

Printing Avery[®] Sticky Labels with SAS[®] and L^AT_EX

Adeline J. Wilcox
US Census Bureau, Washington, DC

ABSTRACT

SAS[®] and L^AT_EX can be used together to print sticky labels when there are too many labels to print with a point-and-click application. File folder labels needed to be created to organize production of 2010 Census questionnaires and other printed materials to be used in our decennial census. Printed materials were listed in a column of an Excel spreadsheet. After exporting the worksheet to CSV file format, I deleted unneeded rows with my favorite text editor, `gvim`. I read the CSV file into SAS, then wrote it out to an ASCII file containing L^AT_EX markup codes. In the L^AT_EX users' literature on the Web, I found a paper with L^AT_EX code that I was able to modify for my purpose. The example code showed me how to read the ASCII file I wrote with SAS into L^AT_EX. Running my L^AT_EX program successfully, I output a PDF file that, routed to the laser printer, produced sticky labels.

INTRODUCTION

Many people produce Avery[®] sticky labels by entering the text to be printed on the labels into Microsoft[®] Word. A point and click method for printing sticky labels from a Microsoft[®] Access[™] database can be found on Microsoft's web site [Microsoft, 2008]. The instructions run six pages, the length of this paper.

When many labels need to be printed, SAS users may prefer to code a more automated process. At SUGI 29, Kirk Paul Lafler recommended SAS PROC FORMS for printing labels [Lafler, 2004]. Another SAS user, Dongming Jiang, found that PROC FORMS did not always print Avery labels properly [Jiang, 2006]. He described using SAS without PROC FORMS to print Avery labels. My paper describes how I used Base SAS and L^AT_EX to print Avery 5029[™] file folder labels. For the labels I needed to print, I used fewer lines of SAS and L^AT_EX code total than Dongming Jiang needed using SAS alone.

SAS INPUT FILE

The information I needed to print on sticky labels came to me in an Excel spreadsheet. I saved its contents to a CSV file which I edited with my favorite text editor, `gvim`. Working with a copy of the CSV file that I named `fewer.csv`, I used `gvim` to delete rows that I did not need to print on sticky labels. I also used my text editor to make two more changes to prepare `fewer.csv` to be read by SAS. First, I corrected a few misspellings. Second, to the four records with an ampersand in the form description, I preceded the ampersands with a backslash, `\`, the escape character for L^AT_EX.

SAS PROGRAM

The first `filename` statement in my SAS program names the file `fewer.csv` that I read the names of the census questionnaires from. Note that I did not edit all unneeded material from the input file because I knew I could tell SAS to read only two columns of the spreadsheet. The first two variables read by my `input` statement are `junk1` and `junk2` because Some of the printed material descriptions in the spreadsheet were too many characters in length to print on the file folder labels. I used the `substr` function in the DATA step named `shorter` to trim them to a length of 44 characters. To check this work, I used `proc print` to list the truncated descriptions.

The first `data _null_` step lists only the official form numbers of the printed materials. I printed the file named `labsmade1.txt` to serve as an easily searchable reference list of the forms for which I created file folder labels.

The second `data _null_` prints the file named `aver5029.txt`, to be read by `LATEX`. The `put` statement in the second `data _null_` prints the `LATEX` commands `\bfseries` and `\normalfont`. Both these commands are surrounded by single quotes telling SAS to print them to the output file without interpreting them. The `\bfseries` command tells `LATEX` to print the official form number in boldface while the `\normalfont` command tells `LATEX` to cease printing boldface type. The forward slash, `/`, before the variable name `formdesc`, is a SAS argument moving the line pointer to the first column of the next line. Another forward slash at the end of the `put` statement creates the blank line that `LATEX` requires between labels. The SAS formats `$char22.` and `$char44.` assure left justification of the form numbers and their descriptions.

```
options ps=50 ls=125 noovp nostimer dkricond=error dkrocond=error nomautosource
mergenoby=error;
/*****
/*PROGRAM: /home/wilco003/cfd2010/labels.sas
/*FUNCTION: create input for my LaTeX program named aver5029.tex
/*INPUTS: fewer.csv
/*OUTPUTS: labels.log, aver5029.txt, labsmade1.txt
/*CALLS TO: nothing
/*SPECIAL NOTES:
/*documentation about location of the spreadsheet containing data to be printed
/*on file folder labels and steps in converting spreadsheet into a CSV file
/*and editing it
/*AUTHOR: Adeline Wilcox          DATE: 29Feb08
/*UPDATED: Adeline                DATE: 03Mar08
/*To the 4 records with an ampersand in the form description, I preceded these
/*ampersands with an the \ escape char for LaTeX.
*****/
filename sharon '~/onthesgi/fewer.csv';
filename avery '~/onthesgi/aver5029.txt';
filename mine '~/onthesgi/labsmade1.txt';
libname mine '~/onthesgi/';
data read3xls(drop=junk1 junk2); infile sharon dsd lrecl=143;
length ombnum $ 22 readform $ 120;
input junk1 $ junk2 $ ombnum readform;
run;
data shorter; set read3xls;
length formdesc $ 44;
formdesc=substr(readform,1,44);
run;
ods listing;
proc print data=shorter;
var formdesc;
run;
data _null_; set read3xls; file mine;
put ombnum $char22.;
run;
data _null_; set shorter; file avery;
put '\bfseries ' ombnum $char22. '\normalfont ' / formdesc $char44. / ;
run;
```

