

A Collection of Tips for Advanced SAS® Programming: Macros and Hash

Darryl Prebble
Consultant
Prebble Consulting Inc.

SAS[®] Macro Programming

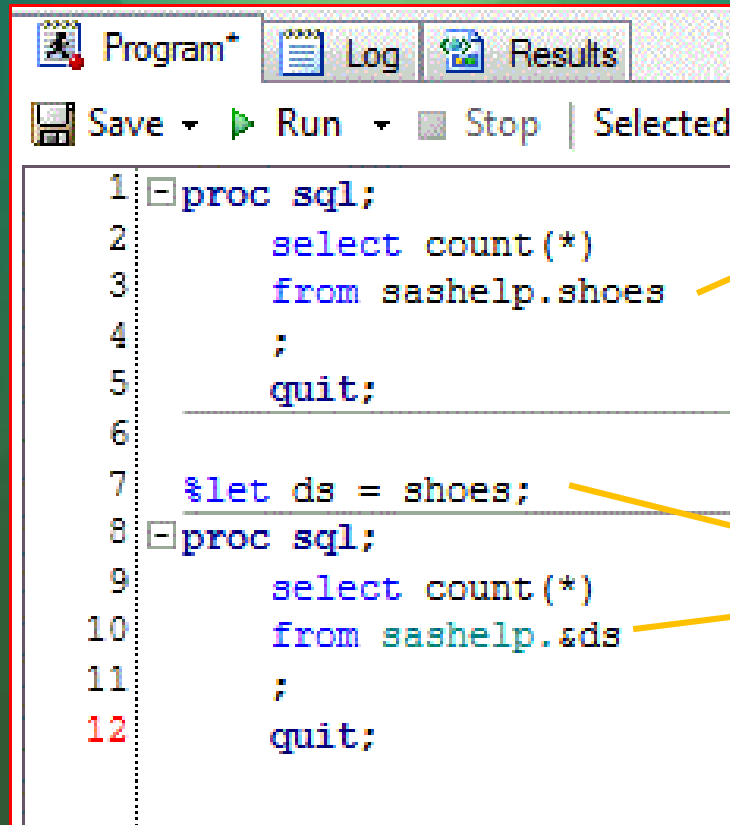
A Quick Primer...

- It's just text substitution
- Can be as simple as the substitution of a single variable to a dynamic generator of thousands of lines of code

SAS[®] Macro Programming

A Quick Primer...

- Simple text substitution:



```
1 proc sql;
2     select count(*)
3     from sashelp.shoes
4     ;
5     quit;
6
7     %let ds = shoes;
8 proc sql;
9     select count(*)
10    from sashelp.&ds
11    ;
12    quit;
```

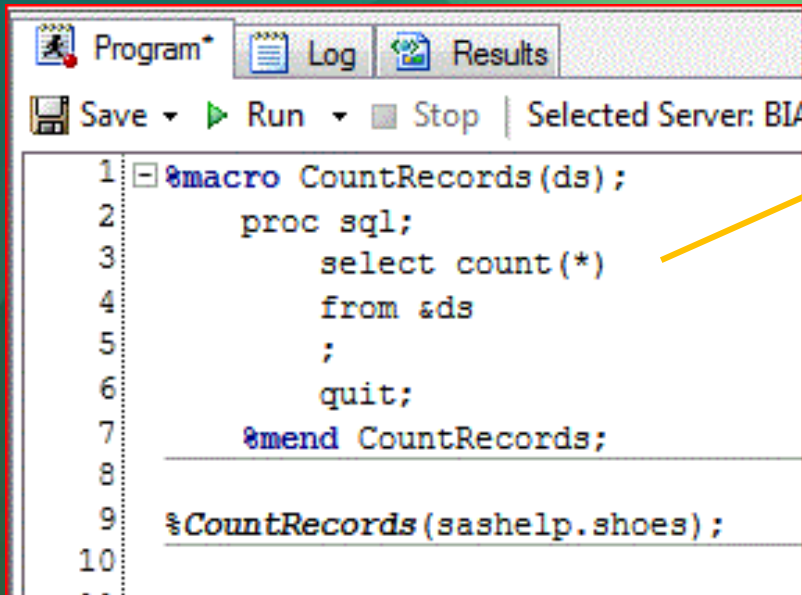
This...

