"Govern the Clock": Adding the Run Date and Time of a SAS® Program to your LOG, LST, and RTF Output

Eric Elkin, ICON Clinical Research, San Francisco, CA

ABSTRACT

This presentation will focus on solutions for automatically documenting the actual time and date of program submission in the SAS log, listing, and RTF output. Three related problems will be addressed. First, the default date and time printed to the SAS output is the date and time that the SAS session was started, not the time that the program was actually run. Secondly, the date and time are not printed in the log. Finally, there are issues with the date and time that appear in the RTF file when it is created using the default SAS date and time output. The solution presented here involves the global OPTIONS statement and macro variables whose values are assigned using CALL SYMPUT.

INTRODUCTION

"I must govern the clock, not be governed by it" – Golda Meir

This paper addresses three issues related to the documentation of the program run date and time in the SAS log and output. First, the date and time are not printed at all in the SAS log, which would be helpful for documentation purposes. Secondly, the date and time that the SAS session was started is shown in the SAS output (i.e., listing window or .LST file), which is often not the time that the program was actually run to produce the output. When running multiple programs during the same SAS session it could be helpful to know the order in which the output was produced. Finally, the date and time are included in RTF output as a "current date/time" function. That is, when an RTF file is opened after it is created, the function resolves to the date and time the file is opened and not the original date and time that it was created. This can definitely cause documentation problems when reviewing results.

In order to address these problems in an automated and efficient manner, the solution presented here was developed using macro variables, CALL SYMPUT, and global option settings.

SOLUTION

The core of the solution is based on capturing the program's run date and time as macro variables. The DATE and TIME functions return the current date and time. The PUT statements shown below capture the values of the DATE and TIME functions into temporary variables. (These values can be formatted as the programmer chooses with the two formats shown below as examples.) Then, these values are assigned to macro variables using the CALL SYMPUT statements.

```
data _null_;

    tempdate=put(date(),weekdate32.);
    temptime=put(time(),hhmm5.);

    call symput("currdate",tempdate);
    call symput("currtime",temptime);

run;
```

The programmer can then use these macro variables in a title statement which provides text for the output (whether as listing window, .LST file, or RTF file). This text will not change the next time the RTF file is opened.

```
title "RUN DATE=&currdate. at RUN TIME=&currtime.";
```

Use the SYMBOLGEN global option so that the run date and time appear in the SAS log. SYMBOLGEN will resolve the `&currdate` and `&currtime` macro variables in the SAS log, thereby automatically writing the run date and time there. In addition, the default SAS date and time are no longer needed and can be excluded with the global option NODATE.

```
options nodate symbolgen;
```

For this solution, order matters. Therefore, the final code to solve all three problems is the following:

```
options nodate symbolgen;

data _null_;

    tempdate=put(date(),weekdate32.);
    temptime=put(time(),hhmm5.);

    call symput(“currdate”,tempdate);
    call symput(“currtime”,temptime);

run;

title "RUN DATE=&currdate. at RUN TIME=&currtime.";
```

CONCLUSION

This solution automatically, efficiently, and accurately documents the run date and time in the SAS log, listing, and RTF output.

CONTACT INFORMATION

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

Eric Elkin
ICON Clinical Research
188 The Embarcadero, Suite 200
San Francisco, CA 94105
Work Phone: 415-371-2153
Fax: 415-856-0840
Email: eelkin@ovation.org