

Using SAS® Macro Language to Develop User-Written Functions

Deb Pine, Thotwave Technologies, Cary, NC

ABSTRACT

SAS Macro Language is a powerful tool that can be used to write macros that work very similar to SAS functions for the most common repetitive tasks. For example, a macro can be written that works just like a SAS function to return a value from a key in a lookup table. This approach to development paves the way for code reusability, an internal toolbox, and code standardization.

INTRODUCTION

The purpose of this paper is to explain how to write user-written functions using SAS Macro Language. The audience should understand BASE SAS and the SAS Macro Language.

WRITING THE FUNCTION

As developers we are constantly performing the same type of task over and over. For repetitive tasks, a macro can be used that returns a value to the calling program.

A SIMPLE FUNCTION

A task we can all relate to is retrieving the number of observations in a data set. The function below retrieves the number of observations and returns the value back to the calling program.

The source code looks like this:

```
%macro obs( DATA );

%local DSID NUM_OBS RC;
%let DSID =-1;

%if %exist(&data) %then
  %let DSID = %sysfunc(open(&DATA, I));

%if &DSID <= 0 %then %do;
  %let NUM_OBS = .;
  %put WARNING: Data set &DATA could
  not be opened. Unable to
  determine number of observations.;
%end;
%else %do;
  %let NUM_OBS = %sysfunc(attrn(&DSID,
NLOBS));
  %let RC = %sysfunc(close(&DSID));
%end;

&NUM_OBS

%mend obs;
```

The %obs macro works just like a SAS function in that it returns a value. Using this type of logic there is no need to explicitly assign a value to a macro variable and then explicitly check the value of the macro variable in the calling program.

In order for a user-written function to work properly the last line of code in the macro must be the return value. In %obs macro if any additional code or comments are put in after the line "&NUM_OBS" the macro will cause an error.

This is what the function looks like when it is called:

```
%if %obs(&dsn) = 0 %then
  %put No observations in data set.;
```

The function can also be used as a parameter passed into another macro call.

```
%errorcheck(%obs(&dsn),Unexpected number of
observations);
```

By creating a function to determine the number of observations in a data set the developer has just saved themselves 10+ lines of code. If you multiply that by 100 programs, the efficiency gain is clear.

A COMPLEX FUNCTION

This section takes a look at coding and using a more complex function than the example above. The macro below performs a routine task of a table lookup. This task can also be accomplished using PROC FORMAT or KEY= in a data step.

The macro below uses SAS Functions and call routines to perform a lookup instead of PROC FORMAT so that it is not affected when the underlying data changes. This approach is also favorable over the key= solution, as that requires the maintenance of an index.

```
%macro lookup(dsn=, n=);

%local rc DSID label table;
%let dsid =
%sysfunc(open(&dsn.(where=(N=&n.))));
%if &dsid >0 %then %do;
  %let rc = %sysfunc(fetch(&dsid));
  %let label =
%sysfunc(getvarc(&dsid,%sysfunc(varnum(&dsi
d,label))));
  %let rc = %sysfunc(close(&dsid));
%end;
%if &label = %str() %then %let label
=unknown;

&label

%mend lookup;
```

The macro %lookup opens the data set passed in as the parameter dsn and looks up the value for the variable label based on n.

Here is the data set used as a lookup table:

```
data fruit;
  input N label $ ;
  cards;
apple 1
banana 2
cherry 3
;
run;
```

This is what the function looks like when it is called:

```
%put The lookup return value is
%lookup(data=fruit,n=1);
```

In this example, this statement will return the value "apple".

CONCLUSION

User-written functions are a powerful tool for the SAS developer. An internal toolkit can be developed and matured over time saving the developer countless number of hours on repetitive tasks.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Deb Pine
Thotwave Technologies
2054 Kildaire Farm Rd. #322
Cary, NC 27511
Work Phone: 1.800.584.2819
Fax: 1.800.584.2819
Email: dpine@thotwave.com
Web: www.thotwave.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.