L^AT_EX INPUT FILE

My SAS program named `labels.sas` output the ASCII file named `aver5029.txt`. Eight records are listed below.

```

\bfseries D-1                \normalfont
Mailback-English Initial, Repl, Fulfill, U/L

\bfseries D-1(E/S)           \normalfont
Bilingual-Eng/Span (Initial)

\bfseries D-1(C)             \normalfont
Fulfill - Chinese (Simplified)

\bfseries D-1(K)             \normalfont
Fulfill - Korean

\bfseries D-1(R)             \normalfont
Fulfill - Russian

\bfseries D-1(S)             \normalfont
Fulfill - Spanish

\bfseries D-1(V)             \normalfont
Fulfill - Vietnamese

\bfseries D-1(UE)            \normalfont
Update Enumerate - English

```

L^AT_EX PROGRAM

I used the L^AT_EX package `labels.sty`, also known as a "style file", written by Rahtz et al. [Rahtz, 2003]. I have the L^AT_EX document preparation software available to me both on the server and my desktop computer. As I like to edit my `.tex` files with `gvim` on my desktop computer [Wilcox, 2002], it is convenient for me to run the MikTeX 2.4 distro of L^AT_EX there as well. The L^AT_EX labels macro, `labels.sty` is in `C:\texmf\tex\latex\labels`.

If you do not have the L^AT_EX computer typesetting software running on any computer you use, consider that L^AT_EX installation can be a daunting process requiring a team of wizards or a simple download from the MikTeX web site at <http://www.miktex.org/>. If you know a L^AT_EX user, you may be able to convince him to run your L^AT_EX program for you.

My L^AT_EX program named `aver5029.tex` is listed on the next page. I modified the L^AT_EX code listed by Rahtz et al. in Section 2.4 of their paper on printing sticky labels with L^AT_EX. To execute my L^AT_EX programs quickly, I've mapped a `gvim` key to invoke the `pdflatex` driver with the command `!c\texmf\miktex\bin\pdflatex %`. When run, `aver5029.tex` reads the ASCII file named `aver5029.txt` and writes a PDF file named `aver5029.pdf`. The PDF file named `aver5029.pdf` can be printed on Avery 5029™ labels.

```

\documentclass[letterpaper,12pt]{article}
\usepackage[newdimens]{labels}
\LabelCols=2%
\LabelRows=15%
\LeftPageMargin=2mm%
\RightPageMargin=2mm%
\TopPageMargin=9mm%
\BottomPageMargin=-9mm%
\InterLabelColumn=4mm%
\InterLabelRow=-1mm%
\LabelInfotruer
\begin{document}%
\labelfile{h:/onpeecee/aver5029.txt}
\end{document}

```

To determine the where the labels should be printed, Rahtz et al. give the formula below.

$$\text{LabelRows} \times \text{Label height} + (\text{LabelRows} - 1) \times \text{InterLabelRow} = \text{paperheight} - \text{TopPageMargin} - \text{BottomPageMargin}$$

Avery 5029 labels come on sheets of 15 rows, two labels per row, making my value of `LabelRows` equal to 15. The wrapper on a ream of printer paper in my office gave me the dimensions of a sheet as 216 mm x 279 mm, so the value of `paperheight` is 279.