...is the same as this!

SAS® Macro Programming

A Quick Primer...

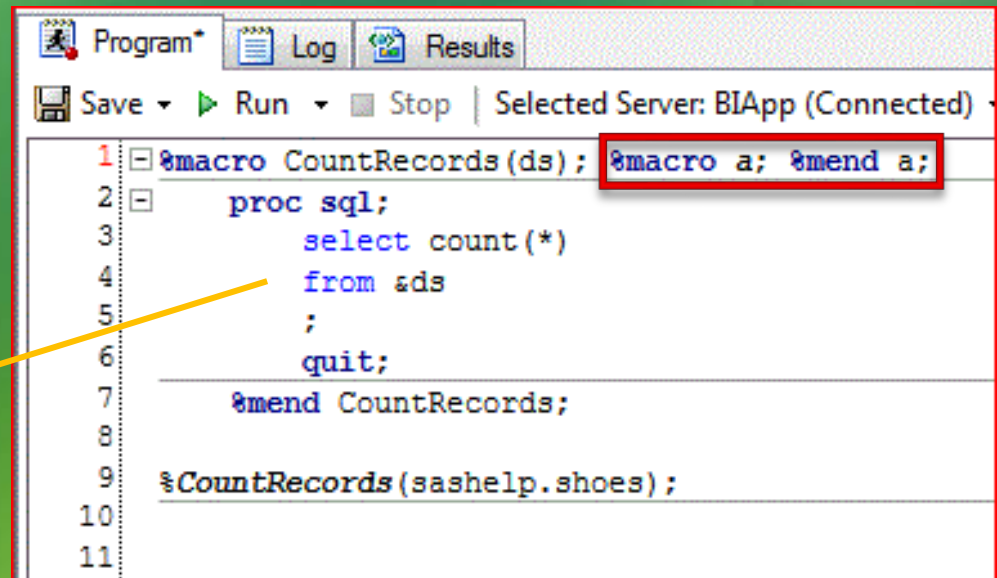
● Writing a Macro “Function”:



```
1 %macro CountRecords(ds);
2   proc sql;
3     select count(*)
4     from &ds
5     ;
6   quit;
7 %mend CountRecords;
8
9 %CountRecords(sashelp.shoes);
10
```

The screenshot shows a SAS IDE window with a menu bar (Program*, Log, Results) and a toolbar (Save, Run, Stop). The code editor displays a macro definition. The text is in a monospaced font but lacks the standard SAS color coding (e.g., blue for keywords, red for macro names). A yellow arrow points from the text 'Where did the colour coding go?' to the macro definition code.

Where did the colour coding go?



```
1 %macro CountRecords(ds); %macro a; %mend a;
2   proc sql;
3     select count(*)
4     from &ds
5     ;
6   quit;
7 %mend CountRecords;
8
9 %CountRecords(sashelp.shoes);
10
11
```

The screenshot shows the same SAS IDE window, but now the code is color-coded. The macro name `%macro` and `%mend` are blue, the macro name `CountRecords` is red, and the macro call `%CountRecords` is blue. A red box highlights the first line of code. A yellow arrow points from the text 'Ahhh.... Better. 😊' to the macro definition code.

Ahhh.... Better. 😊

Tip: %macro a; %mend a;

SAS® Macro Programming

A Realistic Example...

