Paper 33

The Power of PROC APPEND

Ted Logothetti, US Census Bureau

ABSTRACT

PROC APPEND is the fastest way to concatenate two SAS® data sets. This paper discusses some of the features of this procedure, including how much it lessens processing time, and some tips on how to use it. It also lists some limitations of the procedure.

INTRODUCTION

PROC APPEND is the fastest way to combine permanent SAS® data sets vertically. This paper begins by describing how the procedure works and why it is efficient. It demonstrates the power of this procedure by racing it against the SET statement in a DATA step. It shows the difference in processing time between concatenating ten large data sets using PROC APPEND versus concatenating the same data sets using the SET statement.

After discussing its power and uses, the paper explores the details of the syntax. It then gives some tips and tricks. Finally, it discusses some of the limitations of the procedure.

This paper is intended for an audience that knows the basic concepts of SAS®, such as variable types, the SET statement, concatenating data sets, and basic SAS® macro syntax.

WHAT PROC APPEND DOES AND WHY IT IS USEFUL

PROC APPEND does exactly what its name implies: it appends, or adds, observations from one data set to the end of another, target data set. It does this without reading observations from the target dataset. As a result, it is extremely efficient.

PROC APPEND Reads From One Data Set

This is may be the procedure's biggest advantage over other methods of concatenation. It reads only records that are being added to the base data set and not those from the base data set itself. This reduces I/O and processing time to a fraction of what would be required from other methods, especially if the base data set is large and the number of records being added is relatively small.

Imagine if you have to combine two data sets, where one has 9 million observations and the other has 1 million observations. If you combine the two data sets using the SET statement in a DATA step, SAS® has to read all 10 million observations. With PROC APPEND, SAS® has to read only 1 million observations, only a tenth as many. That lowers the I/O by 90 percent.

Block I/O Or Fast-Append Method

This advantage is often overlooked, but it is important. Instead of adding observations to a data set one at a time, PROC APPEND adds observations in blocks. This method reduces I/O and is a reason why PROC APPEND concatenates permanent SAS® data sets more efficiently than the SET statement. In a race between PROC APPEND and the SET statement, PROC APPEND always finishes faster. The next section details the race and its outcome.

Also, if the target data set has an index, it is updated after all observations have been added. Again, this reduces I/O.

THE RACE: PROC APPEND VS. SET STATEMENT

A simple test shows how much more efficient this procedure is at combining permanent SAS® data sets than the SET statement. In this test, a SAS® macro creates ten SAS® data sets. The data sets contain 15 variables each, four of which are numeric. Each data set contains one million observations. The code to create the data sets is below:

```
%macro datasets;

libname a 'h:\temp';
proc datasets kill lib=a;
quit;

      %do i=1 %to 10;

            data a.large&i;
                  length char1-char10 $ 10;
                  array chars{*} $ char1-char10;

                  do k=1 to 1000000;
                        do i=1 to 10;
                          lastone=ceil(10*ranuni(0));
                          do j=1 to lastone;
                                    byte=byte(48+floor(ranuni(0)*75));
                                    chars[i]=cats(chars[i],byte);
                          end;
                        end;

                        output;
                        call missing(of char1-char10);
                  end;
            run;

      %end;
%mend;

%datasets
```

The test then combines these ten large data sets into one data set in two ways. In the first, the data sets are combined using the SET statement in a DATA step:

```
data a.test_from_set ;
  set a.large:;
run;
```

In the second, the data sets are combined using a macro with PROC APPEND:

```
%macro append;

%do i=1 %to 10;
proc append base=a.test_from_append data=a.large&i;
run;
%end;
%mend;

%append
```

Following SAS®'s guidelines for testing, the each piece of code was submitted with different SAS® sessions to get an average. A program that concatenated the data sets with the SET statement was submitted in batch six times; another program that concatenated the data sets with PROC APPEND was submitted in batch six times. An average was then taken.

Table 1 shows the results of the processing times.

