ABSTRACT

Once you realize how easy it is to let SAS® get your interval dates for you, you will never be entering date parameters again.

The intended audience is Base SAS® users of all experience levels on any platform.

INTRODUCTION

To simplify the running of repeating reports which are scheduled at regular intervals, we can use SAS® functions to create macro variables to represent the date range for the current execution of our report. This can prevent accidental introduction of errors and make us more productive, since this code is reusable. In a batch production environment, where SAS® jobs can be scheduled to run on a regular basis without human intervention, this avoids the necessity of changing parameters each time we run. These techniques will be applicable even if we need to submit from our Editor Window, so we will not have to revise date-specific code every time.

Problem: Run Quarterly Report

Four times a year, a report must be generated to compare current year and quarter year-to-date with the previous year. We do not want to make changes to the program or have to enter or hard-code date parameters for each run.

Constraints:
1. Data collection requires at least thirty days after the end of the quarter to assure that all data has been entered.
2. This report needs to run on a quiet night, not in contention with weekly or monthly jobs.

It will be put on the production schedule to run automatically every three months, on the second Thursday night of the month after the quarter ends (or about six weeks after each quarter.) In 2002, it ran on May 9, August 8, November 14 and February 13, 2003 for 2002 - Q1, Q2, Q3, Q4.

How can we get the start and end dates for the report automatically in our program?

DATA _NULL_;
    CALL SYMPUT('qtr_end', '' || TRIM(LEFT(PUT(INTNX('MONTH',DATE(),-2,'END'), DATE9.))) ||'d' ));
RUN ;
%PUT End of Quarter= &qtr_end ;
FINAL PROGRAM CODE
To compare this year and quarter year-to-date with last year, we require start and end dates for each year. The four macro variables &start and &end for this year and &start0 and &end0 for last year are created in the program segment below. Later in the program, they are used for data selection. Notice that the end date is selected here with the INTNX parameters 'QTR', -1 instead of 'MONTH', -2. They are equivalent in this case. The prior year value is 'QTR', -5. The start dates are the first day of this year 'YEAR', 0,'BEG' and last year 'YEAR', -1,'BEG'. The last parameter is optional here because 'BEG' is the default value.

DATA _NULL_;
CALL SYMPUT('start', '"'||TRIM(LEFT(PUT(INTNX('YEAR',DATE(),0,'BEG'), DATE9.)))||'"d') ;
CALL SYMPUT('end', '"'||TRIM(LEFT(PUT(INTNX('QTR',DATE(),-1,'END'), DATE9.)))||'"d') ;
CALL SYMPUT('start0', '"'||TRIM(LEFT(PUT(INTNX('YEAR',DATE(),-1,'BEG'), DATE9.)))||'"d') ;
CALL SYMPUT('end0', '"'||TRIM(LEFT(PUT(INTNX('QTR',DATE(),-5,'END'), DATE9.)))||'"d') ;
RUN ;
%PUT This year &start - &end ;
%PUT Last year &start0 - &end0 ;
* ... ;
PROC SQL ;
SELECT ...
FROM ... T
WHERE (filedate BETWEEN &start AND &end
OR filedate BETWEEN &start0 AND &end0) ;
QUIT ;

ALTERNATE CODE
For data retrieval from databases other than SAS®, the date formats may be MMDDYY10 or DATETIME instead of DATE9. Then, the single quotes and the letter d will not be necessary.

DATA _NULL_;
CALL SYMPUT('qtr_end', TRIM(LEFT(PUT(INTNX('MONTH',DATE(),-2,'END'), MMDDYY10.)))) ;
RUN ;
%PUT End of Quarter= &qtr_end ;

CONCLUSION
Use SAS® functions to create macro variables to represent the date range for the current execution of repeating reports. Avoid the necessity of changing parameters each time.

REFERENCES

ACKNOWLEDGMENTS
Thanks to LeRoy Bessler, Bessler Consulting & Research, Milwaukee, WI, for encouraging me to give another presentation.

CONTACT INFORMATION
Your comments and questions are valued and encouraged.
Contact the author at:
Nina L. Werner, BA, MBA, MHP
Dean Health Plan, Inc.
1277 Deming Way
Madison, WI 53717
Work Phone: (608) 827-4224
Fax: (608) 836-6335
Email: Nina.Werner@Deancare.com
Web: www.Deancare.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.

Key words: CALL SYMPUT, DATE(), INTNX, PUT, macro variable, TODAY().