Paper CT-10

# Fatal Witlessness: Appending Datasets! WARNING! This may cause truncation of data!

Arunim Gupta, Independent Consultant, Bangalore, KA, India


## ABSTRACT

Datasets appended with the SET statement in a DATA step or by using the PROC APPEND procedure may or may not have a common source. When two data sets are appended, the programmer often assumes that the character variables in both datasets have the same lengths. If that is not true, SAS® commonly puts this message in the log: "WARNING: Multiple lengths were specified for the variable NAME by input data set(s). This may cause truncation of data." where NAME is a character variable with different length in each of the datasets being appended. Programmers who ignore this warning unknowingly leave themselves with truncated variables, which later on give erroneous and fatal results. This paper describes the problems caused by this issue and how to avoid them in simple cases. It then presents a completely automated (Macro) solution to the problem for situations that include large numbers of variables and large numbers of datasets.


## The Problem

Let us consider insurance datasets as shown below:

```
DATA DATA_1;
  INFILE DATALINES DELIMITER=',' DSD;
  INPUT NAME: $6. FUND_NAME: $15. BANK: $3. FUNDS ;
  DATALINES;
JOHNNY,GROWTH ULIP-BOA,BOA,60000
MARY,DEBT ULIP-BOA,BOA,90000
;
DATA DATA_2;
  INFILE DATALINES DELIMITER=',' DSD;
  INPUT NAME: $3. FUND_NAME: $20. BANK: $11. FUNDS ;
  DATALINES;
JOE,GROWTH ULIP-RELIANCE,WELLS FARGO,10000
JEF,SILVER FUND-RELIANCE,WELLS FARGO,20000
;
DATA DATA_3;
  INFILE DATALINES DELIMITER=',' DSD;
  INPUT NAME: $11. FUND_NAME: $20. BANK: $13. FUNDS ;
  DATALINES;
JOHNNYLEVER, GROWTH ULIP-RELIGARE, GOLDMAN SACHS, 80000
MARIGOLD, GOLD ULIP-GOLDMAN, GOLDMAN SACHS, 70000
;
```

We have three datasets, each containing following variables:

- NAME (of the account holder).
- FUND_NAME (fund in which account holder has invested).
- BANK (bank who has sold the policy) and ,
- FUNDS (amount invested).

Now, suppose we want to append these datasets so that we have a common dataset for all the customers of all the banks.

## Append Options in SAS®

### *Option 1: Using the SET Statement*

```
DATA APPEND_1;
SET DATA_1 DATA_2;
RUN;
```

Following is the warning given in the log when the above code is run:

```
WARNING: Multiple lengths were specified for the variable FUND_NAME by input data set(s). This may cause truncation of data.
WARNING: Multiple lengths were specified for the variable BANK by input data set(s). This may cause truncation of data.
```

And, following is the Output for the above run:

**APPEND_1**

| NAME | FUND_NAME | BANK | FUNDS |
|---|---|---|---|
| JOHNNY | GROWTH ULIP-BOA | BOA | 60000 |
| MARY | DEBT ULIP-BOA | BOA | 90000 |
| JOE | GROWTH ULIP-REL | WEL | 10000 |
| JEF | SILVER FUND-REL | WEL | 20000 |

Few Observations that we can deduce from the above output:

1. Variables FUND_NAME and BANK of the Second Dataset are truncated.
2. Variable NAME is intact in both the datasets.

This happened because the length of variable FUND_NAME and BANK in DATA_2 is more than that in DATA_1; while the length of the variable NAME is less in DATA_2 than in DATA_1.

A look at the *Proc Content* of appended data (APPEND_1) shows following lengths assigned to the variables:

Alphabetic List of Variables and Attributes

| # | Variable | Type | Len |
|---|---|---|---|
| 3 | BANK | Char | 3 |
| 4 | FUNDS | Num | 8 |
| 2 | FUND_NAME | Char | 15 |
| 1 | NAME | Char | 6 |

Looking at the attributes of variables in DATA_1, it can easily be seen that SAS® assigns the **same lengths** to the variables in the new appended dataset (APPEND_1) as in the **first dataset** mentioned in the SET Statement

Vice-Versa, if we mention DATA_2 as the first dataset in the SET Statement and DATA_1 as the second, we can avoid truncation in variables FUND_NAME and BANK, but variable NAME now gets truncated as can be seen in the table (APPEND_2) below.

