

## Paper 93-26

## Performing Multiple Statements for Each Record in a SAS® Data Set

Edward Moore, Highmark Blue Cross Blue Shield, Pittsburgh, PA

### ABSTRACT

A mainstay of database languages is the SCAN loop. This feature allows a programmer to loop through a table and perform tasks using the information in each record. SAS has similar functionality with the built-in loop of the DATA step. However, the constraints of DATA step programming greatly limit what tasks a programmer can perform. A programmer cannot, for example, begin a PROC step or another DATA step while looping within a DATA step. The limitations inherent in the DATA step can be frustrating constraints, but it is possible to create a work-around using macros and the %DO loop. To do this, a SAS program must pull one record at a time from a data set and then store any fields of interest within macro variables. Any program step can now access the contents of a record. Performing these tasks within a %DO loop provides a good simulation of the SCAN loop available in other database languages.

### PROCEDURE

A programmer can loop through each record in a SAS data set using a DATA step, but the programmer cannot begin any other program steps while within this loop. Similarly, a programmer can repeat a series of DATA and/or PROC steps a number of times using the %DO loop, but she lacks access to the contents of the records of another data set. By using both of these constructs, a programmer can read each record in a table and perform any number of program steps using the contents of the record.

In order to do this, a programmer must first obtain the number of records in a table and store this value to a macro variable. A %DO loop will be incremented once for each record, and it is necessary to know how many times this counter must be incremented. This following bit of code (modified from a similar bit of code from *In the Know... SAS Tips & Techniques From Around the Globe*) accomplishes that.

```
DATA _NULL_;
  IF 0 THEN SET FileName NOBS=X;
  CALL SYMPUT('RECCOUNT',X);
  STOP;
RUN;
```

This code stores the number of records in the data set `FileName` into a macro variable named `RECCOUNT`. Note that "IF 0 THEN" code was used to increase program efficiency. As noted in Mason's book, `NOBS` (number of observations) is set at compile time, and it is not actually necessary to perform the `SET` statement to obtain the number of observations.

Now that the programmer knows the number of observations in a table, she can use a %DO loop to increment from one to the number of observations. For each increment of the loop, one observation is drawn from the table. The proper observation is drawn using the `FIRSTOBS` option on the DATA step. Any variables that the programmer wishes other program steps to access can be stored to macro variables using `CALL SYMPUT`.

### STEPS TO IMPLEMENTING A SCAN LOOP IN SAS

1. Store the number of observations in a data set to a macro variable (`RECCOUNT`)
2. Increment a variable (`I`) from one to `RECCOUNT` using a %DO loop

3. Use the `FIRSTOBS` option in a DATA step to advance to the record `I`
4. Store record values to macro variables using `CALL SYMPUT`
5. Perform any desired PROC and DATA steps

### EXAMPLE

In the following example, a log file (`DATALOG`) has been created of all the data sets in a library named `TEST`. Periodically, it is necessary to print the contents of these data sets into a series of quick reports. `DATALOG` consists of two columns, `FILENM` (the file name) and `DESC` (a short description). The reports should be formatted with a standard PROC PRINT and have a title consisting of the table description. The following piece of code accomplishes that task.

---

```
/* Macro to SCAN through DATALOG */
%MACRO SCANLOOP(SCANFILE, FIELD1, FIELD2);
  /* First obtain the number of */
  /* records in DATALOG          */
  DATA _NULL_;
    IF 0 THEN SET &SCANFILE NOBS=X;
    CALL SYMPUT('RECCOUNT',X);
    STOP;
  RUN;

  /* loop from one to number of */
  /* records                      */
  %DO I=1 %TO &RECCOUNT;

    /* Advance to the Ith record */
    DATA _NULL_;
      SET &SCANFILE (FIRSTOBS=&I);

      /* store the variables */
      /* of interest in      */
      /* macro variables     */
      CALL SYMPUT('VAR1', &FIELD1);
      CALL SYMPUT('VAR2', &FIELD2);
      STOP;
    RUN;

    /* now perform the tasks that */
    /* wish repeated for each     */
    /* observation                  */

    PROC PRINT DATA=&VAR1;
      TITLE "&VAR2";
    %END;
  %MEND SCANLOOP;

  /* Call SCANLOOP */
  %SCANLOOP(DATALOG, FILENM, DESC);
  RUN;
```

---

The macro `SCANLOOP` provides a general framework for looping through a data set and performing a set of steps at each record. `SCANLOOP` is passed three variables: `SCANFILE` (the data set to be scanned), `FIELD1` (a variable in `SCANFILE` that you want to make available to other steps) and `FIELD2` (another variable that you want to make available). As described in the previous section, the number of records in the passed file name is stored to a macro

variable `RECOUNT`. The variable `I` is incremented from one to `RECOUNT`. A data step is used to advance to the `I`th record of the data set, and a number of values are stored in macro variables using `SYMPUT`. Finally, the desired task (in this case a `PROC PRINT`) is performed, and the whole cycle is repeated.

It should be noted that this example is quite simple, and this procedure can be generalized to any number of situations. A useful extension would be to use the `%IF..%ENDIF` statements to create the branching structures common in other database languages. For example, if `DATALOG` contained file size in addition to name and description, the above structure could be used to print only data sets of a certain size.

## REFERENCES

Mason, Phil; *In the Know... SAS Tips & Techniques From Around the Globe*; Cary, NC: the SAS Institute, 1996.

## ACKNOWLEDGMENTS

SAS is a registered trademark of SAS Institute Inc. in the United States and other countries. ® indicates registration in the United States.

I would like to thank Kathleen Maloney, JoAnn Matthews and Sharon Kijowski of Highmark Blue Cross Blue Shield and Kristin Corwin of the US Census Bureau for their suggestions and revisions.

## CONTACT INFORMATION

Edward Moore  
Customer Consulting  
Highmark Blue Cross Blue Shield  
Fifth Avenue Place  
120 Fifth Avenue, Suite P2501  
Pittsburgh, PA 15222  
(412) 544-2503  
edward.moore@highmark.com