



THE  
POWER  
TO KNOW

## Blogs

All Topics ▾ All Industries

# SAS author's tip: Getting the macro language to perform a %DO loop over a list of values

Live Chat 

Cindy Puryear

on

SAS Learning Post

| January 30, 2015

This week's author tip is from Robert Virgile and his book "[SAS Macro Language Magic: Discovering and Developing the Power of the Macro Language](#)". I chose this tip because discovering and developing this technique will help you make the most of the macro language.

We hope you find this tip useful. You can also read [an excerpt](#) from Virgile's book.

### Macros Tip #1: A "Most Commonly Asked" Question

One macro language question arises again and again. How can I get macro language to perform a %DO loop over a list of values? For example, this syntax is illegal in macro language:

```
%do name=Amy, Bob, Fred;
```



But what if that is exactly what you need: execute a %DO loop for each name on the list? No limitation, as long as you program the logic yourself. Here are the steps you'll need to take.

First, create a macro variable holding the list of names. A %LET statement might look like this:

```
%let name_list = Amy Bob Fred;
```

Of course, there are many ways to generate a list of values. You could pass it as a parameter to a macro:

```
%call_my_macro (name_list=Amy Bob Fred)
```

Or, if a data set stores the names, SQL can retrieve them into a macro variable:

```
proc sql noprint;
  select distinct name into : name_list separated by ' ' from my.dataset;
quit;
```

Live Chat

Once the list of names is ready, execute the %DO loop for each name. A few considerations:

- While a variety of approaches exist, none of them take very long to execute. It might be faster than a faster-running method.
- All of these statements must appear within a macro definition. %DO loops and %LOCAL, etc., elsewhere.

Begin with a more complex approach:

```
%local i next_name;
%let i=1;
%do %while (%scan(&name_list, &i) ne );
  %let next_name = %scan(&name_list, &i);
  *** DO whatever needs to be done for &NEXT_NAME;
  %let i = %eval(&i + 1);
%end;
```



When &i is 1, %SCAN assigns the first word in &NAME\_LIST as the value of &NEXT\_NAME. After %SCAN reads all the words in &NAME\_LIST, it returns a null value and to read? As always, beauty is in the eye of the beholder.

An alternative approach has macro language count the number of names in the list. While € purpose, advances in macro language make counting words a straightforward task:

```
%local i next_name;
%do i=1 %to %sysfunc(countw(&name_list));
  %let next_name = %scan(&name_list, &i);
  *** DO whatever needs to be done for &NEXT_NAME;
%end;
```

The DATA step function COUNTW counts the number of words in a list, and %SYSFUNC p DATA step functions.

Live Chat

Of course, complications arise. Will your logic still work when:

- The list is empty?
- An item on the list is a significant word in macro language? For example, what if &NE

Those issues can arise, but are beyond the scope of this tip. This tip is to understand the st to process each item on a list of values.

For more information about the macro language and the magic you can create with it, check [Language Magic: Discovering Advanced Techniques](#)".

## Tags

## Share



## ABOUT AUTHOR

**Cindy Puryear**

Senior Marketing Specialist, SAS Publications

## 13 COMMENTS

shubham on April 2, 2019 9:53 am

GREAT WORK  
worked perfectly for me

Live Chat 

[Leonid Batkhan](#) on August 29, 2017 10:37 am

Very useful introduction to SAS macro loops! Here is a complementary blog post that [driven SAS macro loops](#).

Cindy on March 9, 2017 10:00 am

Jordan, the code featured in the blog doesn't represent the entire program that would because the sample statements within the blog cannot appear just anywhere. The prc includes the statements, and then call the macro (this is mentioned in one of the bulle explained the context briefly and focused more on the statements themselves. For mo <http://blogs.sas.com/content/sastraining/2015/01/30/sas-authors-tip-getting-the-macro-list-of-values/>. Thanks. Cindy

Jordan on March 4, 2017 9:23 pm

When I use this code, I get errors about using the %local, %do, and %end statements getting these?



Thanks!

---

Donald Armstead on February 23, 2017 12:43 am

Thank you Cindy! Very helpful. Cleaned up some very sloppy code for me.

---

Lisa on February 15, 2017 7:37 pm

Very helpful post! However, when I copied and pasted the SELECT...INTO code into realized the apostrophes pasted into EG as accent marks. Once I changed them to a wise for others who may try the same thing.

---

Cindy Puryear on February 20, 2017 9:04 am

Thank you Lisa for letting me know about this error along with how you fixed it.

---

Jason Van Slyke on June 9, 2015 4:12 pm

Thank you Cindy!

Every time I've needed to do this in %do things were so rushed I couldn't take time to way through it. I have saved this in my cheat sheet!

Thx, Jason

---

Cindy on June 10, 2015 2:02 pm

Jason,

Robert and I are so pleased this post helped you.

Cindy



Live Chat 

Al Barron on June 9, 2015 4:10 pm

Hi Cindy.

Suppose I were to assign the "Amy, Bob and Fred" values a font (say, their first+last r the font values through the macro call ?

thanks,  
Al

Cindy on June 16, 2015 2:51 pm

Al,

Robert and I appreciate your question and apologize for the delay. Robert was ask if you could be a little more specific on what you are trying to do?

Thanks. Cindy

Live Chat 

Ali on June 9, 2015 3:19 pm

Will you please explain %INCLUDE next time. Thanks.  
Ali

Cindy on June 16, 2015 2:53 pm

Great idea, Ali. Robert said to watch out for his next blog :)

Cindy



LEAVE A REPLY

Your Comment

Your Name

Your Email

Your Website

Live Chat 