**APPEND_2**

| NAME | FUND_NAME | BANK | FUNDS |
|---|---|---|---|
| JOE | GROWTH ULIP-RELIANCE | WELLS FARGO | 10000 |
| JEF | SILVER FUND-RELIANCE | WELLS FARGO | 20000 |
| JOH | GROWTH ULIP-BOA | BOA | 60000 |
| MAR | DEBT ULIP-BOA | BOA | 90000 |

### *Option 2: Using Proc Append (Without Force Option)*

We create a copy (APPEND_3) of the data set (in our case DATA_1) to which we want to append the second data (DATA_2).

```
DATA APPEND_3;
SET DATA_1;
RUN;
```

This has to be done since *Proc Append* is structured to have a base dataset, to which other data is appended. Next we append DATA_2 to APPEND_3 using *Proc Append*.

```
PROC APPEND BASE=APPEND_3 DATA=DATA_2;
RUN;
```

Given below is relevant part of log when we run the above statements:

```
NOTE: Appending WORK.DATA_2 to WORK.APPEND_3.
WARNING: Variable NAME has different lengths on BASE and DATA files (BASE 6 DATA 3).
WARNING: Variable FUND_NAME has different lengths on BASE and DATA files (BASE 15 DATA 20).
WARNING: Variable BANK has different lengths on BASE and DATA files (BASE 3 DATA 11).
ERROR: No appending done because of anomalies listed above. Use FORCE option to append these files.
NOTE: 0 observations added.
NOTE: The data set WORK.APPEND_3 has 2 observations and 4 variables.
NOTE: Statements not processed because of errors noted above.
NOTE: The SAS System stopped processing this step because of errors.
```

We see that above code has produced an error and subsequent to that it has stopped processing.

***Option 3: Using Proc Append (With Force Option)***

```
PROC APPEND BASE=APPEND_3 DATA=DATA_2 FORCE;
RUN;
```

In this case the code works, but gives the following warning in the log,

```
NOTE: Appending WORK.DATA_2 to WORK.APPEND_3.
WARNING: Variable NAME has different lengths on BASE and DATA files (BASE 6 DATA 3).
WARNING: Variable FUND_NAME has different lengths on BASE and DATA files (BASE 15 DATA 20).
WARNING: Variable BANK has different lengths on BASE and DATA files (BASE 3 DATA 11).
NOTE: FORCE is specified, so dropping/truncating will occur.
```

The output in this case is:

**APPEND_3**

| NAME | FUND_NAME | BANK | FUNDS |
|---|---|---|---|
| JOHNNY | GROWTH ULIP-BOA | BOA | 60000 |
| MARY | DEBT ULIP-BOA | BOA | 90000 |
| JOE | GROWTH ULIP-REL | WEL | 10000 |
| JEF | SILVER FUND-REL | WEL | 20000 |

A close look at the table confirms that APPEND_3 is same as APPEND_1.
Thus, in either case, whether we use the *SET statement* or the *Proc Append*, we are left with the same truncated dataset.

## Fatal Witlessness: Erroneous Results and Analysis

Let us now forget all the warnings given by SAS® and proceed with appending the datasets.

```
Data APPEND_4;
Set DATA_1 DATA_2 DATA_3;
run;
```

We get the following data after we run this code:

**APPEND_4**

| NAME | FUND_NAME | BANK | FUNDS |
|---|---|---|---|
| JOHNNY | GROWTH ULIP-BOA | BOA | 60000 |
| MARY | DEBT ULIP-BOA | BOA | 90000 |
| JOE | GROWTH ULIP-REL | WEL | 10000 |
| JEF | SILVER FUND-REL | WEL | 20000 |
| JOHNNY | GROWTH ULIP-REL | GOL | 80000 |
| MARIGO | GOLD ULIP-GOLDM | GOL | 70000 |

Few Observations on this table:

- There are no BANKS like, 'WEL' and 'GOL', in the individual data sets as we get to see in this table.
- The total FUNDS in Account Holder JOHNNY'S account as per this table is 140000, whereas the actual amount in JOHNNY'S account should be 60000. This happens because JOHNNYLEVER in third dataset gets truncated as JOHNNY. While JOHNNY is a valid account holder name, and it makes sense when JOHNNY appears instead of JOHNNYLEVER under the column 'NAME', there is no such person as MARIGO in the original datasets. We get records corresponding to a person who doesn't exist!

