the beginning of a new time period
4. Debug a variable in a dataset

5. Download customer's data hands free
6. Upload the outcomes to the client's site
7. Archive the data
8. Manage emails
9. Quality control on proper execution of all of the steps

Some of the above problems required immediate solution, before the system could come into operation. Some of them, which were not so crucial for the process, we manage to solve later. Those solutions may give a hint or at least inspire you into finding yours.

### DEVELOPING THE SYSTEM: STEP-BY-STEP GUIDE:

Applying SAS on Cruise Control system to Existing Project:

The easiest way to start with is to apply the system to an existing project as you can compare the results and see the benefits, as soon as the system is tuned properly. Before doing any changes, make sure that all of the programs you are about to modify are copied into new folder and properly renamed to avoid confusions in the future:

1. Identify the major steps in the process
2. Find out the what would serve as a PUSH BUTTON value to create all of the parameters you need to update the programs
3. Create the initial version of the DRIVING program with those steps and insert the PUSH BUTTON value into a macro variable

EXAMPLE: DRIVING Program

```
%LET DM=Jan2010;                                *Cruise Control PUSH BUTTON = Data Month;
```

4. Group the existing programs, according to the steps, outlined above
5. Create the SERVICE program for the first step by:
  - a. open the first EXECUTION program in the first step,
  - b. locate the first place where change needs to be made,

Example: EXECUTION program

```
data extract;
  set master;
  where '01jan2010'd LE issdate LE '31jan2010'd;
run;
```

- c. go to the SERVICE program and generate macro variables containing the values to be replaced

Example: SERVICE program

```
data _null_;
  dmfirst=intnx('month',"01&DM"d,0);          * data month first day;
  dmlast=intnx('month',dmfirst,0,'end');      * data month last day;
  call symput('DMF', put((dmfirst), DATE9.)); * macro variable first day;
  call symput('DML', put((dmlast), DATE9.)); * macro variable last day;
run;
```

- d. copy the macro variable from the SERVICE program into the EXECUTION program

Example: EXECUTION program

```
data extract;
  set master;
  where "&DMF"d LE issdate LE "&DML"d;
run;
```

- e. repeat the above for all of the necessary changes in the first EXECUTION program
- f. repeat the above for all EXECUTION programs in the first step
6. Create the SERVICE programs for the next steps by repeating the above
7. Select and submit each step from the DRIVING program, remove the errors and make sure the outcomes are generated properly
8. Incorporate as many quality control steps as you need to make the entire process of verification simple and easy.
9. Compare the outcomes with the ones from the old process.
10. Once you are completely confident that the new system works, move it into production area.

Applying SAS on Cruise Control system to a New Project:

The only difference between applying the system into existing and new projects will be that the EXECUTION programs have to be written and brought to work in a regular way, first. Once this is done, you can apply the steps for Existing Projects above. However, having the experience from former applications of the system will inevitably give you the insights of what and how to be done for easy handling of those steps.

## CONCLUSION

This system is a very powerful tool for introducing significant improvements in any area of application of SAS programming, if applied properly.

Wrapping every SAS program into the system becomes an invaluable skill, which gives you tremendous advantage on the work place and boosts your confidence in what you do.

The time one gains from applying the system, can be used to advance far ahead into your area of knowledge, expertise and career.

Any company manager can see how the system can boost the productivity, significantly, by applying the information shared in this short article.

## REFERENCES

Teo Gamishev, "How to Optimize Existing SAS® Programs for Generating Standard Reports",

<http://sasarticles.blogspot.com/2010/06/how-to-optimize-existing-sas-programs.html>

Teo Gamishev, "Affordable SAS® Tips", available at

<http://sasarticles.blogspot.com/2010/06/affordable-sas-tips.html>

Teo Gamishev, "Hands Free Data Transfer Using SAS®", available at

<http://sasarticles.blogspot.com/2010/03/hands-free-data-transfer.html>

Teo Gamishev, "How to Manage Emails from SAS®", available at

<http://sasarticles.blogspot.com/2010/03/how-to-manage-emails-from-sas.html>

## CONTACT INFORMATION

Name:	Teo Gamishev, M.S. in Statistics
Enterprise:	SC Budget and Control Board, Office of Research and Statistics
Address:	1919 Blanding Street
City, State ZIP:	Columbia, SC 29201
Work Phone:	803-898-9945
Fax:	803-898-9972
E-mail:	<a href="mailto:gamishev@hotmail.com">gamishev@hotmail.com</a>
Web:	<a href="http://sasarticles.blogspot.com/">http://sasarticles.blogspot.com/</a>

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. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.