

Calculating the Most Expensive Printing Jobs

Roger L. Goodwin

I. INTRODUCTION

As the SAS manual states, the macro facility is a tool for extending and customizing SAS and for reducing the amount of text that the programmer must enter to do common tasks [SAS Institute, (2)]. Programmers use SAS macros for mundane, repetitive tasks. In this application, we present a SAS macro that calculates the top twenty most expensive printing jobs for each Federal agency.

Given the request for the top twenty most expensive printing jobs, it became apparent that this request would become repetitive [H. Paulson (1)]. The US Government Printing Office anticipated an increase for financial summaries from government agencies and developed the following SAS macro specifically for Treasury's request. GPO has contracts with most Federal government agencies to procure print. GPO can produce a report for the top twenty most expensive printing jobs for each Federal agency. This, of course, assumes the agency does business with GPO.

II. THE FIELDS OF INTEREST

This Section presents the fields on the different input files. Some of these fields can only be found on a single file. Other fields are common to multiple files.

Strategically, it makes sense that we will sum by a common variable or set of common variables to satisfy Treasury's request. For commercial print jobs, the jacket number, print order number and fiscal year provides a unique combination to identify a financial transaction with an accompanying contract. This concept does exclude contract modifications, re-billings, and other un-foreseen circumstances. In such cases, the jacket number, print order number and fiscal year repeat for identification purposes.

A. Common Fields

For commercial print jobs, there are some common fields between both Accounts Payable and Accounts Receivable. Those fields include:

- 1) Jacket number and print order number.
- 2) Requisition number.
- 3) Fiscal year of jacket.

B. Accounts Payable (AP) Fields

This Section lists some fields more commonly found in the historic Accounts Payable data files. These data files are called VOPPS. VOPPS stands for Voucher Payment and Processing System [US Government Printing Office (4)]. Before the new Oracle system was developed and deployed, the VOPPS system recorded payments in process for contractors. The revised 2014 GPO Printing Procurement Regulations do not mention the VOPPS system [US Government Printing Office (5)]. The regulations still reference progress payments and vouchers. The record layout for VOPPS data dates as far back as October 1992.

- 1) Contractor code.
- 2) Contractor name.
- 3) Contractor street address.
- 4) Contractor city, state, and zip code.
- 5) Amount paid to the contractor.
- 6) Date paid.

C. Accounts Receivable (AR) Fields

This Section lists some fields more commonly found in the historic Accounts Receivable data files. These data files are called JBAC. JBAC stands for Jacket and Billing Address Code.

- 1) Agency Location Code.
- 2) Agency Billing Address Code.
- 3) Amount invoiced.
- 4) Date of invoice.

For a given Fiscal Year of AR invoices, the AP invoices can appear either in the same Fiscal Year or in a prior Fiscal Year. The AP invoice dates are less-than or equal-to the AR invoice dates. GPO pays the contractor before billing the agency.

III. MACRO FLOW CHART

This Section gives an overview of the SAS macro used to calculate the most expensive printing jobs for government agencies with commercial contracts with GPO. The flow chart of the macro appears in Figure 1. An example of the call to the macro REPORT appears next:

```
%macro report;
```

```
...
```

```
%mend report;
```

```
%report();
```

