

**Pujan Shah**  
**Auto Claims & Reserving**  
**November, 2016**

## **Automation of Date Macro Variable**



# Overview



- Automation of Date Macros.
  - If you have a program that you run once every month, every week or even daily and have to update macro variables that contain date, then this bit of code can help.

```
data _null_;
```

```
Lastday_of_PriorMonth = put(intnx('month',today(),-1,'E'),Date9.);  
call symput('Prior_MonthEnd',''|Lastday_of_PriorMonth||"'d");
```

```
run;
```

# Functions



- **Intnx** function fetches the desired date.
- **Put** function returns the date in desired format and assigns it to a temporary data set variable in character format.
- **Call Symput** function assigns the value in data set variable to an existing or new macro variable. This macro variable can be in a character format or date value.
- This code can be modified to automate any type of date and time variable. More info on each of these function is available on SAS website.

```
data _null_;
```

```
FirstDay_of_NextWeek = put(intnx('week',today(),1,'B'),Date9.);  
call symput('Prior_WeekB','"||FirstDay_of_NextWeek||"');
```

```
run;
```

# Function Detail



- Intnx

- Time **Interval** argument can be either one of standard SAS options or a customized interval.
  - An interval can also be modified to be a multiple of standard interval. For e.g., 'month2' will be a two month interval.
  - An interval can also be shifted to start from a particular point. For e.g., 'week.2' will be a week interval starting from second day of the week.
- **Starting-from** argument can be any date or time variable.
  - It's useful to use today() as the starting-from argument because it gives an anchor point to building logic around and is automatically derived every time the program is run.
  - Alternatively, a fixed date can also be used as the anchor point to the build logic around.
- **Increment** argument is a integer number that specifies the interval shift from starting point.

# Function Detail



- **Alignment** argument directs the program to select the date from the selected interval. Options for this argument are:
  - B – Beginning, it selects the first date in the interval.
  - E – End, it selects the last date in the interval.
  - M – Middle, it selects the last date in the middle of the interval.
  - S – Same, it selects the date with same location in the interval as starting date.
- Put
  - This function will allow you to format your date or time in any SAS supported format. For e.g., Date9. , Year4., yymmnn4., worddate.,etc.
- Call Symput
  - This function allows you to create a new macro variable or modify existing macro variable to assign the desired date.

```
Call symput('Prior_WeekB', FirstDay_of_NextWeek);
```

```
Call symput('Prior_WeekB', ""||FirstDay_of_NextWeek||"d");
```

# Example

---

```
data _null_;  
TestDay = put(intnx('month2', today(), -1, 'B'), Date9.);  
call symput('TestDayMacro', "" || TestDay || "'d");  
run;  
TestDay = 01SEP2016
```

```
data _null_;  
TestDay = put(intnx('month2', today(), -1, 'E'), Date9.);  
call symput('TestDayMacro', "" || TestDay || "'d");  
run;  
TestDay = 31OCT2016
```

Month2 intervals: Jan-Feb, March-April, May-June,  
Jul-August, Sep-Oct, November-December

# Example

---

```
data _null_;
```

```
TestDay = put(intnx('week', today(), -1, 'B'), Date9.);
```

```
call symput('TestDayMacro', "" || TestDay || "'d");
```

```
run;
```

```
TestDay = 30OCT2016
```

```
data _null_;
```

```
TestDay = put(intnx('week.2', today(), -1, 'B'), Date9.);
```

```
call symput('TestDayMacro', "" || TestDay || "'d");
```

```
run;
```

```
TestDay = 31OCT2016
```

Week interval starts on the first day of the week which is Sunday.

“.2” shifts the interval starting point from second day of the week which is Monday.

# Example

---

```
data _null_;
```

```
TestDay = put(intnx('month', today(), -1, 'S') + 1, Date9.);  
call symput('TestDayMacro', "" || TestDay || "d");
```

```
run;
```

```
TestDay = 12OCT2016
```

Date output from intnx function is an integer number and can be used as such. In this example I'm adding one day to the output from intnx function.  
 $1/1/1960 = 0$