

Saving Money On Direct Mail Campaigns With SAS/AF ®

Rod Lashley, STATKING Consulting, Inc.
Kris Kellogg, STATKING Consulting, Inc.

Businesses are constantly looking for ways to increase sales and market share. One proven advertising method when the target consumer is well defined is the direct mail campaign. By utilizing this method, a company can get their message directly to the decision-making people. This strategy works well in both the retail and service industries.

The costs of maintaining and using these lists can quickly become prohibitive for small and medium-size companies. Duplicate entries in the master list cause additional mailing costs. They may also result in lost revenue when a prospective client receives multiple mailings. Recent changes in the way the US Postal Service handles bulk mail has also added complexity to an already difficult problem.

In order to improve efficiency and reduce advertising costs, STATKING developed an internal SAS® software application to maintain their mailing list. This application controls all aspects of the direct mail campaign. The application uses SAS/AF for the graphical user interface, Base SAS for data management, SAS/GRAPH® for addressing and printing envelopes, and SAS/FSP® for maintaining the advertising correspondence.

This paper covers an application that is still in development. This paper will detail the development process of this application and describe what lies ahead.

Database Design and Maintenance

The design of the database is a critical step in the implementation of the application. In addition to containing the obvious address information about clients, the database must hold additional information about the client. The database design will also have a major impact on the design of the user input frames.

By keeping the database in a SAS data set, it is easy to design queries to be a part of the application as well as to enable ad hoc queries outside of the application. Some of the important information to keep concerning tracking prospective clients is what type of statistics / SAS applications will interest them, the last mailing they received, whether they responded, and whether or not they are former clients.

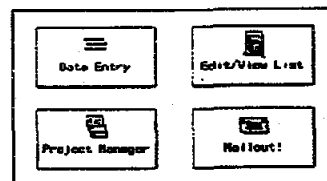
The initial cleaning of the database was done outside of the application using proc sort. All entries were then examined for close matches and duplicates were

removed. Continued cleaning of the database will be implemented by the database administrator whenever a new entry is added or modified.

Graphical User Interface

The graphical user interface (GUI) for the direct mail application consists of a series of SAS/AF Frames. The application is designed to allow for total maintenance of the list. This includes database administration, project tracking of mailings, actual creation of the letters and addressing of the envelopes, and statistical tracking of results. The initial frame contains five icons linking the actual processing frames together. These icons are Data Entry, Edit/View List, Project Manager, Mailout, and Statistics.

Mailout Automation



STATKING Consulting Inc.

Statistics

The Data Entry frame consists of the 12 fields for the entry of company name, address, employee's name and position, phone numbers, and date entered into the database.

Prior to opening the data entry frame, some background processing takes place. The type variable, which designates the general area of interest for the prospective clients is read from an ASCII flat file and stored in an SCL list. Default values are also set at this time.

To enter a new person into the database, the user enters the appropriate information into the input fields. By clicking on the 'process' button, checks on capitalization and other formatting can be run. Once the user is satisfied with the data in the input fields, they click on the 'add' button. This

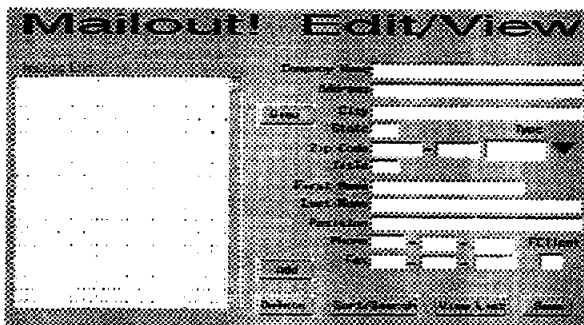
will store the information in the list box and clear the input fields. The list box contains all of the names entered in a single session. This serves as a temporary holding buffer before the master database is updated. Information stored in this list box may be viewed by different variables or deleted. To delete an entry from the list box before updating the master database, the user selects the row they want to delete, and clicks on the 'delete' button. Once the user is satisfied with the information stored in the list box, they can click on the 'save' button to append all of the information to the database.

The majority of the processing for this frame is done with SCL lists. When the add button is clicked, the data is read from the input fields and stored in SCL variables. The input fields are then cleared, and the variables are stored in a list according to the default order. This order is also stored in a list. If the user decides to view the data in a different order, the order is changed, and the ordering of the display list is modified.

Maintaining all of the data in lists also allows for easy deletion of data from the temporary buffer. When an item(s) is selected for deletion, the position in the list is returned from the list box. The list is then updated, from the bottom up to maintain the integrity of the positions returned from the list box.

