Underutilized Features in SAS® Macro Language

George J. Hurley
Sr. Research Manager
The Hershey Company
Milwaukee June 2013

Overview

- %EVAL; %SYSEVALF
- %SYSFUNC
- Quoting functions
- Parmbuff option

%EVAL; %SYSEVALF

```
%let x=3;
%let y=4;
%let z=&x+&y;

%put x=&x;
%put y=&y;
%put z=&z;
```

```
1     %let x=3;
2     %let y=4;
3     %let z=&x+&y;
4
5     %put x=&x;
x=3
6     %put y=&y;
y=4
7     %put z=&z;
z=3+4
```

This isn't always bad:

```
data one;
x=&x;
y=&y;
z=&z;
run;

proc print noobs;
run;
```

```
x    y    z

3    4    7
```

But, sometimes you need the value stored in the macro variable – %EVAL

%EVAL; %SYSEVALF

```
%let x=3;
%let y=4;
%let z=%eval(&x+&y);

%put x=&x;
%put y=&y;
%put z=&z;
```

```
17
18   %let x=3;
19   %let y=4;
20   %let z=%eval(&x+&y);
21
22   %put x=&x;
x=3
23   %put y=&y;
y=4
24   %put z=&z;
z=7
```

Or:

```
%let x=3;
%let y=4;
%let z=&x+&y;

%put x=&x;
%put y=&y;
%put z=%eval(&z);
```

```
25
26   %let x=3;
27   %let y=4;
28   %let z=&x+&y;
29
30   %put x=&x;
x=3
31   %put y=&y;
y=4
32   %put z=%eval(&z);
z=7
```

%EVAL; %SYSEVALF

```
%let w=3;
%let x=3.5;
%let y=4;
%let z1=%eval(&x+&y);
%let z2=%sysevalf(&x+&y);
%let d1=%eval(&y/&w);
%let d2=%sysevalf(&y/&w);

%put x=&x;
%put y=&y;
%put z1=&z1;
%put z2=&z2;
%put d1=&d1;
%put d2=&d2;
```

```
157  %let w=3;
158  %let x=3.5;
159  %let y=4;
160  %let z1=%eval(&x+&y);
ERROR: A character operand was found in the %EVAL function or %IF
condition where a numeric
       operand is required. The condition was: 3.5+4
161  %let z2=%sysevalf(&x+&y);
162  %let d1=%eval(&y/&w);
163  %let d2=%sysevalf(&y/&w);
164
165  %put x=&x;
x=3.5
166  %put y=&y;
y=4
167  %put z1=&z1;
z1=
168  %put z2=&z2;
z2=7.5
169  %put d1=&d1;
d1=1
170  %put d2=&d2;
d2=1.33333333333333
```

What happened?

%EVAL does not do floating point arithmetic

%EVAL; %SYSEVALF

```
%let w=3;
%let y=4;
%let z=%eval(&w>&y);
%put &z;
```

```
178  %let w=3;
179  %let y=4;
180  %let
z=%eval(&w>&y);
181  %put &z;
0
```

```
%macro compare(first,second);
  %if &first>&second %then %put
&first > &second;
  %else %if &first=&second %then
%put &first = &second;
  %else %put &first<&second;
%mend compare;
%compare(1,2)
%compare(-1,0)
%compare(12.8,2.2)
%compare(1.8,2.2)
```

```
184  %macro compare(first,second);
185     %if &first>&second %then
%put &first > &second;
186     %else %if &first=&second
%then %put &first = &second;
187     %else %put &first<&second;
188  %mend compare;
189  %compare(1,2)
1<2
190  %compare(-1,0)
-1<0
191  %compare(12.8,2.2)
12.8<2.2
192  %compare(1.8,2.2)
1.8<2.2
```

Wait! 12.8 is not less than 2.2

%EVAL; %SYSEVALF

```
%let w=3;
%let y=4;
%let z=%eval(&w>&y);
%put &z;
```

```
178  %let w=3;
179  %let y=4;
180  %let
z=%eval(&w>&y);
181  %put &z;
0
```

```
%macro compare(first,second);
  %if %sysevalf(&first>&second)
%then %put &first > &second;
  %else %if
%sysevalf(&first=&second) %then
%put &first = &second;
  %else %if
%sysevalf(&first<&second) %then
%put &first<&second;
%mend compare;
%compare(1,2)
%compare(-1,0)
%compare(12.8,2.2)
%compare(1.8,2.2)
```

```
214  %macro compare(first,second);
215      %if %sysevalf(&first>&second) %then
%put &first > &second;
216      %else %if %sysevalf(&first=&second)
%then %put &first = &second;
217      %else %if %sysevalf(&first<&second)
%then %put &first<&second;
218  %mend compare;
219  %compare(1,2)
1<2
220  %compare(-1,0)
-1<0
221  %compare(12.8,2.2)
12.8 > 2.2
222  %compare(1.8,2.2)
1.8<2.2
```

Now it is correct

%SYSFUNC

```
data _null_;
dt='20may13'd;
call symput ("dt",dt);
run;

%let ann10=%sysfunc(intnx(YEAR, &dt, 2, S));
%let ann10f=%sysfunc(intnx(YEAR, &dt, 2, S),mmddyy10.);

%put George and Debbie - 8 Years, &dt;
%put George and Debbie - 10 Years, &ann10;
%put George and Debbie - 10 Years, &ann10f;
```

```
438  %let ann10=%sysfunc(intnx(YEAR, &dt, 2, S));
439  %let ann10f=%sysfunc(intnx(YEAR, &dt, 2, S),mmddyy10.);
440
441  %put George and Debbie - 8 Years, &dt;
George and Debbie - 8 Years,        19498
442  %put George and Debbie - 10 Years, &ann10;
George and Debbie - 10 Years, 20228
443  %put George and Debbie - 10 Years, &ann10f;
George and Debbie - 10 Years, 05/20/2015
```

Wouldn’t this look bad engraved on jewelery!

