

## Let the Code Report the Running Time

Zi Feng, University of South Florida

### ABSTRACT

A programmer/analyst may often find it necessary to know the exact start, stop, and elapsed time of a SAS code or a specific part of the code especially when it is executed through a scheduled job. SAS STIMER sometimes gives confusing information and manual calculations are still needed. This paper presents a tip on how to have the exact real running and elapsed time reported in the log. A macro utility is provided as an example of how this process can be streamlined and made flexible so that clear and customizable information can be reported.

### INTRODUCTION

It is often useful to know the exact start, stop, and elapsed time of a SAS code. SAS STIMER option writes CPU and real time information to the log. It however becomes less useful and confusing when one is interested in the elapsed time of larger components rather than a single step. It would be more helpful to have the flexibility to get the running time information for whichever specific part of the code without the need of manual calculation and have it automatically reported in the log.

### A MACRO EXAMPLE OF REPORTING THE RUNNING TIME

The current paper presents a simple idea to achieve this goal, which is to have SAS report the time at both of the starting and ending points of the specified code component and calculate the difference between the two time points as the elapsed time. An example is provided to show how this simple idea can be executed through a macro utility and how this process can be streamlined and made flexible so that clear and customizable information can be reported.

Below is a sample macro to accomplish this task. 'BEGIN' or 'END' (not case sensitive) can be passed to the parameter BGN\_END. %RunTime(BEGIN) needs to be placed right before the block of code to be monitored, which will write the exact start time. %RunTime(END) needs to be placed right after the block of the code, which will write the end time and the calculated elapsed time to the log.

```
%Macro RunTime(BGN_END);

%GLOBAL runbegin runend rundiff;

%If &BGN_END EQ %Then %Do; /* use BEGIN by default if no parameter value is
specified */
    %let BGN_END= BEGIN;
%End;

%If %Uppcase(&BGN_END) = BEGIN %Then %Do; /*report BEGIN time */
    %let runbegin = %Sysfunc(Datetime());
    %let runend = ;
    %let msg = NOTE: The specified block of code BEGAN AT %Sysfunc(Sum(&runbegin),
Datetimel6.);
%End; %Else
%If %Uppcase(&BGN_END) = END %Then %Do; /*report END time and calculate elapsed
time*/
    %let runend = %Sysfunc(Datetime());
    %let rundiff = %Sysfunc(Sum(&runend, -&runbegin));
    %let msg = NOTE: The specified block of code ENDED AT %Sysfunc(Sum(&runend),
Datetimel6.);
    %let msg = &msg....Elapsed time of (hh:mm:ss): %Sysfunc(Sum(&rundiff), Time8.);
%End;

%Else %Do; /*To print error message if other parameter value than BEGIN and END is
used*/
    Options Nosource;
    %let runbegin =%Sysfunc(Datetime());
```

```

%Let runend =%Sysfunc(Datetime());
%Put;%Put;
%Put ERROR- NOTE: Wrong parameter. Please use RunTime(BEGIN) or RunTime(END).;
%Put;%Put;
Options Source;

%goto SKIP; /*Skip to end of the macro*/
%End;

Options Nosource;
%Put;%Put;
%Put %Superq(msg); /*Print the running time message*/
%Put;%Put;
Options Source;

%SKIP: /*end of macro*/

%Mend RunTime;

```

## MACRO USAGE EXAMPLES

As seen below, %RunTime can be placed to include any steps one is interested in retrieving the running time information from.

```

%RunTime(BEGIN);
data test1;
  do key=1 to 86298000;
    length data $12;
    data=put(key, words12.);
    output test1;
  end;
run;
%RunTime(End);

%RunTime(BEGIN);
data test2;
  do key=1 to 86298234;
    length data $12;
    data=put(key, words12.);
    output test2;
  end;
run;
%RunTime(End);

```

The start, end, and elapsed time are reported in the log shown in the screen shot below. %RunTime(BEGIN) placed before the code block reports the start time. %RunTime(END) placed after the code block reports the end and calculated elapsed time. As seen, value assigned to parameter BGN\_END is not case sensitive.

```
561 %RunTime(BEGIN);
```

NOTE: The specified block of code BEGAN AT 01AUG13:08:53:57.

```
562 data test1;  
563   do key=1 to 86298000;  
564     length data $12;  
565     data=put(key, words12.);  
566     output test1;  
567   end;  
568 run;
```

NOTE: The data set WORK.TEST1 has 86298000 observations and 2 variables.

NOTE: DATA statement used (Total process time):

real time	50.64 seconds
cpu time	48.62 seconds

```
569 %runTime(End);
```

NOTE: The specified block of code ENDED AT 01AUG13:08:54:47....Elapsed time of (hh:mm:ss): 0:00:51

```
570
```

```
571 %RunTime(BEGIN);
```

NOTE: The specified block of code BEGAN AT 01AUG13:08:54:47.

```
572 data test2;  
573   do key=1 to 86298234;  
574     length data $12;  
575     data=put(key, words12.);  
576     output test2;  
577   end;  
578 run;
```

NOTE: The data set WORK.TEST2 has 86298234 observations and 2 variables.

NOTE: DATA statement used (Total process time):

real time	55.96 seconds
cpu time	48.40 seconds

```
579 %runTime(End);
```

NOTE: The specified block of code ENDED AT 01AUG13:08:55:44....Elapsed time of (hh:mm:ss): 0:00:57

By default, 'BEGIN' is assigned to BGN\_END if it is not specified.

```
581 %RunTime;
```

NOTE: The specified block of code BEGAN AT 01AUG13:08:55:45.

```
582 data test2;  
583   do key=1 to 86298500;  
584     length data $12;  
585     data=put(key, words12.);  
586     output test2;  
587   end;  
588 run;
```

NOTE: The data set WORK.TEST2 has 86298500 observations and 2 variables.

NOTE: DATA statement used (Total process time):

real time	52.98 seconds
cpu time	49.82 seconds

```
589 %runTime(End);
```

NOTE: The specified block of code ENDED AT 01AUG13:08:56:38....Elapsed time of (hh:mm:ss): 0:00:53

If a value other than 'BEGIN' and 'END' is assigned to BEG\_END, a note will be reported to the log to remind that only 'BEGIN' or 'END' is valid.

```
601 %RunTime(start);
```

```
NOTE: Wrong parameter. Please use RunTime(BEGIN) or RunTime(END).
```

```
602 data test3;  
603   do key=1 to 8629850;  
604     length data $12;  
605     data=put(key, words12.);  
606     output test3;  
607   end;  
608 run;
```

```
NOTE: The data set WORK.TEST3 has 8629850 observations and 2 variables.  
NOTE: DATA statement used (Total process time):  
      real time          4.85 seconds  
      cpu time           4.86 seconds
```

```
609 %runTime(stop);
```

```
NOTE: Wrong parameter. Please use RunTime(BEGIN) or RunTime(END).
```

## DISCUSSION

This paper represents a straightforward idea of reporting running time in the log for a specific component of a SAS program. The included sample macro utility may come in handy in monitoring running status and performance of a SAS job and provides a basis for a streamlined and flexible process so that clear and customizable information can be reported.

## CONTACT INFORMATION

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

Zi Feng  
College of Business, University of South Florida  
4202 E. Fowler Avenue  
Tampa, FL 33620  
zifeng@mail.usf.edu

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.