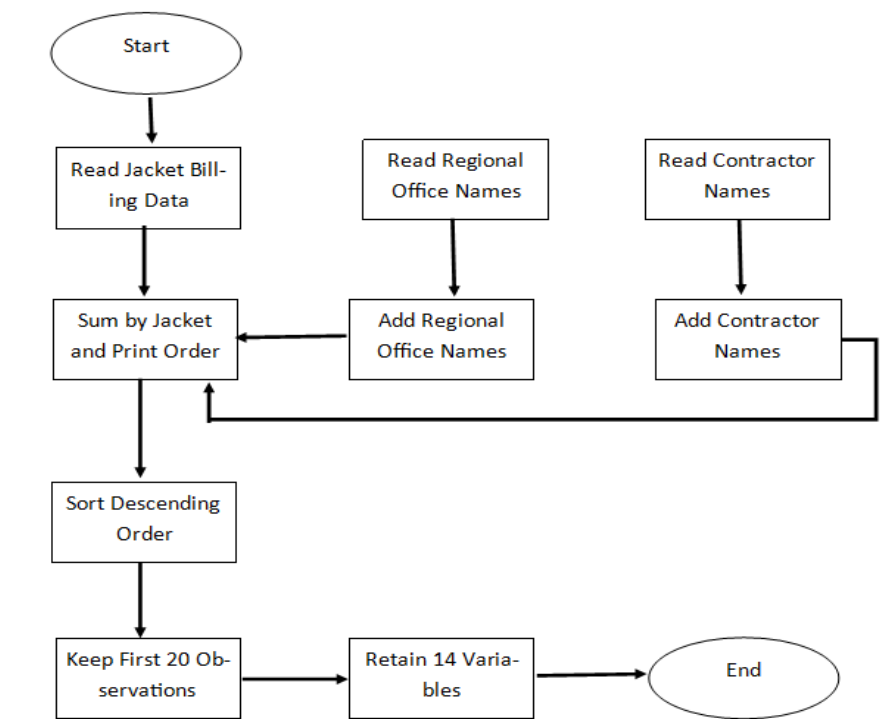


Fig. 1. This figure shows the flowchart of the SAS macro.

- Three macro variables.
- Two library references.
- Nine data steps.
- A single PROC MEANS.
- A single PROC EXPORT.

In the list above, two of the data steps are conditioned on the FY macro variable. This allows us to read the billing data, variably. For instance, If an agency requests their most expensive 2006 printing jobs, then the macro will select the 2006 JBAC files. If an agency requests their most expensive 2007 printing jobs, then the macro will select the 2007 JBAC files. We assume that print contractors billed GPO in the given Fiscal Year or prior to the given Fiscal Year.

The list above contains a single PROC MEANS to calculate the total dollars and number of orders.

The list above contains a single PROC EXPORT to save the output file as an Excel spreadsheet. At most, the SAS macro saves twenty observations.

As stated earlier, the benefit for creating the SAS macro using the macro facility is for reducing the amount of text that the programmer must enter to do common tasks. The same concept applies to the use of libraries. We grouped the input files and assigned a shared name. If the location of the files changed, then only one line of code needs to change versus changing multiple lines of code. We will discuss some examples next.

IV. LIBRARIES

The user defined SAS macro %REPORT() references two libraries. The libraries make the code easier to modify if the directories change. Instead of changing multiple lines of SAS code, the programmer only needs to change two lines of SAS code.

- IN — This library name contains the AP data files with the numeric data. This is also the output directory.
- IN2 — This library name contains the AR data files with the complete agency name and billing information.

Example 1: Experience suggests writing the data step with the library call to IN2. Should the input directory change, the programmer would only change one line of source code.

```
libname in2 "c:\pics_vopps\";

...

%IF &fy EQ 2007 %THEN %DO;
DATA jbac(KEEP = jacket bac bac4 print_order billing_amount quantity); *2007;
SET in2.jbac51m_oct06
    in2.jbac51m_nov06
    in2.jbac51m_dec06
    in2.jbac51m_jan07
    in2.jbac51m_feb07
    in2.jbac51m_mar07
    in2.jbac51m_apr07
    in2.jbac51m_may07
    in2.jbac51m_jun07
    in2.jbac51m_jul07
    in2.jbac51m_aug07
    in2.jbac51m_sep07;

...

RUN;
%END;
```

Example 2: This is an example of the excessive coding. Should the input directory change, the programmer would have to change twelve lines of source code.

```
%IF &fy EQ 2007 %THEN %DO;
DATA jbac(KEEP = jacket bac bac4 print_order billing_amount quantity); *2007;
SET "c:\pics_vopps\jbac51m_oct06"
    "c:\pics_vopps\jbac51m_nov06"
    "c:\pics_vopps\jbac51m_dec06"
    "c:\pics_vopps\jbac51m_jan07"
    "c:\pics_vopps\jbac51m_feb07"
    "c:\pics_vopps\jbac51m_mar07"
    "c:\pics_vopps\jbac51m_apr07"
    "c:\pics_vopps\jbac51m_may07"
    "c:\pics_vopps\jbac51m_jun07"
    "c:\pics_vopps\jbac51m_jul07"
    "c:\pics_vopps\jbac51m_aug07"
    "c:\pics_vopps\jbac51m_sep07";

...

RUN;
%END;
```

There are two benefits to using library names in SAS:

- 1) Grouping of similar files.
- 2) Minimization of software maintenance

V. MACRO VARIABLES

We programmed three macro variables to simplify coding and future changes. We use the macro variables throughout the program to allow flexibility in changing the input files from fiscal-year to fiscal-year; obtaining the contractor names and addresses; and changing the output file name according to the Fiscal Year. The macro variable names are as follow:

- FY — The Fiscal Year of the AR invoice.
- CCODES_PNUMBERS — The input file name of AP invoices.
- C NAMES — The input file of contractor names and addresses.