- A careful look at FUND_NAME suggests that two very distinct funds in our original datasets, GROWTH ULIP-RELIANCE and GROWTH ULIP-RELIGARE have come out to be a single fund, 'GROWTH ULIP-REL', in the final dataset. A usual approach when truncation happens is to map the truncated data with the original values after, of course, figuring out the basis of truncation (in this case, it is the leftmost 15 characters in FUND_NAME which appears in our appended datasets). However in present scenario, even this is not possible, as there are two unique values corresponding to 'GROWTH ULIP-REL', namely, 'GROWTH ULIP-RELIANCE' and 'GROWTH ULIP-RELIGARE'. Moreover, the sample datasets contain just two funds in each. More usually, there would be thousands of funds floating in the market. Imagine having such a dataset with truncated fund names! Under that scenario, even if someone chose to not ignore the warning, it would not be humanly possible to rectify the column values.

## A Simple, but not a Workable Solution: LENGTH STATEMENT

```
DATA DATA_1;
LENGTH NAME $11. FUND_NAME $20. BANK $13. ;
SET DATA_1;
RUN;
DATA DATA_2;
LENGTH NAME $11.BANK $13. ;
SET DATA_2;
RUN;
DATA Append_5;
SET DATA_1 DATA_2 DATA_3;
RUN;
```

**APPEND_5**

| NAME | FUND_NAME | BANK | FUNDS |
|---|---|---|---|
| JOHNNY | GROWTH ULIP-BOA | BOA | 60000 |
| MARY | DEBT ULIP-BOA | BOA | 90000 |
| JOE | GROWTH ULIP-RELIANCE | WELLS FARGO | 10000 |
| JEF | SILVER FUND-RELIANCE | WELLS FARGO | 20000 |
| JOHNNYLEVER | GROWTH ULIP-RELIGARE | GOLDMAN SACHS | 80000 |
| MARIGOLD | GOLD ULIP-GOLDMAN | GOLDMAN SACHS | 70000 |

A simple solution is to look for ' Maximum Length' of a variable in all the datasets to be appended, and then assign that length to the variable, using the *LENGTH statement.* This needs to be done for each of the variables in the datasets.

Why is this not a workable solution? Because there can be thousands of datasets with thousands of variables in them, and finding Maximum Length of each variable, and writing *LENGTH statements* for each of the datasets won't be an easy task !

