

CC_05
Report Dates: %Sysfunc vs. Data _Null_
Bill Parman, Cigna®, Chattanooga, TN, USA

ABSTRACT

This paper presents the macro function, %Sysfunc, as an alternative to the use of the DATA _NULL_ data step for the purpose of creating and populating macro variables with date values used for the selection of data and titling of reports.

KEY WORDS

%SYSFUNC, DATA _NULL_, TODAY, INTNX, CALL SYMPUTX, and PUT

INTRODUCTION

Assumption: The reader is familiar with the SAS® functions: TODAY(), INTNX(), and PUT().

The evolution of storing dates in macro variables often goes from hard-assigning dates to macro variables with the %Let statement to the use of the TODAY and INTNX functions combined with the CALL SYMPUTX statement in a DATA _NULL_ data step. Although both of these approaches have their places, the use of the %SYSFUNC macro function takes the evolution forward one more step.

Using SASHELP.SNACKS, the following scenario is used to demonstrate the objectives of this paper: Suppose you are asked to create a report program that runs once each month and reports data covering one month in arrears of the run date. For example, running in October, reports September data.

SOLUTION 1: USING %LET STATEMENTS

```
%Let Ext_Beg_Dt = 01Sep2002;
%Let Ext_End_Dt = 30Sep2002;
Proc Sql;
  Title1 "Sales Report for &Ext_Beg_Dt. through &Ext_End_Dt.";
  Select
    Product
    , Sum(QtySold)           As Qty
    , Sum(QtySold * Price) As Sales
  From
    SASHelp.Snacks
  Where
    Date Between "&Ext_Beg_Dt."d And "&Ext_End_Dt."d
  Group by
    Product;
Quit;
```

EXPLANATION FOR SOLUTION-1

Solution-1 works, but each month, when it is run, the dates assigned to the macro variables, Ext_Beg_Dt and Ext_End_Dt must be manually reassigned. This is inefficient and cumbersome.

SOLUTION 2: USING A DATA_NULL_ DATA STEP

```
Data _Null_;
    Today = "15Oct2002"d; /* Normally, Today = Today(); would be used
here */
    Ext_Beg_Dt = Intnx("Month", Today, -1, "B");
    Ext_End_Dt = Intnx("Month", Today, -1, "E");
    Call SymputX("Ext_Beg_Dt", Ext_Beg_Dt);
    Call SymputX("Ext_End_Dt", Ext_End_Dt);
    Call SymputX("Rpt_Beg_Dt", Put(Ext_Beg_Dt, date9.));
    Call SymputX("Rpt_End_Dt", Put(Ext_End_Dt, date9.));
    Stop;
Run;
Proc Sql;
    Title1 "Sales Report for &Rpt_Beg_Dt. through &Rpt_End_Dt.";
    Select
        Product
        ,Sum(QtySold)           As Qty
        ,Sum(QtySold * Price) As Sales
    From
        SASHelp.Snacks
    Where
        Date Between &Ext_Beg_Dt. And &Ext_End_Dt.
    Group by
        Product
    ;
Quit;
```

EXPLANATION FOR SOLUTION-2

1. The macro variables, Ext_Beg_Dt, Ext_End_Dt, Rpt_Beg_Dt and Rpt_End_Dt are programmatically created, thus, eliminating the manual intervention required by Solution-1.
2. So that the Today() function is executed once, an assignment statement is used to populate the variable, Today, with the results of the TODAY() function.
3. Referencing the variable, Today, in the INTNX() function, the Ext_Beg_Dt is computed as the integer date for the first day of the month, one month in arrears of Today()'s date.
 - a. The alignment parm, "B", is used to compute the first day of the month. It stands for "Beginning."
4. Referencing the variable, Today, in the INTNX() function, the Ext_End_Dt is computed as the integer date for the last day of the month, one month in arrears of Today()'s date.
 - a. The alignment parm, "E", is used to compute the last day of the month. It stands for "Ending."
5. Using CALL SYMPUTX, the integer dates in the SAS® DATA STEP variables, Ext_Beg_Dt and Ext_End_Dt are stored in the macro variables by the same name.

6. By nesting the PUT() function inside CALL SYMPUTX, the integer dates stored in the DATA STEP variables, Ext_Beg_Dt and Ext_End_Dt, are converted to alphanumeric strings in date9 format and stored in the macro variables, Rpt_Beg_Dt and Rpt_End_Dt.
7. Since the DATA_NULL_step is not reading an input file, a STOP statement is used to terminate the DATA STEP.
8. In the PROC SQL step, the macro variables, Ext_Beg_Dt and Ext_End_Dt are used to subset SASHELP.SNACKS for the month of September while the macro variables, Rpt_Beg_Dt and Rpt_End_Dt are used to title the report.

