Getting the Right DATES

Marje Fecht

Senior Partner, Prowerk Consulting

SAS Global Forum 2014 Conference Chair

Getting the RIGHT Date can be Tricky

This presentation focuses on working with dates in **SAS** and **Teradata**.

- Comparing Dates
- Extracting Date Components
- Moving Forward and Backward in Time
- Outputting Dates in Desired Formats
- Generalizing Code to Accommodate Dates

SAS Date/Time

How does SAS store dates?

- A SAS Date is a **numeric variable** and represents the **integer** number of days since Jan 1, 1960
- Arithmetic operations work with SAS Date Values

```
Enddate = startdate + 14;
```

- To determine the system date, use

```
Date = today();
```

SAS Date example

```
data dates;
   SystemDate = today();
   putlog 'Unformatted System Date is: ' SystemDate  ;
   putlog 'formatted with date9.     : ' SystemDate date9.  ;
   putlog 'formatted with worddate.  : ' SystemDate worddate.  ;
   putlog 'formatted with weekdate.  : ' SystemDate weekdate.  ;
   putlog 'formatted with weekdate9. : ' SystemDate weekdate9.  ;
run;
```

```
Unformatted System Date is: 19283
formatted with date9.     : 17OCT2012
formatted with worddate.  :   October 17, 2012
formatted with weekdate.  :   Wednesday, October 17, 2012
formatted with weekdate9. : Wednesday
```

NOT a SAS Date

If dates are stored in SAS as **character**, they are **not** SAS dates.

```
data chardate;
  chardate = '1957-03-15';
  putlog 'Character Variable Value: ' chardate;

  /**convert to a SAS date **/
  SAS_date = input(chardate , yymmdd10.);

  putlog 'Unformatted value of SAS_Date:' SAS_date;
  putlog 'Formatted with ddmmyyS10. :'  SAS_Date ddmmyyS10.;
run;
```

```
Character Variable Value: 1957-03-15
Unformatted value of SAS_Date:-1022
Formatted with ddmmyyS10. :15/03/1957
```

Teradata Dates

- In Teradata, the default **input format** for dates is **yyyy-mm-dd**
- Internally, Teradata stores the date as a 4 byte signed integer calculated as

  (Year – 1900) * 10000 + (month * 100) + Day
- When you extract dates from Teradata and store the data in SAS, they are stored as SAS date values

Comparing Dates

SAS

Assume me_dt is a SAS date .

```
If me_dt = '31mar2012'd ;


%let enddate = 31mar2012;
If me_dt = "&enddate"d;
```

Teradata

Assume me_dt is a Teradata date .

```
Where me_dt = '2012-03-31'

%let enddate='2012-03-31';
Where me_dt = &enddate

%let enddate = 2012-03-31;
Where me_dt =
     %str(%')&enddate%str(%')
```

Notes about Teradata

- If you represent the date literal in the wrong format, Teradata will give you a conversion error

```
Where me_dt = '03-31-2012'
```

- If you use double quotes around a date literal, teradata thinks it is a column and you get an error

```
Where me_dt = "2012-03-31"
```

- To get system clock date, use **`current_date`**

Extracting date parts

SAS

Use SAS Functions or Formats

Month
Day - *Day of Month*
Year
Qtr - *Calendar Qtr*
Hour
WeekDay - *1=Sunday, …*
etc

Teradata

Use Extract

```
Extract( MONTH from me_dt)
```

Use CAST

```
CAST ( datestamp01 as DATE )
  /* isolate date from a
date-time value */
```

Extract Date Parts in SAS

```
data extract;
   SystemDate = today();
   Today_Year = year(SystemDate);
   Today_Month = month (SystemDate);
   Today_Day = day (SystemDate);
   Today_WeekDay = weekday (SystemDate);
   Today_WeekDay_2 = put (SystemDate , weekdate9.);
   /* use PUTLOG to write */   run;
```

```
SystemDate=19283 SystemDate=17OCT2012
Today_Year=2012 Today_Month=10 Today_Day=17
Today_WeekDay=4
Today_WeekDay_2=Wednesday
```

Moving in Time

If you need to move forward or backward by months, take advantage of functions to make it easy.

SAS

INTNX

- Moves forward or backward
- use common date intervals

INTCK

- Counts date intervals between two dates

Teradata

ADD_MONTHS

```
ADD_MONTHS (me_dt , 3)
```

INTNX = *move in intervals*

INTNX - handy to dymanically create different variations of dates.

- *increments dates by intervals*

INTNX (*interval*, *from*, *n* < , *alignment* >) ;

- ***interval*** **-** interval name eg: 'MONTH', 'DAY', 'YEAR'
- ***from*** **-** a SAS date value (for date intervals) or datetime value (for datetime intervals)
- ***n*** **-** number of intervals to increment from the interval that contains the *from* value
- ***alignment*** **-** alignment of resulting SAS date, within the interval.

  Eg: **B**EGINNING, **M**IDDLE, **E**ND.

Check for Current Month

```
/**** Extract all data for current month ***/

  if me_dt =
         INTNX ( 'MONTH'    /*increment = month*/
                 , 0         /* move ZERO months*/
                 , today() /* start at today */
                 , E)       /* return END of mth */
             ;
```

Best Practices: Version Control

```
%let datetime = %sysfunc(compress(%sysfunc(today(),
                        yymmddN8.)_%sysfunc(time(),hhmm6.), ': '));
** route Log and Listing to permanent location **;
proc printto
        log ="&dir.\logs\&stage.\&filename._&datetime..log"
        print="&dir.\output\&stage.\&filename._&datetime..lst";
run;
... < other program logic > ...
** reroute to default locations **;
proc printto; run;
```

The resulting "Versioned" file names would be

HR123_20120507_1329.log

HR123_20120507_1329.lst

Note: The N in yymmddN8 requests NO separators (Dashes, Slashes, etc)

Program Control

```
*** Run monthly report on first of each month;

%macro dayone;
   %if %sysfunc(day( %sysfunc(today()) )) = 1
      %then %do;
         %include 'monthly_report.sas';
      %end;
%mend;

%dayone

*** Run daily report every day;
%include 'daily_report.sas';
```

Fiscal Year

```
/**** Create a Macro to compute fiscal year
          start ➔ beginning of FY
 ***/
%macro fy (date , start=7);
   year(&date) + /* BELOW returns a 0 (F) or a 1 (T) */
         (month(&date) ge &start and &start ne 1)
%mend;

/*** example of macro usage ***/
data try_FY;
  Txn_date = '21nov2012'd;
  FiscalYear = %FY( Txn_Date , start = 11 );
  putlog FiscalYear =
        Txn_Date=date9.;
run;
```

Before we finish . . .

Dates are a primary component of most all results we produce

- VALIDATE that your programs ***Get the Right Dates!***

Thank you

to Laura House for her review and suggestions

Any errors in this document are all my fault ☺

Thank You !

Marje Fecht

marje.fecht@prowerk.com