- Databases like Teradata and Netezza require constants (like date constraints) to be in single quotes – but macro variables won't resolve in single quotes...

```
%macro single(var);  
    %unquote(%str(%)&val%str(%))  
%mend;
```

```
%let datevar = 2015-01-01;
```

```
%proc sql;  
    /* connect to teradata yadda yadda */  
    create table temp as  
    select * from connection to teradata(  
        select *  
        from table  
        where date > %single(&datevar)  
    );  
    disconnect from teradata;  
quit;
```

This will resolve to: '2015-01-01'

SAS® Macro Programming

Using INTO for Data Driven Macro Variables...

```
proc sql;
  select count(*) into : record_count
  from sashelp.shoes
  ;
quit;
%put RECORD_COUNT ==> &record_count;
```

Single value, stored into a macro variable. Like a %let statement.

```
27          %put RECORD_COUNT ==> &record_count;
RECORD_COUNT ==>          395
```

```
proc sql;
  select distinct region
  into : region_list
  separated by ','
  from sashelp.shoes
  ;
quit;
%put REGION_LIST ==> &region_list;
```

Many values, into one variable.

```
34          %put REGION_LIST ==> &region_list;
REGION_LIST ==> Africa,Asia,Canada,Central America
States,Western Europe
```

SAS® Macro Programming

Using INTO for Data Driven Macro Variables...

- *So many* applications of this concept...
 - Pull SQL'ized variable lists that can then be used for table inserts.
 - Pulling max/min values for table driven constraints.

```
proc contents data = sashelp.shoes out=temp noprint;run;

proc sql noprint;
  select name into : varlist
         separated by ','
  from temp
  order by varnum
  ;
quit;

%put VARLIST ==> &varlist;
```

SAS® Macro Programming

Using INTO for Data Driven Macro Variables...

- *So many* applications of this concept...
 - Spin through using macro iteration like %DO %WHILE for data driven programming.

```
%let i=1;
%do %while(%qscan(%quote(&region_list), &i, ',') ~=);
  %let region = %qscan(%quote(&region_list), &i, ',');
  ...
  %let i = %eval(&i+1);
%end;
```

- Query a table of email addresses, and spin through them sending different data to different emails. All automated!

Hash Merging in SAS®

Changing gears here...

What is Hash Merging, and why do I care?

Hash Merging in SAS®

What is it?

● Cons:

- Not suitable for full table joins with lots of fields coming from both tables.
- Think of this as a Lookup Table on steroids – good for bringing a handful of fields from one table into another.
- Everything is loaded into RAM, so can be resource intensive. (Who cares?! Amiright SAS Admins??)
- Can hide data issues (ie. One-to-Many is baaaad)

Hash Merging in SAS®

What is it?

● Pros:

- In memory merging means, VERY FAST. I've seen 70% performance gains over SQL Joins and Sort/Sort/Merge methods!
- You can define multi-variable keys, which gives you an advantage over formats.
- Can hide data issues! (One-to-Many? Who cares!)
 - If the Hash object (ie. "Lookup table") is loaded with duplicate keys, only first record is retained. Order matters.

Hash Merging in SAS®

SQL Example

```
proc sql;  
    create table master as  
    select a.client_name  
           , a.sub_name  
           , b.*  
    from client_list a  
    inner join sales b  
        on(  
            a.client_id = b.client_id  
            and a.sub_id   = b.sub_id  
        )  
;  
quit;
```

Hash Merging in SAS®

Hash Example

```
data master;
  length client_name sub_name $50;
  if _n_ = 1 then do;
    declare hash h(dataset:'client_list');
    h.DefineKey('client_id', 'sub_id');
    h.DefineData('client_name', 'sub_name');
    h.DefineDone();
    call missing(client_name, sub_name);
  end;
  set sales;
  rc = h.find();
  drop rc;
run;
```

```
data master;
  %HashMerge(INTO=sales, FROM=client_list, KEY=client_id sub_id, VARLIST=client_name sub_name);
run;
```

Hash Merging in SAS®

Hash Tips

- `h.DefineKey('client_id', 'sub_id');`
 - Multi-key capability – advantage over formats
- `h.DefineData('client_name', 'sub_name');`
 - Multi-variable lookup – another advantage over traditional formats

Hash Merging in SAS®

Reference Material

- This is meant as an intro discussion of the concepts involved in Hashing. There is TONS more to learn.
 - <http://support.sas.com/rnd/base/topics/datastep/./hash-tip-sheet.pdf>
 - <http://www2.sas.com/proceedings/forum2007/271-2007.pdf>

Thanks!

Contact Info

Feel free to reach out any time!

dprebble@gmail.com