DEVELOPING A GENERIC APPLICATION TO PREPARE MAILING LISTS

Preston A. McCoy, AMEX Life Assurance Company*

Abstract: AMEX Life Assurance Company used direct mail to expand coverage among existing customers. Rate SAS® software is used to select desired records and perform complex processing for these mailing list pulls. The intent is always to offer a second product to existing policyholders. However, selection criteria and other requirements vary greatly for each mailing list. A standard mail pull system was desired. Therefore a flexible set of programs was created that would simplify the process, save user and programmer time, improve standardization, provide quicker turn around, and reduce errors. This paper will cover the experiences of programmers and users in developing and implementing this mail list pull system using base SAS® software.

BUSINESS BACKGROUND: Normally, the marketing staff identify a broad group of policyholders to mail to, and then define sub-groups or segments called cells. For example they may offer Accidental Death (AD) coverage to those who already have Hospital (HL) or Term Life (TL) coverage. Cells are assigned to a certain age group, State, Product, Card Type, etc. or may be assigned randomly when testing new mailing copy.

SELECTION AND SUPPRESSION: To specify the mail list, both selection and suppression rules are defined. Selections define desirable traits, such as Residence state, age, Active Status, the Financial/Insurance Product owned, etc. Suppressions define undesirable traits such as payment history, claims experience, and so on.

Both the selection and suppression criteria vary for each mailing. Some suppressions apply to only one policy record at a time (example-bad zip code), while others apply to all the records associated with one person (example-bad credit). To illustrate, consider the AD example above. If a person with HI & TL has previously filed a fraudulent hospitalization claim or if he already has AD coverage, he should not get the AD solicitation. No offer should be made. This is called a Global suppression, and the person is excluded from the mailing by deleting all other policy records that belong to that person.

BEFORE the new system was created, the process worked something like this: Specifications were prepared in narrative form usually based on a copy of a previous request. There were about 6-8 pages of narrative, and the meaning was often obscure. The programmer would read the request and clarify questions with the user. Five or six programs would be written and the jobs run. One program was needed for each of four data bases and at least one more to merge, do suppressions, assign cells, write the reports and create the mail tape and dumps. The programmer would prepare a cover memo to pull together information about selections, suppressions, final counts, dumps, displays, record layout, etc.

Users would check results. If a problem were found, a complete re-run could result. Lost time and JCL problems were common with re-runs. User confidence was shaky at best.

Associated jobs to update suppression files (claims, do-not-mail, etc.) were a separate effort and not part of the job stream.

Backleads, the method used to flag the AMEX® data base that a person had been mailed, was time consuming non-automated process. Variable names were not standardized, making some programs incompatible with each other and with existing macros and formats. The Macro and Format libraries were limited and inconsistent.

THE PEOPLE: Three programmers were involved, two journeymen and one assistant. The assistant supported the routine creation of mail tapes and the others helped out when necessary. The assistant felt frustrated and over worked and sometimes missed deadlines. He was also confused by changes the others made to the shared Format and includes libraries. Variable names were inconsistent between programs and from programmer to programmer. Problems occurred when attempting to share programs, Includes, Macros, Formats or SAS® data sets.

THE USERS: Marketing Managers and their assistants decide what mailings will be made, to whom, and when. The results of past mailings guide their decisions about future mailings. They were asked about current and future needs and also asked to identify things that bothered them about the current process. We learned that our users felt frustrated with the process and lacked confidence in the results. Furthermore, they did not really like or even understand their own specification document(s). Also, their need for post mailing analysis, (called responder analysis) to see who responded, was not being met.

SYSTEM REQUIREMENTS: Previous cross sell mailings were reviewed to learn the full range of requirements. Old user specifications were examined to see what selection and exclusion criteria had been used, and what cell assignment logic had been required. Record layouts, mail tapes and kept data sets were reviewed to see what variables and formats had been used.

THE SOLUTION combined three elements;
1. a new specification form,
2. standardized variables, formats, and macros, and
3. three new programs that employed giant macro calls.