I fiddled with the values `TopPageMargin`, `BottomPageMargin`, and `InterLabelRow`, printing test output on plain paper. When I got the text properly positioned, I printed the sticky labels. You might need to find a more precise solution to the formula than I did.

The command `labelfile` names the external label source file.

PRINTING THE LABELS

Figure 1 shows part of the file named `aver5029.pdf`. Figure 1 correctly shows the vertical spacing between labels and the space between the two columns of labels. However, it does not show the margins as they are printed on the sticky labels. The left margin of Avery 5029 labels is too narrow to display within the margins required for NESUG papers. When I printed the sticky labels using a `hp color LaserJet 9500hdn` printer, I placed the Avery sticky labels upside down underneath a blank piece of paper. In my shop, printing is set up to print a banner page in front of every print job.

D-1 Mailback-English Initial, Repl, Fulfill, U/L	D-1(E/S) Bilingual-Eng/Span (Initial)
D-1(C) Fulfill - Chinese (Simplified)	D-1(K) Fulfill - Korean
D-1(R) Fulfill - Russian	D-1(S) Fulfill - Spanish
D-1(V) Fulfill - Vietnamese	D-1(UE) Update Enumerate - English

Figure 1: Eight Sticky Labels from the PDF File Output by \LaTeX

DISCUSSION

When hundreds or thousands of sticky labels must be printed, storing the text to be printed on the labels in a SAS dataset, ASCII file, or database such as Microsoft Access can be the first step in printing the labels quickly. Using SAS and \LaTeX to print fewer than 200 sticky labels might have taken me longer than manually entering the label information into MS Word. But I was able to quickly make a change to all the labels when they had to be reprinted with the form description on the second line of the file folder label. And should thousands of labels ever be needed, I will be ready to print them quickly.

Rahtz et al. give more documentation, essential for printing other sorts of sticky labels.

\LaTeX users also have solutions for printing business cards [Budge, 2000] and mailing form letters [Lamport, 2006].

Not only do the paper products used in Europe have different dimensions but so do the sticky label products sold there. American users of open source software maintained in Europe can run into trouble if they fail to modify their code or applications.

CONCLUSION

A skill set that includes \LaTeX can help a SAS programmer make quick work of printing labels.

REFERENCES

Alexander. M. Budge. FLASHCARDS. 16 March 2000.

<ftp://ftp.cise.ufl.edu/pub/mirrors/tex-archive/macros/latex/contrib/flashcards/flashcards.pdf>

Dongming Jiang. *An Alternative Method of Producing Labels Without the FORMS Procedure*. SUGI Paper 149-31. SAS Institute Inc. 2006. Proceedings of the Thirty-first Annual SAS[®] Users Group International Conference. Cary, NC: SAS Institute Inc.

Kirk Paul Lafler. *Basic SAS PROCedures for Generating Quick Results*. SAS Institute Inc. 2004. Proceedings of the Twenty-Ninth Annual SAS[®] Users Group International Conference. Cary, NC: SAS Institute Inc. <http://www2.sas.com/proceedings/sugi29/256-29.pdf>

Microsoft Office Online. *Create mailing labels in Access*.

<http://office.microsoft.com/en-us/access/HA100073521033.aspx?pid=CH1006456610333>

Sebastian Rahtz, Leonor Barroca, Grant Gustafson, Julian Gilbey. *A package for making sticky labels in \LaTeX 2003/05/22*. <http://www.cs.york.ac.uk/euroedit/>.

Leslie Lamport, Frank Mittelbach, Rainer Schöpf. *Standard Letter Document Class for L^AT_EX version 2_ε*. December 27, 2006.

<http://www.tug.org/texlive/Contents/live/texmf-dist/doc/latex/base/letter.pdf>

Adeline J. Wilcox. *gvim 6.x, An Alternative To The Display Manager Editor* Poster presented at the Fifteenth Annual NorthEast SAS Users Group Conference, Buffalo, New York, 29Sep - 02Oct, 2002. <http://www.nesug.org/proceedings/nesug02/ps/ps001.pdf>

ACKNOWLEDGEMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute in the USA and other countries. ® indicates USA registration.

CONTACT INFORMATION

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

Adeline J. Wilcox
US Census Bureau
Decennial Management Division
4600 Silver Hill Road, Stop 7100
Washington, DC 20233-7100
Phone: (301) 763-9410
Email: adeline.j.wilcox@census.gov