Replacement of user supplied dates with variables that were calculated using SAS® date functions

Darlene M. Gerber, Geneva, Ohio
Allan M. Kominek, Euclid, Ohio

ABSTRACT

This paper takes you through the process of replacing user supplied dates with variables that have been assigned SAS® date values. It explains the code that calculates the correct SAS date values utilizing SAS date functions in Release 6.07 batch. What happens when you’re assigned a project that requires a number of user supplied date changes on a monthly basis, and you, the programmer, are not going to be responsible for making these changes? The solution requires a method to supply the correct dates to the programs, with the least amount of user interaction. This goal can be accomplished through the use of date functions and the SAS® Display Manager System(DMS).

ASSIGNMENT

The assignment was to make an already existing system more user friendly. The original system was executed by means of a TSO Clist™. The dates within each program were changed by editing the partitioned dataset members. The decision was made to redesign the system from scratch in a Version 6.07 SAS/AF® catalog. By rewriting the system, the established programs could be made more efficient and user friendly. The first task was to eliminate the need for editing the individual programs by automating the date calculations. There were seven programs requiring changes to a total of 15 dates. Within these programs there were 4 very different date ranges. The four ranges were the 3rd Saturday of the previous month last year to the last day of the present month, the 3rd Saturday of the present month to the 3rd Friday of the next month, the 3rd Saturday of the previous month to the 3rd Friday of the present month, today to the last day of the month 3 months from today. Not your normal first of the month to the end of the month date ranges.

DEFINITION OF SAS DATE FUNCTIONS THAT WERE USED

SAS date values are used in mathematical calculations and other programming techniques. The SAS System processes calendar date values by converting dates to integers representing the number of days between January 1, 1960, and a specified date(1).

A SAS function returns a value from a computation or system manipulation that requires zero or more arguments. Although most functions use arguments supplied by the user, a few obtain their arguments from the host operating system. The syntax of a function consists of a function name and function arguments; the arguments must be enclosed in parentheses(1). For example, use the following form for a function with two arguments:

function-name(argument-1,argument-2)
Parentheses must always be supplied even if the function does not require an argument. A function must be used as part of a SAS statement (1).

TODAY=TODAY();

Why use functions? The functions convert the date to a numeric value which can be easily manipulated by arithmetic operators.

TODAY() - The TODAY or DATE function produces the current date as a SAS date value representing the number of days between January 1, 1960, and the current date (1).

MONTH(DATE) - The MONTH function returns the month from a SAS date value as a numeric value from 1 through 12.

DATE specifies a SAS expression representing a SAS date value (1).

MDY(MONTH, DAY, YEAR) - The MDY function produces a SAS date value from numeric expressions that represent the month, day, and year.

MONTH specifies a numeric expression representing an integer from 1 through 12.

DAY specifies a numeric expression representing an integer from 1 through 31.

YEAR specifies a numeric expression representing an integer identifying a specific year (1).

WEEKDAY(DATE) The WEEKDAY function produces an integer representing the day of the week, where 1 = Sunday, 2 = Monday, ... 7 = Saturday, from a SAS date value.

YEAR(DATE) - The YEAR function produces a four-digit numeric value representing the year from a SAS date value.

EXAMPLE OF CODE

The following example shows the code used to determine the range for the 3rd Saturday of the previous month last year, to the last day of the present month. The code was tested in DMS extensively. Put statements were used to echo back what the code was doing to the date values.

The current date is 05/10/93, a Monday. The beginning of the range is the third Saturday in April of 1992, 04/18/92. Here is a brief description of the approach taken. From the current date the month and year can be determined using the MONTH() and YEAR() functions. It is then a simple matter to determine the previous month and previous year, by simply subtracting one from the month and year, unless the current month is January, in which case, the previous month is December. The date of the first day of the previous month and previous year can be determined by using the MDY() function and using 1 for the day. This will provide a reference point from which the third Saturday can be determined. To further clarify this reference point, the day of week must be determined for the first day of the month. In 1992, the first of April fell on a Wednesday. The final reference point is the first Saturday of the month. This can be calculated by adding the difference between 7, SAS considers Saturday the seventh day of the week, and the day of the week for the first of the month, which in this case is 4, Wednesday, 04/01/92 + (7 - 4) is 04/04/92. The third Saturday is merely this date plus 14 or 04/18/92. The end of the range, the last day of the current month can be calculated by adding one to the current month, checking to make sure that if the month equals 13 that it is translated to January of next year, using the MDY() function.
to create the SAS date for the first of the next month, and then subtracting one from this.