Example 3: With the use of the macro variable &fy, the following code merges the data for Fiscal Year 2007.

```
%LET fy = 2007;

...

DATA eda;
MERGE branch_sums(IN=ina) agency_names(IN=inb);
BY bac;
IF ina;
my_fy = "&fy";
req_number = COMPRESS(req_number, "/-");
RUN;
```

Example 4: Without the use of the macro variable, the SAS code would be as follow. The programmer would have to remember to change every instance of "2007" in the program when an agency made a request for a different Fiscal Year.

```
DATA eda;
MERGE branch_sums(IN=ina) agency_names(IN=inb);
BY bac;
IF ina;
my_fy = "2007";
req_number = COMPRESS(req_number, "/-");
RUN;
```

VI. AGENCY ESTIMATES

This section shows how to calculate the top twenty printing jobs per agency. Recall there were two common fields between the AP files and the AR files. Those fields were the jacket number and the print order number.

```
PROC MEANS DATA = jbac SUM NOPRINT;
BY jacket print_order;
ID bac bac4;
VAR billing_amount quantity;
OUTPUT OUT = est_sumed SUM = sum_billing sum_qnty;
RUN;
```

We used the PROC MEANS presented to summarize "the most expensive printing" paid by agencies. Data steps added afterwards add contractor information and additional agency information. The additional information is more human readable, than say, contractor codes and billing address codes. Naturally, the phrase "the most expensive printing" assumes the agency submits print business to GPO.

VII. SUMMARY

We covered the advantages of using the LIBNAME statement and the advantages of using user defined SAS macro variables. Both reduce the amount of text that the programmer must enter to do common tasks. The LIBNAME statement and the SAS macro variables also minimize software maintenance in the event the code has to be changed.

VIII. REFERENCES

- 1) H. Paulson (US Secretary of the Treasury), Untitled Letter to Agencies, *Subcommittee on Federal Financial Management, Government Information, Federal Services and International Security*, June 18, 2008.
- 2) SAS Institute, *SAS(R) 9.4 Macro Language: Reference*, July 2013.
- 3) SAS Institute, *SAS/ACCESS(R) 9.4 Interface to PC Files: Reference, Second Edition*, December 2013.
- 4) US Government Printing Office, *Printing Procurement Regulation*, May 1999, page VI-6.
- 5) US Government Printing Office, *Printing Procurement Regulation, Chapter XIII. Contract Administration And Compliance*, April 2014.

IX. ABOUT THE AUTHOR

Roger Goodwin has 15 years experience with several government agencies. Two of the agencies are statistical in nature; the third agency is both production and commercial in nature. Roger Goodwin completed statistical assignments on many computer platforms, which include PCs, Vax VMS OS, Unix OS, and IBM mainframes. He usually performs his statistical analyses in SAS, but has used Excel for simpler calculations. He developed reports for cost, progress, and billing reports using SAS and SAP Business Objects.

Roger Goodwin holds a BS in Computer Science and an MS in Applied Statistics from Old Dominion University. He completed a certificate in Software Engineering Processes from Learning Tree. He completed the Project Management Professional certification from PMI. He authored several papers in IEEE conferences and two online journals that summarize his experiences in government. He authored papers in the North East SAS Users Group that describes some of the SAS code that he wrote.

X. CONTACT INFORMATION

Affiliate:

US Government Printing Office

Roger L Goodwin

Email: rgoodwin AT gpo DOT gov

732 North Capitol Street NW

Room C-628

Washington DC 20401

Roger L Goodwin

Email: rogergoodwin AT frontiernet DOT net

1989 Leetown Road

Summit Point, WV 25446