Reporting Made Easier

Presented by: Devbani Raha

Staff Epidemiologist, CCNS

Cancer Care Nova Scotia

Objective

- Have some ideas of how to modify programs to minimize changing code when re-running code
- Learn SAS functions:
  - today()
  - intnx()
  - put()
  - call symput()
- Learn an application for data _null_
- Simplify recurrent reports

Background

- Often once a report is developed and deemed useful, client asks for report on a recurrent schedule:
  - weekly, monthly, quarterly, etc
- During development phase, pertinent dates are hard coded:

```
data newdata;
  set largedata
  (where=('01apr2011'd<=event_date<='30sep2011'd));
… <more code>;
run;
```

Sample dates used for this presentation

- The reporting period will start at the beginning of the current fiscal year, i.e. 01Apr2011
- The reporting period will end at the end of two quarters prior to the current quarter to time allow for data entry:
  - Current quarter (in terms of fiscal years is 2011Q4)
  - End of previous quarter: 31Dec2011
    - but we want a lag of one quarter to allow time for data entry, so…
  - Report will end 30Sep2011

First steps… or Solution 1

Original Code

```
data newdata;
   set largedata
   (where=('01apr2011'd<=event_date<
   ='30sep2011'd));
… <more code>;
run;
```

With macros

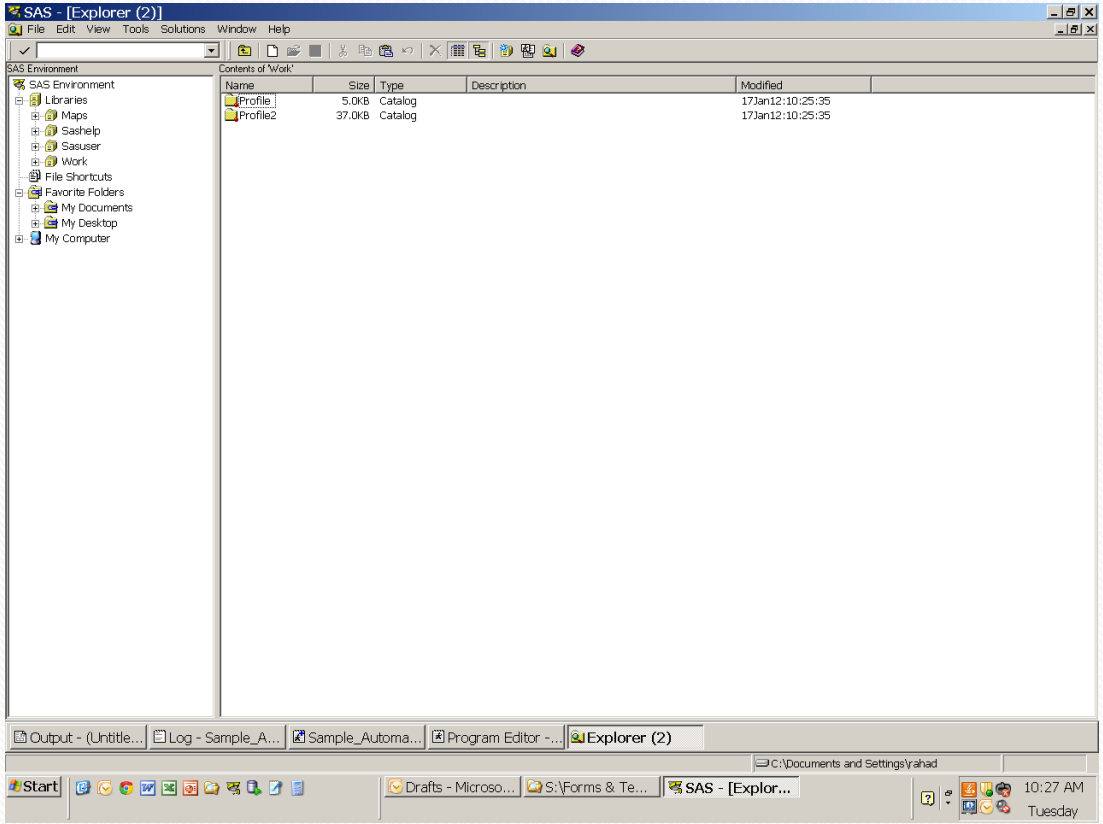
```
%let stdt='01apr2011'd;
%let enddt='30sep2011'd;

data newdata;
   set
   largedata(where=(&stdt<=event
   _date<=&enddt));
… <more code>
run;
```

