P93 - 25

Placing Dates in Your Titles: Do It Dynamically

Arthur L. Carpenter
California Occidental Consultants

ABSTRACT

Suppose you want your SAS® output to contain a date/time stamp. By default each output procedure places this information in the upper right corner on each page, this is a good start and is often 'good enough', but it has limitations. These limitations include:

- no choice in date format
- no choice in location
- date/time measured from the start of the current SAS session

It is not uncommon to want to overcome these limitations by placing the date (time or date/time) in the title. To be practical, we do not want to recode the TITLE statement before each execution of the program. Recoding is avoided by placing the value dynamically (the program itself determines the correct values). A dynamic title will capture and use the actual date/time at the time of execution and could even use the format of your choice. This means that your favorite programs can produce titles with an accurate date and/or time stamp, without any intervention on your part.

KEYWORDS

DATA _NULL_, macro function, %SYSFUNC, &SYSDATE, automatic macro variable

INTRODUCTION

You want the output from a SAS job to contain the current date. By default (when the DATE system option is on) a date/time stamp is automatically generated in the upper right corner of each output page. Usually this is acceptable, but not always. You can not select either the format of the date nor its position on the page. Worse yet the date and time reflects the start of the current SAS session (or start of the batch job). This date/time may not be accurate enough. If you want to control the format of the date, its location, or its accuracy, you will need a different approach.

For the purposes of this discussion the following examples concentrate on DATE, but they apply equally to TIME and DATETIME values.
Some users may try to calculate a date value in the DATA step, and try to insert the value directly into a TITLE statement. TITLE and FOOTNOTE statements, however are global statements, and even when included within the confines of a DATA step, values from variables cannot be placed in the title. Fortunately there are alternatives, and placing values in TITLE and FOOTNOTE statements is commonly done through the use of macro variables.

There are three common approaches to defining and placing macro variables in titles and footnotes. These approaches use:

- the automatic macro variable &SYSDATE
- macro variables created in a DATA step
- macro variables created using %SYSFUNC

USING &SYSDATE

The automatic macro variable &SYSDATE (there is also a &SYSTIME and &SYSDATE9) only overcomes one of the disadvantages noted above, you can control the location. The primary advantage is that values of automatic macro variables are set by SAS and are always available. To place the date in a title simply specify the variable name.

```
title1 "Using SYSDATE &sysdate";
```

The following title is produced:

```
Using SYSDATE 12APR00
```

The &SYSDATE macro variable is easy to use but cannot be formatted. It will always be in the form of ddmonyy. Also this date will, like the output date, reflect the start date of the current SAS session. SYSDATE9 is similar to &SYSDATE, however, it displays a four digit year.

USING THE DATA STEP

By using DATA step functions, the DATA step can be used to generate a macro variable that contains a current date value using a specified format. This method overcomes all three of the disadvantages mentioned above. Since we do not need to read or create a data set a DATA _NULL_ step is often used. In the following DATA _NULL_ step with an assignment statement and a CALL SYMPUT is used to create a formatted macro variable. This macro variable is then placed in the TITLE statement.

```
data _null_;
today = put(date(),worddate18.);
call symput('dtnull',today);
run;

title2 "DATA _NULL_ &dtnull";
```

The following title is produced:

```
DATA _NULL_     April 12, 2000
```

The WORDDATE18. format has produced the desired date, however since the date is right justified, there is a series of leading blanks. These blanks are easily eliminated through the use of the LEFT and TRIM (to help with centering) functions.

```
data _null_;
today =
trim(left(put(date(),worddate18.)));
call symput('dtnull',today);
run;

title2 "DATA _NULL_ &dtnull";
```

The following title is produced:

```
DATA _NULL_      April 12, 2000
```

USING THE %SYSFUNC FUNCTION

The two macro functions %SYSFUNC and %QSYSFUNC, greatly increase the list of functions available to the macro language by making available almost all DATA step functions. As a consequence they can substantially reduce the need for single observation DATA _NULL_ steps. The first argument of %SYSFUNC is the call for the DATA step function, and the optional second argument is a format that will be applied to the value returned by the function call.

```
title3 "Using SYSFUNC
%sysfunc(date(),worddate18.)";
```

The following title is produced:

```
Using SYSFUNC     April 12, 2000
```

The LEFT and TRIM functions can again be used to remove the space, however care must be exercised or a couple of problems can be encountered.

The first is that function calls cannot be nested within a %SYSFUNC. Fortunately this is rather easily handled because %SYSFUNC requests **can** be nested.

Secondly the resolved values of interior calls to %SYSFUNC are used as arguments to the outer calls. When the resolved value contains special characters (especially commas), they can be misinterpreted. The following revised TITLE3 will not work because the interior %SYSFUNC uses a formatted value which contains a comma.

```
title3 "Using SYSFUNC
%sysfunc(left(%sysfunc(date(),
  worddate18.)))";
```

After the inner %SYSFUNC is executed the result is:

```
title3 "Using SYSFUNC
%sysfunc(left(April 12, 2000))";
```

The LEFT function will see two arguments because of the comma (it is expecting exactly one argument), and the message 'too many arguments' is generated.

The %QSYSFUNC function can be used to mask special characters in the text string that is passed to the next function.

```
title3 "Using SYSFUNC
%sysfunc(left(%qsysfunc(date(),
  worddate18.)))";
```

TITLE3 from above becomes:

```
Using SYSFUNC April 12, 2000
```

Since the use of imbedded calls to %SYSFUNC can be a bit messy, the %LEFT macro function can be used instead.

```
title3 "Using SYSFUNC
%left(%qsysfunc(date(),worddate18.))";
```

TITLE3 from above becomes:

```
Using SYSFUNC April 12, 2000
```

SUMMARY

Through the use of the %SYSFUNC macro function, DATA step functions can now be used to dynamically place formatted date values into a title without the use of the DATA _NULL_ step or programmer intervention.

ABOUT THE AUTHOR

Art Carpenter's publications list includes two chapters in *Reporting from the Field*, three books

(*Annotate: Simply the Basics*, *Quick Results with SAS/GRAPH® Software*, and *Carpenter's Complete Guide to the SAS® Macro Language* and over three dozen papers and posters presented at SUGI, WUSS, and PharmaSUG. Art has been using SAS since 1976, and has served in a variety of capacities with user groups at the local, regional, and national levels.

Art is a SAS Quality Partner™ and SAS Certified Professional™. Through California Occidental Consultants he teaches SAS courses and provides contract SAS programming support nationwide.

AUTHOR CONTACT

Art Carpenter
California Occidental Consultants
PO Box 6199
Oceanside, CA 92058-6199

(760) 945-0613

art@caloxy.com
www.caloxy.com

REFERENCES

A further discussion of this topic and some other macros that deal with dates in titles can be found in the following books.

Burlew, Michele, *SAS ® Macro Programming Made Easy*, Cary, NC: SAS Institute Inc., 1998, 280pp.

Carpenter, Arthur L., *Carpenter's Complete Guide to the SAS ® Macro Language*, Cary, NC: SAS Institute Inc., 1998. 242 pp.

TRADEMARK INFORMATION

SAS, SAS Certified Professional, and SAS Quality Partner are registered trademarks of SAS Institute, Inc. in the USA and other countries.

® indicates USA registration.