	CPU Time in Seconds		Percent Faster
	PROC APPEND	**SET Statement**	
Test 1	3.49	4.1	15
Test 2	3.37	4.53	26
Test 3	3.26	4.34	25
Test 4	3.79	4.71	20
Test 5	3.87	4.56	15
Test 6	3.57	4.57	22
Mean	*3.56*	*4.47*	*20*
Standard Deviation	0.22	0.20	-

Table 1. PROC APPEND vs. SET statement

The CPU times indicate that a programmer can lessen processing time by twenty percent using PROC APPEND instead of the SET statement. For that reason, PROC APPEND is a powerful tool.

Note that the efficiency comes from combining permanent SAS® data sets. The processing times are nearly identical if the test concatenates data sets from the WORK, or temporary, library.

BASIC SYNTAX

The syntax for PROC APPEND is simple. The procedure has only one statement.

Note that the APPEND statement in PROC DATASETS works exactly the same way as PROC APPEND. All options and tips described in this paper apply equally to PROC APPEND or PROC DATASETS with the APPEND statement.

The following comes from the online SAS® documentation:

PROC APPEND BASE=<libref.>SAS®-data-set <DATA=<libref.>SAS®-data-set> <FORCE> <APPENDVER=V6> <GETSORT>;

The table below, again from the online SAS® documentation, describes the options:

Task	**Option**
Add observations from one SAS® data set to the end of another SAS® data set	PROC APPEND
Add observations to the data set one at a time	APPENDVER=V6
Name of destination data set	BASE= (required)
Name of source data set	DATA=
Forces the append when variables are different	FORCE
Copies the sort indicator that was established by using PROC SORT from the DATA= data set to the BASE= data set	GETSORT

Task	Option
Suppresses the warning message when used with the FORCE option to concatenate two data sets with different variables	NOWARN

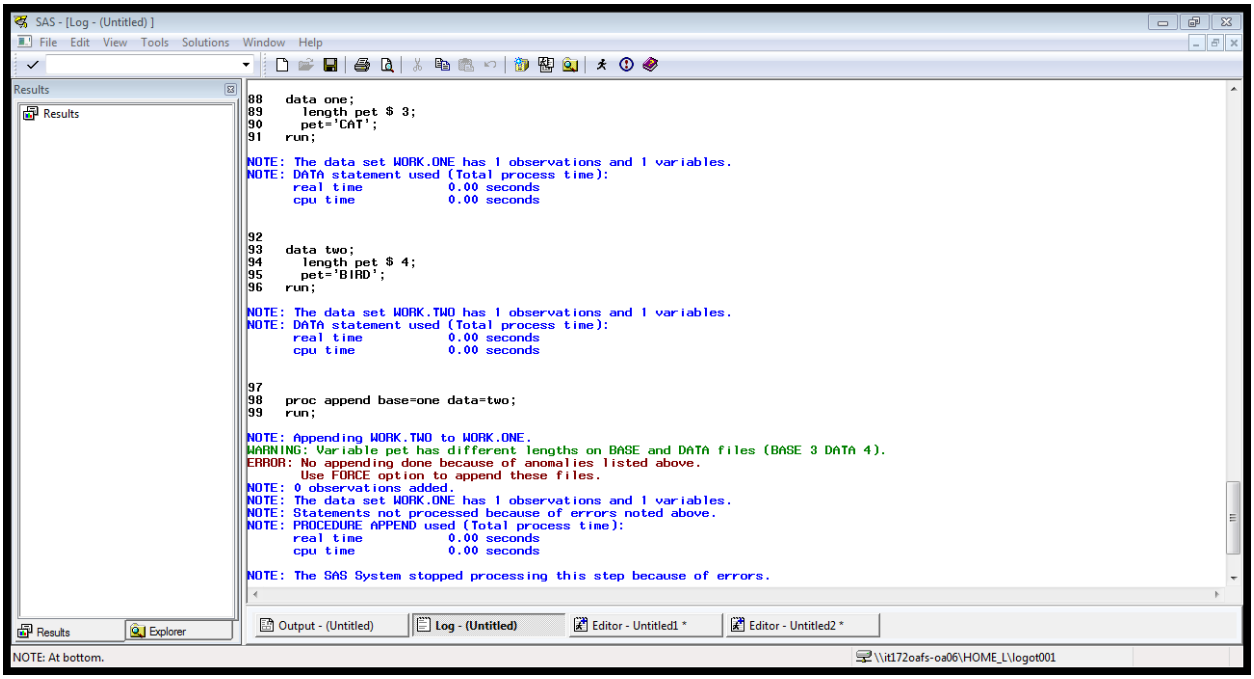
Table 2. Options in PROC APPEND (from https://support.sas.com/documentation)