When the 'save' button is clicked, the newly entered data is then checked against the existing database for duplicates or near-duplicates. If there are no duplicates or near-duplicates, the database is updated and the list box is cleared. If there are possible matches, the possible duplications are displayed, and the user is prompted to accept or reject the new data.

The 'Edit/View List' frame contains a list box that displays the entire mailing list. From the list box, you can sort the list, eliminate records from the database, or do a search on existing records in the database. A blank entry can also be added to the list that can then be modified. Any record may also be selected for deletion.



To search for an existing record, a separate frame containing text entry widgets and a list box is displayed. The user selects the search order by moving variable names from one list box to the other. The search criteria is entered in the text entry widgets.

SCL lists are again used extensively in the processing of this frame. The database entries are initially displayed in the list box. When a row of the list box is selected, the information is extracted from the list and

displayed in the text entry widgets. The data can then be edited and updated in the list by clicking on the 'save' button.

The 'Project Creation' frame allows the user to organize multiple projects at the same time. A project is defined as a specific group from the master database. Defining a project can be done either from a general subset based on a particular field in the database, or by customizing the selection process to select only a few individuals. Once a project is created, a count of the number of records satisfying the criteria will be performed, and the expected number of envelopes and stationary required to complete the mail-out will be displayed.

The frame entitled 'Mailout!' is the section of the application where the physical printing is done. A list of active projects will be given. The user selects an active project to become the current project. Once the current project has been selected, letters will be printed, and then envelopes. A status window is displayed during printing, showing the number of envelopes and letters that have been printed.

It is very important to be able to track the effectiveness of your direct mailings. This is done by the 'Statistics' frame. Selecting this icon from the main menu will enable the user to access the number of responses, the number of work orders, the amount of revenue generated, and the total cost of a project. From this, the user will be able to understand the responsiveness of a direct mail campaign. With this information, it will be easier to notice areas of strength and weakness in the advertisement campaign, and make necessary corrections.

There will be different types of letters for the different types of prospective clients. These letters are maintained in a SAS catalog, using the FSLETTER procedure. Once the project has been chosen, a letter can either be selected from the catalog or a new letter may be written. This letter is then merged with the client list. The output is sent directly to the printer. The printed letters will be ordered in postal zip code order to easily match with the envelopes that will be printed later.

Printing of the envelopes is done using the annotate facility in SAS/GRAPH®. The return address and insignia are designed off-line and stored in a SAS data set. The name and address of the recipient is read from the database according to the active project. After reading the zip+4 zip code and all other delivery point information, the postal service barcode is computed. Once the barcode is computed, proc GSLIDE is used to draw the barcodes on the envelope, as well as the recipient's address, and the return address. Printing is done in sorted order, allowing for combining envelopes and correspondence quickly. This saves time in producing the finished product as well as in bundling the pieces of mail for greatest discount.

Results and Problems

As of this writing, this application is not completely finished. Most of the individual frames are finished and work, but incorporating all of them into an application has not been accomplished. However, a mailing was completed using most of these tools. Preparation time for the lists was reduced from

almost one day to zero since the data is already in SAS data sets. The time in man-hours required to bundle the mailings to get them ready to mail after printing was reduced from over 1 week to 2 days.

The only drawback to the whole process was the time required to print the envelopes. This took slightly longer than the previous process. Using a 486-based personal computer with 100MHz clock speed and 8 MBytes of RAM, coupled with an Hewlett Packard 6P printer, it took roughly one hour to print 100 envelopes. The causes of this seemed to be with SAS/GRAPH. The graphs could not be generated quickly enough, and the printer would go into "sleep" mode between envelopes. The constant warm-up done by printer wasted a lot of time. Further analysis of the process needs to be done to identify the exact cause for the delay, and appropriate action taken. If the graphs could be ready to print about 3-5 seconds faster, the warm-up time would be eliminated, and printing would be done much faster.

In addition to any problems experienced with the process, the Postal Service's seemingly constantly changing requirements for reduction in postage caused a problem. In order to receive the largest discount from the postal service, you must, in addition to having postal barcodes on the mail, have your mailing list certified by approved software. Failure to do this every six (6) months will result in you not receiving any postage discounts. But again, by maintaining the database in a SAS dataset, it will be easy to print out the information needed to certify your mailing list, thus reducing time spent on database maintenance.

Author Contact

Rod Lashley
STATKING Consulting, Inc.
780 Nilles Road Suite E-2
Fairfield, OH 45014
513-858-2989
STATKING@compuserve.com

Kris Kellogg
STATKING Consulting, Inc.
780 Nilles Road Suite E-2
Fairfield, OH 45014
513-858-2989
STATKING@compuserve.com

SAS, SAS/GRAPH, SCL, SAS/AF, SAS/FSP are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries. ® indicates USA registration.