**Initial Steps**

**LOGON TO SAS**

**CREATE A DATA STEP**

**DATA DATES;**

/* ASSIGN TODAY'S DATE TO A VARIABLE *;*/

TODAY=TODAY();
PUT TODAY=;
TODAY=12183 /* May 10, 1993 */;

/* DETERMINE THE FIRST DAY OF THE PRESENT MONTH*/;

MONTH=MONTH(TODAY);
PUT MONTH=;
MONTH=5
DAY=1;
PUT DAY=;
DAY=1;
YEAR=YEAR(TODAY);
PUT YEAR=;
YEAR=1993

/* DETERMINE THE FIRST DAY OF THE PREVIOUS MONTH LAST YEAR Need to take into consideration if month is Feb-Dec. */;

IF MONTH > 1 THEN DO;
FIRSTDAY=MDY(MONTH-1,DAY,YEAR-1);
PUT FIRSTDAY=;
THIRDSAT=((7-DOW)+FIRSTDAY)+14;
PUT THIRDSAT=;
THIRDSAT=11796
END;
ELSE DO;
FIRSTDAY=MDY(MONTH+11,DAY,YEAR-1);
PUT FIRSTDAY=;
END;

/* DETERMINE THE THIRD SATURDAY OF THE PREVIOUS YEAR Determine what day of the week the value of FIRSTDAY represents. */;

DOW=WEEKDAY(FIRSTDAY);
PUT DOW=;
DOW=4
THIRDSAT=((7-DOW)+FIRSTDAY)+14;
PUT THIRDSAT=;
THIRDSAT=11796

/* DETERMINE THE LAST DAY OF THE PRESENT MONTH */;

IF MONTH < 12 THEN DO;
LDATE=MDY(MONTH+1,DAY,YEAR);
PUT LDATE=;
LDATE=12205
LASTDAY=LDATE-1;
PUT LASTDAY=;
LASTDAY=12204

/*Final format to use in batch code*/;

LASTDATE=PUT(LASTDAY,YYMMDD6.);
PUT LASTDATE=;
LASTDATE=930531
END;
ELSE DO;
LDATE=MDY(MONTH-11,DAY,YEAR+1);
PUT LDATE=;
LASTDAY=LDATE-1;
PUT LASTDAY=;

/*Final format to use in batch code*/;

LASTDATE=PUT(LASTDAY,YYMMDD6.);
LASTDATE=930531
END;

/* EXAMPLE OF BATCH CODE */;

SATDATE=920418
LASTDATE=930531
IF COMPDATE >= SATDATE AND COMPDATE <= LASTDATE;

SATDATE=PUT(THIRDSAT,YYMMDD6.);
PUT SATDATE=;
SATDATE=920418

/* DETERMINE THE LAST DAY OF THE PRESENT MONTH */;

IF MONTH < 12 THEN DO;
LDATE=MDY(MONTH+1,DAY,YEAR);
PUT LDATE=;
LDATE=12205
LASTDAY=LDATE-1;
PUT LASTDAY=;
LASTDAY=12204

/*Final format to use in batch code*/;

LASTDATE=PUT(LASTDAY,YYMMDD6.);
PUT LASTDATE=;
LASTDATE=930531
END;
ELSE DO;
LDATE=MDY(MONTH-11,DAY,YEAR+1);
PUT LDATE=;
LASTDAY=LDATE-1;
PUT LASTDAY=;

/*Final format to use in batch code*/;

LASTDATE=PUT(LASTDAY,YYMMDD6.);
LASTDATE=930531
END;

/* EXAMPLE OF BATCH CODE */;

SATDATE=920418
LASTDATE=930531
IF COMPDATE >= SATDATE AND COMPDATE <= LASTDATE;
CONCLUSION

In summary, working with date values can be difficult until you realize that they are only numbers. And numbers can be manipulated. Extensive use of DMS for testing purposes, will make manipulating SAS date values easier. As a final statement keep the following always in your mind. Dates may frustrate you, they may drive you crazy, but most of us would be without jobs if we were not needed to search out information for certain time periods.

REFERENCES


ACKNOWLEDGEMENTS

TSO Clist is a registered trademark of the International Business Machine Corporation.