## The Solution: APPENDMAC, an Automated Macro Code

```
/**********************APPENDMAC MACRO CODE BEGINS************************/
%MACRO APPENDMAC(INFILE1,INFILE2,OUTFILE);
%GLOBAL NUMOFVAR;
%GLOBAL VARLIST;
%GLOBAL COUNT_1;
%GLOBAL COUNT_2;
%GLOBAL COUNT_VAR;
%LET VARLIST=;
%MACRO PROPMAC(F1,F2,OUT);
PROC CONTENTS DATA=&F1 NOPRINT
OUT=FF_POS(KEEP=NAME VARNUM );
RUN;
PROC SORT DATA=FF_POS;
BY VARNUM;
RUN;
%COUNTOBS(FF_POS);
%LET COUNT_1=&COUNT_VAR;
%FORMAT_MAC1(FF_POS);
PROC CONTENTS DATA=&F1 NOPRINT
OUT=FF_PROP(KEEP=NAME TYPE LENGTH WHERE=(TYPE=2));
RUN;
PROC CONTENTS DATA=&F2 NOPRINT
OUT=SF_PROP(KEEP=NAME TYPE LENGTH WHERE=(TYPE=2));
RUN;
PROC SORT DATA=FF_PROP;
BY NAME;
RUN;
PROC SORT DATA=SF_PROP;
BY NAME;
RUN;
DATA MF_PROP;
MERGE FF_PROP(RENAME=(LENGTH=LEN_FF)) SF_PROP(RENAME=(LENGTH=LEN_SF));
BY NAME;
MAX_LEN=MAX(LEN_FF,LEN_SF);
RUN;
%LENGTHMAC();
%COUNTOBS(MF_PROP);
%LET COUNT_2=&COUNT_VAR;
%AMEND(&F1);
%AMEND(&F2);
DATA &OUT;
RETAIN &VARLIST;
SET &F1 &F2;
RUN;
%MEND PROPMAC;
%MACRO FORMAT_MAC2();
%DO I=1 %TO &COUNT_1;
   %LET VARLIST=&VARLIST. &&NAME&I;
%END;
%MEND FORMAT_MAC2;
%MACRO FORMAT_MAC1(DATA);
DATA _NULL_;
SET &DATA;
CALL SYMPUT('NAME'||LEFT(_N_),NAME);
RUN;
%FORMAT_MAC2();
%MEND FORMAT_MAC1;
%MACRO LENGTHMAC();
DATA _NULL_;
SET MF_PROP;
CALL SYMPUT('NAME'||LEFT(_N_),NAME);
CALL SYMPUT('MAX_LEN'||LEFT(_N_),MAX_LEN);
RUN;
```

```
%MEND LENGTHMAC;
%MACRO COUNTOBS(CNTDATA);
DATA _NULL_;
SET &CNTDATA END=EOF;
COUNT+1;
IF EOF THEN CALL SYMPUT('COUNT_VAR',COUNT);
RUN;
%MEND COUNTOBS;
%MACRO AMEND(DATA);
%DO I=1 %TO &COUNT_2;
   %PUT DATA &DATA %STR(;);
        %PUT %CMPRES(LENGTH &&NAME&I %QTRIM($&&MAX_LEN&I.%STR(.;)));
        %PUT SET &DATA %STR(;);
        %PUT RUN %STR(;);
    DATA &DATA ;
         %CMPRES(LENGTH &&NAME&I %QTRIM($&&MAX_LEN&I));
         SET &DATA ;
         RUN ;
  %END;
%MEND AMEND;

%PROPMAC(&INFILE1,&INFILE2,&OUTFILE);
%MEND APPENDMAC;
/************************APPENDMAC MACRO CODE ENDS************************/


/*SAMPLE USER INPUTS*/
%APPENDMAC(INPUTFILENAME1,INPUTFILENAME2,OUTPUTFILENAME);
```

## In Action- APPENDMAC

```
%appendmac(DATA_1, DATA_2,APPEND_INTERMEDIATE);/*FIRST CALL*/
%appendmac(APPEND_INTERMEDIATE, DATA_3,APPEND_6);/*SECOND CALL*/
```

**APPEND_6**

| NAME | FUND_NAME | BANK | FUNDS |
|---|---|---|---|
| JOHNNY | GROWTH ULIP-BOA | BOA | 60000 |
| MARY | DEBT ULIP-BOA | BOA | 90000 |
| JOE | GROWTH ULIP-RELIANCE | WELLS FARGO | 10000 |
| JEF | SILVER FUND-RELIANCE | WELLS FARGO | 20000 |
| JOHNNYLEVER | GROWTH ULIP-RELIGARE | GOLDMAN SACHS | 80000 |
| MARIGOLD | GOLD ULIP-GOLDMAN | GOLDMAN SACHS | 70000 |

***Some Salient Features of APPENDMAC:***

- It takes two inputs at a time.
- It can be used to append any number of datasets. For this, output of one call of *APPENDMAC* should be used as one of the inputs to second call of APPENDMAC, whereas second input should be another of the datasets we intend to append. In present case, output of first call, APPEND_INTERMEDIATE (output when DATA_1 and DATA_2 are appended) is given as first input to second call of APPENDMAC, whereas second input is DATA_3, which then gives final appended data set, APPEND_6 (This is the data which one gets after appending DATA_1, DATA_2 and DATA_3).
- The order of variables in the output is same as that in the first dataset mentioned in the call of APPENDMAC. (Leftmost dataset mentioned in any call of APPENDMAC).
- APPENDMAC has been structured to output all the length modification to be outputted in the *LOG*, so that the user knows what all alterations are taking place in their Datasets.

## CONTACT INFORMATION

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

Arunim Gupta,

Villa No-119,Himagiri Meadows,

Banerghatta Road,Bangalore-560083,

Karnataka,India

Email:arunimgupta@gmail.com

## TRADEMARKS