Solution 2

- If report is scheduled regularly, then can write code that generates:
  - the date the program runs
    - Useful for time stamping any output files (datasets, EXCEL files, list files, etc)
  - the start of the reporting period
  - the end of the reporting period

Automating the code

```
data _null_;
```



Automating the code

```
data _null_;
*** today's date for the day the program is run;
  rundt=today();
```

Intnx()

- creates dates from a fixed point in time
- requires 4 arguments
- intnx('<shift interval>', starting date, <# intervals>,<>)
- shift interval – may be weeks, months, years, etc
- starting date from when to shift
- # intervals – must be integer, may be less than zero
- final argument is optional. Default is beginning of the interval.
  - 'b', 'm', 's', 'e'

Automating the code

```
data _null_;
*** today's date for the day the program is run;
  rundt=today();
*** report period ends on the last day of the quarter
  before. Lagged by one quarter to allow for data entry
  time;
  enddt=intnx('quarter',rundt,-2,'e');
*** reporting period starts at the beginning of the
  current fiscal year;
  stdt=intnx('year1.4',enddt,0,'b');
```

Automating the code

```
data _null_;
*** today's date for the day the program is run;
  rundt=today();
*** report period ends on the last day of the quarter
  before;
  enddt=intnx('quarter',rundt,-2,'e');
*** reporting period starts at the beginning of the
  current fiscal year;
  stdt=intnx('year1.4',enddt,0,'b');
  call symput('rundt',put(rundt,yymmddn8.));
  call symput('stdt',"'"||put(stdt,date9.)||"'D'");
  call symput('enddt',"'"||put(enddt,date9.)||"'D'");
run;
%put &rundt &stdt &enddt;
```

From the SAS Log

- Using a run date of 23Feb2012

```
1    data _null_;
2        rundt=today();
3        enddt=intnx('quarter',rundt,-2,'e');
4        stdt=intnx('year1.4',enddt,0,'b');
5        call symput('rundt',put(rundt,yymmddn8.));
6        call
    symput('stdt',"'"||put(stdt,date9.)||"'d");
7        call
    symput('enddt',"'"||put(enddt,date9.)||"'d");
8     run;

NOTE: DATA statement used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds


9     %put &rundt &stdt &enddt;
20120223 '01APR2011'd '30SEP2011'd
```

- Using Future dates to check

```
data _null_;
12       rundt='01Apr2012'd;
13      enddt=intnx('quarter',rundt,-2,'e');
14       stdt=intnx('year1.4',enddt,0,'b');
15       call symput('rundt',put(rundt,yymmddn8.));
16       call
    symput('stdt',"'"||put(stdt,date9.)||"'d");
17       call
    symput('enddt',"'"||put(enddt,date9.)||"'d");
18    run;

19    %put &rundt &stdt &enddt;
20120401 '01APR2011'd '31DEC2011'd
20
21    data _null_;
22       rundt='01Oct2012'd;
23       enddt=intnx('quarter',rundt,-2,'e');
24       stdt=intnx('year1.4',enddt,0,'b');
25       call symput('rundt',put(rundt,yymmddn8.));
26       call
    symput('stdt',"'"||put(stdt,date9.)||"'d");
27       call
    symput('enddt',"'"||put(enddt,date9.)||"'d");
28    run;
29    %put &rundt &stdt &enddt;
20121001 '01APR2012'd '30JUN2012'd
```

No more re-coding!

```
data newdata;
  set
  largedata(where=(&stdt<=event_date<=
  &enddt));
  … more code;
run;
```

- The starting and ending macros are automated and your code won't need to be touched unless the schedule for your report changes (!)

When to put in the effort?

- When contents of report are stable, i.e. won't decide to add or remove indicators for a period of time
- Schedule for producing the report is clear and on a pre-determined interval: monthly, weekly, quarterly, etc
- Especially useful for reports that use multiple SAS programs (or a system of programs)

Questions

Devbani Raha

Devbani.Raha@cdha.nshealth.ca

Cancer Care Nova Scotia