SOLUTION 3: USING %SYSFUNC

```
%Let Today = '15Oct2002'd; /* Normally, %Let Today = %Sysfunc(Today());
would be used here */
%Let Ext_Beg_Dt = %Sysfunc(Intnx(Month, &Today, -1, B));
%Let Ext_End_Dt = %Sysfunc(Intnx(Month, &Today, -1, E));
%Let Rpt_Beg_Dt = %Sysfunc(Intnx(Month, &Today, -1, B), date9.);
%Let Rpt_End_Dt = %Sysfunc(Intnx(Month, &Today, -1, E), date9.);

Proc Sql;
  Title1 "Sales Report for &Rpt_Beg_Dt. through &Rpt_End_Dt.";
  Select
    Product
    , Sum(QtySold)           As Qty
    , Sum(QtySold * Price) As Sales
  From
    SASHELP.Snacks
  Where
    Date Between &Ext_Beg_Dt. And &Ext_End_Dt.
  Group by
    Product
  ;
Quit;
```

EXPLANATION FOR SOLUTION-3

1. The macro variables, Ext_Beg_Dt, Ext_End_Dt, Rpt_Beg_Dt and Rpt_End_Dt are programmatically created with use of the macro function, %Sysfunc, in %LET macro statements. This eliminates the need for the DATA_NULL_step from Solution-2.
2. Normally, the value assigned to the macro variable, Today, would use the following statement: %Let Today = %Sysfunc(Today());
3. Referencing the macro variable, Today, in the INTNX() function, the value for Ext_Beg_Dt is computed as the integer date for the first day of the month, one month in arrears of Today()'s date.
 - a. NOTE: <Month> and are NOT quoted. This is a requirement of using the %Sysfunc macro function.
4. Referencing the macro variable, Today, in the INTNX() function, the value for Ext_End_Dt is computed as the integer date for the last day of the month, one month in arrears of Today()'s date.
 - a. NOTE: <Month> and <E> are NOT quoted. This is a requirement of using the %Sysfunc macro function.

5. By repeating steps 3 and 4, but changing the name of the recipient macro variables to Rpt_Beg_Dt and Rpt_End_Dt and adding the SAS® date format, date9., the report dates are created and ready for reference in the Title1 statement.

SOLUTION FOR ALL THREE EXAMPLES

| Sales Report for 01Sep2002 through 30Sep2002 | | |
|--|-----|--------|
| PRODUCT | QTY | SALES |
| Baked potato chips | 0 | 0.00 |
| Barbeque pork rinds | 121 | 180.29 |
| Barbeque potato chips | 241 | 359.09 |
| Bread sticks | 0 | 0.00 |
| Buttery popcorn | 346 | 342.54 |
| Carmelized popcorn | 148 | 442.52 |
| Cheddar cheese bread sticks | 34 | 50.66 |
| Cheddar cheese popcorn | 0 | 0.00 |
| <Rows left out> | | |
| Sun-dried tomato multigrain chips | 185 | 528.65 |
| Tortilla chips | 380 | 376.20 |
| WOW cheese puffs | 0 | 0.00 |
| WOW potato chips | 0 | 0.00 |
| WOW tortilla chips | 262 | 783.38 |
| Wheat crackers | 0 | 0.00 |

Output 1: Solution for all 3 Examples

SUMMARY

By now one can see the inflexibility of Solution-1. As for Solution-2 and Solution-3, one could make the case, either solution is as good as the other; however, when writing more sophisticated macro programs, one might lean towards the use of Solution-3; however, the author recommends Solution-3.

Additionally, this simple example may prompt those writing macro programs to investigate other DATA STEP functions that can be used with %SYSFUNC. Keep in mind, though, not all DATA STEP functions can be used with %SYSFUNC. Read the SAS® documentation for additional information about SAS® DATA STEP functions that can and cannot be used with %SYSFUNC.

RESOURCES

SAS® 9.3 Macro Language: Reference

<http://support.sas.com/documentation/cdl/en/mcrolref/62978/HTML/default/viewer.htm#p1o13d7wb2zfcnn19s5ssl2zdxvi.htm>

CONTACT INFORMATION

Name: Bill Parman
Enterprise: Cigna®
Address: 401 Chestnut St
City, State ZIP: Chattanooga, TN 37402
E-mail: Francis.Parman@Cigna.Com

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.