Quoting functions

```
%put The author's 10th anniversary is &ann10f;
```

```
444  %put The author's 10th anniversary is
&ann10f;
```

Where’s the output?!

```
WARNING: The quoted string currently being processed has become
more than 262 characters long. You might have unbalanced quotation
marks.

NOTE 49-169: The meaning of an identifier after a quoted string
might change in a future SAS release.  Inserting white space between
a quoted string and the succeeding identifier is recommended.
```

```
%put The author%bquote(')s 10th anniversary is
&ann10f;
```

```
733  %put The author%bquote(')s 10th
anniversary is &ann10f;
The author's 10th anniversary is 05/20/2015
```

Quoting functions

%str	+ − * / < > = ¬ ^ ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN; as well as matched single quotes, double quote, or parenthesis
%nrstr	Same as %str, but also includes & and %
%quote	+ − * / < > = ¬ ^ ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN; as well as matched single quotes, double quote, or parenthesis
%nrquote	Same as %str, but also includes & and %
%bquote	' " ( ) + − * / < > = ¬ ^ ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN
%nrbquote	Same as %bquote, but also includes & and %
%superq	& % ' " ( ) + − * / < > = ¬ ^ ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN; however, it operates somewhat differently (see discussion)

What’s the difference between %str and %quote?

Quoting functions

%str family – compilation function
%quote family – execution function

```
%let vars=My 10th anniversary is %nrstr(&ann10f);
%let varb=My 10th anniversary is %nrbquote(&ann10f);
%put &vars;
%put &varb;
%put %nrstr(&vars);
%put %nrbquote(&vars);
```

```
79   %let vars=My 10th anniversary
is %nrstr(&ann10f);
80   %let varb=My 10th anniversary
is %nrbquote(&ann10f);
81   %put &vars;
My 10th anniversary is &ann10f
82   %put &varb;
My 10th anniversary is 05/20/2015
83   %put %nrstr(&vars);
&vars
84   %put %nrbquote(&vars);
My 10th anniversary is &ann10f
```

Quoting functions

%superq – works differently; useful to mask references arising within a macro variable

```
data storelist;
store='A&P';
output;
run;

data one; set storelist;
            call symput("store1", store);
run;

%let sq1=%superq(store1);
%let nb1=%nrbquote(&store1);
%let ns1=%nrstr(&store1);

%put &sq1;
%put &nb1;
%put &ns1;
```

This is not a mistake on my slide

```
 366
367  %let sq1=%superq(store1);
368  %let
nb1=%nrbquote(&store1);
WARNING: Apparent symbolic
reference P not resolved.
369  %let ns1=%nrstr(&store1);
370
371  %put &sq1;
A&P
372  %put &nb1;
A&P
373  %put &ns1;
&store1
```

Parmbuff

```
%macro runme / parmbuff;
%put syspbuff=&syspbuff;
        %if %length(&syspbuff)<=2 %then %put I have no friends;
        %else %do;
                %let num=1;
                %let dsname=%scan(&syspbuff,&num);
                %do %while(&dsname ne);
                        %put &dsname is my friend;
                        %let num=%eval(&num+1);
                %let dsname=%scan(&syspbuff,&num);
                %end;
        %end;
%mend runme;


%runme ;
%runme ();
%runme (Charlie);
%runme (Bob, Charlie);
%runme (Mary, Jonah, Marie);
```

Parmbuff

```
319  %runme ;
syspbuff=
I have no friends
320  %runme ();
syspbuff=()
I have no friends
321  %runme (Charlie);
syspbuff=(Charlie)
Charlie is my friend
322  %runme (Bob, Charlie);
syspbuff=(Bob, Charlie)
Bob is my friend
Charlie is my friend
323  %runme (Mary, Jonah, Marie);
syspbuff=(Mary, Jonah, Marie)
Mary is my friend
Jonah is my friend
Marie is my friend
```

Parmbuff

```
%macro runme (temp=3)/ parmbuff;
%put temp=&temp;
%put syspbuff=&syspbuff;
          %if %length(&syspbuff)<=2 %then %put I have no friends;
          %else %do;
                    %let num=1;
                    %let dsname=%scan(&syspbuff,&num);
                    %do %while(&dsname ne);
                              %put &dsname is my friend;
                              %let num=%eval(&num+1);
                    %let dsname=%scan(&syspbuff,&num);
                    %end;
          %end;
%mend runme;


%runme ;
%runme (temp=5);
%runme (Charlie);
%runme (temp=5,Charlie);
%runme (Charlie,temp=5);
```

Parmbuff

```
362  %runme ;
temp=3
syspbuff=
I have no friends
363  %runme (temp=5);
temp=5
syspbuff=(temp=5)
temp=5 is my friend
364  %runme (Charlie);
temp=3
syspbuff=(Charlie)
Charlie is my friend
365  %runme (temp=5,Charlie);
ERROR: All positional parameters must precede keyword parameters.
366  %runme (Charlie,temp=5);
temp=5
syspbuff=(Charlie,temp=5)
Charlie is my friend
temp=5 is my friend
```

Review

- %EVAL; %SYSEVALF
- %SYSFUNC
- Quoting functions
- Parmbuff option

Author Info

George J. Hurley
The Hershey Company
19 E Chocolate Ave.
Hershey, PA 17033

ghurley@hersheys.com