The only required argument is the BASE= data set. The BASE= data set specifies the target or destination data set. You can think of it as the "output" data set.

The DATA= option specifies the data set that PROC APPEND combines with the target or output data set. The observations from this data set are attached to the bottom of the target data set. You can think of this as the source data set.

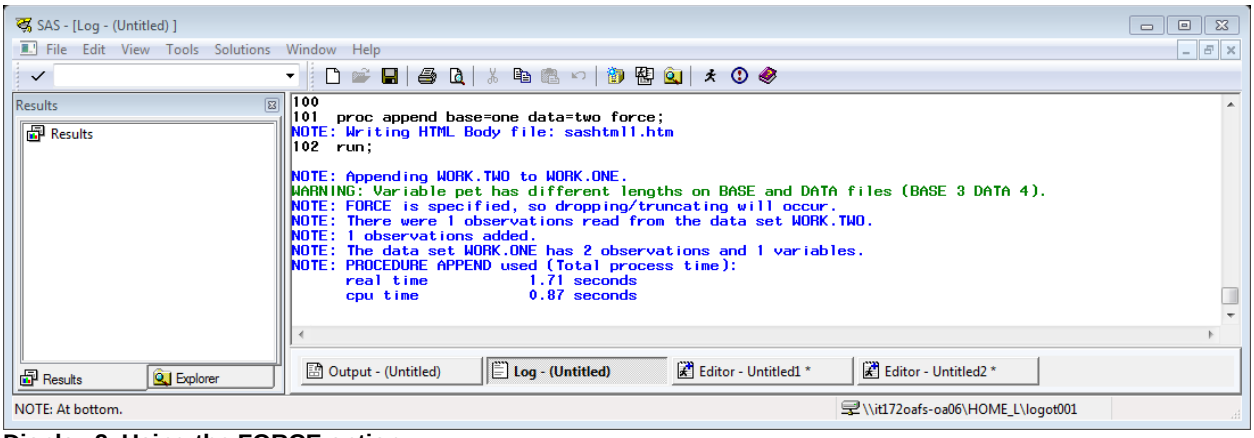
Two other options merit discussion, since they are used frequently. The FORCE option does what its name implies: it forces the two data sets, the source and the destination data sets, to be concatenated, whether they like it or not. PROC APPEND is unique in one respect: if the variables in the source data set differ in their name or type from the variables in the destination data set, PROC APPEND will throw an error message. In addition, if the length of a variable is shorter in the destination data set than it is in the source data set, PROC APPEND will generate an error message.

In the example below, the variable PET has a length of three in the destination data set ONE and a length of four in the source data set TWO. When SAS® tries to append TWO to ONE, it gives an ERROR message in the log and stops the procedure. Display 1 illustrates what happens in the log:



Display 1. Error Thrown When Variables Have Different Attributes

Just as the log states, the programmer can get around this by using the FORCE option. When SAS® tries to append with the force option, it will concatenate the data sets, but it will still issue a warning:



Display 2. Using the FORCE option

The FORCE option can be dangerous, as the log indicates. SAS® determines the length of the variables based on the size in the base or target data set. If a variable is shorter in the destination than it is in the source, it will be truncated. In this example, the value for the variable PET will be “BIR” instead of “BIRD”.

