

Dates with Macros

Alex Chaplin

March 7th, 2018

The scenario – July 2018. A warm and sunny day.

- You get a request to pull January 2018 data from different data sources across different platforms
- You create a SAS program to do this
- You hard code dates in different formats to match date conventions used on data sources and within the data
- Output looks good. You pass along the results to your business partner
- Your business partner says this is great, can you pull all four months prior and all three months after?
- You anticipate lots of time fiddling with the dates in your code to get them right to pull for each month

But wait...

It doesn't have to be like
this

Why?

Because macros are
awesome for these types
of requests!

Dates with macros

Example output - mylib.BASELINE_POP

```
%let yymm = 1801; /* Starting YYMM */  
%let i = 4; /* Number of prior months to extract */  
%let p = 3; /* Number of subsequent months to extract */  
/* Drop appended data from previous run? 0= No 1=Yes */  
%let drop_data = 1;
```

	▲ proc_yymm	▲ proc_bdb2dt	▲ proc_edb2dt	▲ proc_6db2dt	▲ proc_pedb2dt
1	1801	'2018-01-01'	'2018-01-31'	'2018-01-06'	'2017-12-31'
2	1712	'2017-12-01'	'2017-12-31'	'2017-12-06'	'2017-11-30'
3	1711	'2017-11-01'	'2017-11-30'	'2017-11-06'	'2017-10-31'
4	1710	'2017-10-01'	'2017-10-31'	'2017-10-06'	'2017-09-30'
5	1709	'2017-09-01'	'2017-09-30'	'2017-09-06'	'2017-08-31'
6	1802	'2018-02-01'	'2018-02-28'	'2018-02-06'	'2018-01-31'
7	1803	'2018-03-01'	'2018-03-31'	'2018-03-06'	'2018-02-28'
8	1804	'2018-04-01'	'2018-04-30'	'2018-04-06'	'2018-03-31'

dates_with_macros.sas

- Specify
 - start month, default is current month end
 - number of months back
 - number of months forward
 - keep/drop data accumulated from prior run
- Uses SAS macro language to reference month date variables for current, prior and subsequent months making use of
 - forward rescan rule to loop through months
 - %sysfunc to reference base SAS functions
 - intnx function for date manipulation and creating custom dates
- Code for data extract goes inside SAS macro date_macro_demo

dates_with_macros.sas includes..

dt_macro.sas

- Assigns date macro variables including crnt_mo for current month end in YYMM format
- dates_with_macros.sas assigns crnt_mo or other date variable to macro variable yymm used by month_dates.sas

month_dates.sas

- Takes date macro variable yymm, assigned by dates_with_macros.sas, and creates date macro variables in different formats for
 - month of yymm
 - each of 24 prior months
 - each of 24 subsequent months

dates_with_macros.sas - Top Part

```
libname mylib "filepath/mydata";
```

```
/* Include libraries */
```

```
filename inclib "filepath/myinclib";
```

```
/* Assign date macro variables
```

```
%include inclib(dt_macro);
```

Within dt_macro.sas

Assigns macro variable crnt_mo to current month end in yymm format e.g. if we are in March 2018 then crnt_mo is 1802

```
call symput('crnt_mo',put(intnx('month',today(),-1),yymmdd4.));
```


dates_with_macros.sas – Bottom Part

```
/* Specify starting YYYY */
%let yymm = &crnt_mo.; /* Automated batch processing */
%let yymm = 1801;      /* Custom run */
/* Number of previous months to extract */
%let i = 4;
/* Number of subsequent months to extract */
%let p = 3;
/* Drop appended data from previous run? 0= No 1=Yes */
%let drop_data = 1;
/* Assign macro date variables 24 months before and after yymm */
%include inclib(month_dates);
run;

%ext_data() /* Call macro to extract month end data */
```

dates_with_macros.sas – Middle Part

Define macro to extract data

```
%macro ext_data();  
  /* Drop data accumulated from prior run if drop_data = 1 */  
  %if &drop_data = 1  
  %then %do;  
    proc sql;  
      drop table mylib.BASELINE_POP;  
    quit;  
  %end;  
  
  /* Extract data for selected month end */  
  %date_macro_demo(&cmyymm.,&cm_bdate_db2_lit.,&cm_date_db2_lit.,&cday6_db2_lit.,&&pdate_db2_lit&k.)
```

dates_with_macros.sas – Middle Part

Define macro to extract data - continued

```
/* Increment counters in loops to extract data for other months */  
/* One loop for months ahead and one loop for months prior */  
  
/* Use forward rescan rule to reference macro variables previously  
defined in month_dates.sas */  
%date_macro_demo(&&pmyymm&k.,&&pbrate_db2_lit&k.,&&pdate_db2_lit&k.,&&pday6_db2_lit&k.,&&pdate_db2_lit&m.)  
  
/* Use eval to increment macro variable counters */  
%let k = %eval(&k+1);  
%let m = %eval(&m+1);  
  
%mend ext_data;
```

dates_with_macros.sas – Middle Part

Define the demo macro to create the extract file for each month

```
%macro date_macro_demo(proc_yymm, proc_bdb2dt,  
proc_edb2dt, proc_6db2dt, proc_pedb2dt);
```

```
/* Demo uses symget to assign variables from macro values */
```

```
data BASELINE_POP&proc_yymm.;
```

```
    proc_yymm = symget("proc_yymm");
```

```
    proc_bdb2dt = symget("proc_bdb2dt");
```

```
    proc_edb2dt = symget("proc_edb2dt");
```

```
    proc_6db2dt = symget("proc_6db2dt");
```

```
    proc_pedb2dt = symget("proc_pedb2dt");
```

```
run;
```

dates_with_macros.sas – Middle Part

Define the demo macro to create the extract file for each month -
contd

```
/* Append each month */  
proc append base=mylib.BASELINE_POP  
data=BASELINE_POP&proc_yymm.;  
run;  
  
/* Delete work file to free up storage and improve run efficiency */  
proc sql;  
    drop table BASELINE_POP&proc_yymm.;  
quit;  
  
%mend date_macro_demo;
```

Recommendations

- Test your code in SAS EG
 - SAS macro debugging options on / %put _user_
 - Limit data extract
 - DB2 fetch first 100 rows only with ur
 - HADOOP limit 100
 - Oracle where rownum <= 100
 - SAS data step obs=100
 - SAS PROC SQL outobs=100
 - SQL Server top 100
 - Teradata sample 100
- Run the final version in the background on UNIX
 - SAS macro debugging options off
 - You can log off and leave it to run overnight for large extracts

Acknowledgements

Authors and contributors to date_macro.sas and month_dates.sas

Bob Lawson

Amit Parikh

Michelle Tyler

Alex Chaplin

References

[Listen Data](#)

[SAS® 9.4 Macro Language: Reference, Fifth Edition](#)

[SAS® Macros 101 How I learned to stop worrying and love macros](#)

Alex Chaplin

QUESTIONS