PROC APPEND also requires the exact same variables to be in both the source and destination data sets. If the two data sets do not have the same variables, SAS® will generate an error message. The FORCE option again can override this constraint. But SAS® will drop variables in the source data set that are not in the destination data set.

The NOWARN option is useful with the FORCE option. It tells SAS® not to write warning messages to the log if variables exist in one data set but do not exist in the other.

PROC APPEND has one other unique characteristic. Variables always assume the variable type that is in the destination data set. If the force option is used and the types differ, then the values from the source data set are set to missing.

For example, let’s say that variable X is in both the target and source data sets. In the target data set, it is character, but in the source data set, it is numeric. All the values in the source data set will be set to missing if the two data sets are combined using PROC APPEND with the FORCE option.

It is best to use PROC APPEND only with data sets whose variable characteristics are exactly the same. But what if the programmer is stuck with two data sets that do not have the same variables? Some suggestions are provided in the next section.

SOME TIPS AND TRICKS FOR USING PROC APPEND

The DROP= or KEEP= options allow the programmer to keep only compatible variables, or variables that are needed, in the source data set. If the programmer knows that the source data set contains variables that are not in the destination data set, he or she can use the KEEP or DROP option to exclude those variables. This would obviate the need to use the FORCE and NOWARN options. Note that the DROP and KEEP options work ONLY with the source data set and NOT with the destination data set.

WHERE statements also work with PROC APPEND. As with many procedures and the DATA step, the WHERE statement selects observations based on a certain condition. The WHERE statement applies only to the source data set and not to the target data set.

PROC APPEND also accepts WHERE data set options. As with the WHERE statement, the WHERE data set option works only with the source data set and not the target data set. While APPEND will not give an error message if the WHERE data set option is used with the target data set, it is completely useless, as SAS® will do no subsetting. In addition, it will slow down the PROC, since SAS® will not use the fast-method to append.

Note that the MSGLEVEL=I SAS® global option writes in the SAS® log about the append method in use. Its syntax is the following:

```
options msglevel=i;
```

LIMITATIONS

Despite its power, PROC APPEND has some limitations:

- It cannot create or modify variables. It does one thing only: it concatenates.
- The attributes of the variables in the two data sets must be exactly the same. While it does have the FORCE and NOWARN options, their use might be discouraged in a production environment. The DROP or KEEP options can exclude variables in the source data set that are not in the destination data set.
- It can append only two data sets at a time. Of course, a simple macro can resolve this issue by appending repeatedly.
- The programmer has to be careful about using it. Remember that PROC APPEND does not replace data sets, it only adds to them. If the programmer tests with it, then uses it again but forgets to wipe out the appended data sets, he or she will find many unwanted duplicates in the resulting data set.

CONCLUSION

PROC APPEND concatenates data sets efficiently and is easy to use. It is faster than the SET statement in the DATA step. It is limited, however, in how it can be used. It is most useful when the variables in both data sets have exactly the same characteristics.

REFERENCES

Carr, David W. 2008. “When PROC APPEND May Make More Sense Than the DATA STEP.” *Proceedings of the SAS® Global Forum 2008*. Paper 085-2008. Available at http://www2.sas.com/proceedings/forum2008/085-2008.pdf

Rhodes, Dianne Louise. "So You Want To Write A SUGI Paper? That Paper About Writing A Paper". *Proceedings of the SUGI 29.* Paper 145-29. Available at http://www2.sas.com/proceedings/sugi29/145-29.pdf

SAS® Institute. “Syntax: APPEND Procedure” 2009. Available at http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/viewer.htm#a000235213.htm

SAS® Institute. “The DATASETS Procedure: APPEND Statement” 2009. Available at http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/viewer.htm#a000235213.htm.

CONTACT INFORMATION

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

Name: Ted Logothetti
Enterprise: US Census Bureau
Address: 4600 Silver Hill Rd
City, State ZIP: Washington DC 20233

Work Phone: 301-763-1901
E-mail: Theodore.C.Logothetti@Census.gov