With help from the users, a standardized request form was developed to cover all foreseeable cross sell situations. It lists selection criteria and suppression criteria. Cell assignment is still based on the mail plan. Here are portions of that form (which is actually 4 pages long):

Users would check results. If a problem were found, a complete re-run could result. Lost time and JCL problems were common with re-runs. User confidence was shaky at best.
Here is an example of the completed macro call from the first program:

MACRO CALL FROM PROGRAM 1:

```
%xselldar=_M012301
sel=AD
siaddr=Y,

********** SELECT THE FOLLOWING**********
product=G,
product=TL, AH SN,
status=S,
c_state=22,
s_state=2,
suspend=G,
suspendcode=0,
age=0,
age=041591,
age=19-59,
age=19-55,
state=S,

states=AL AK AR CO CT DE DC FL GA HI ID IL IN IA KS KY LA MA MD MS MO MT NE NV NH NJ ID OH PA RI TN UT VT WI WY,
usy,N,
Canada=N,
foreign=N,
insured=Y,
payor=N,
states2=MN CA,

********** SUPPRESS THE FOLLOWING:*******/
notame=x,S,
oclaim=x,G,
badclaim=G,
claim=03 04 06 24,
badclaim=2=S,
claimseta=32 35,
direct=S,
nodir=S,
pdoz=G,
pdozage=31,
pdozage=2=00,
issue=S,
all=LT,
issuecode=010180,
mutcol=ALL,

******** PARAMETERS UNIQUE TO AND PASSED TO MACRO XSEL2 ******
cards=GR,CO,
keepcard=CO,
oldnew=OLD,
test=N,
```

End of completed macro call from the first program:
CROSS SELL TAPE SPECIFICATIONS

Due date: __________________________ Requested By: __________________________

Description:

Products being sold:
(People owning or applying for any of these products will not be mailed)

SELECTION CRITERIA:
products:

- (AD) Accidental death
- (AH) Accidental Hospital
- (SN) Skilled Nursing
- (AF) Air Flight
- (CP) Competitive TL
- (TL) Term Life

Status Codes:

- (22) Normal Active
- (12) No Initial Premium
- (08) Inactive

Age Range(s): As of this date: __________ Optional Age Range 2: __________

State selection for Age Range 1:

- (AL) Alabama
- (AK) Alaska
- (AZ) Arizona
- (MT) Montana
- (NE) Nebraska
- (NV) Nevada
- (NH) New Hampshire

State selection for Age Range 2:

- AL
- AK
- AZ
- MT
- NE
- NV
- NH

SUPPRESSION CRITERIA NOTES:

INITIAL: Information from suppressed policy record not desired.
GLOBAL: Suppresses all policies for person if any meet the criteria.
SPECIFIC: Suppress only the policy record that meets the criteria.

SUPPRESSION CRITERIA:

Not AMEX Card member

Bad Claim per Claim Code on claim system

Typical codes (03, 04, 05, 25, 31, 33, 35)

OR (Select your own):

- (00) New
- (03) Investigation
- (09) Misc. Reopen

- (01) Pending
- (04) Foreign-Invest.
- (07) In Rehab (LTD)

- (02) Foreign-info, needed
- (05) Pending-waiver prem.
- (08) No reserve set

- (20) Continuous Payment
- (21) Paid

STANDARDS: In addition to better specifications, variable name, format libraries and macro libraries were standardized based on existing libraries. We realized that some existing programs would not work with the revised libraries, but decided it was worth it to get the desired standardization. When that was well started, the new Mail List programs could be designed.

We decided that three programs were needed. The first was a large macro with many parameters that exactly match the new request form. The new macro calls about twenty other macros, some of which call still other macros, or refer to formats in the library. Most of these macros were new and were added to the library. All followed the newly developed standards.
CELL ASSIGNMENT: While selection and suppression logic could be 'canned' as macro code, cell assignment was too varied for that approach. So, the second program allows the programmer to write whatever is required.

REPORTS AND TAPE: The third program creates the master tape/card and necessary reports and dumps. A small macro call specifies what variables go on the tape, the source of dummy records added for security, the size of the hex dump, and the report titles. Reports are produced automatically, and include a record layout and an Operational Advice to get the tape from the data center. This is a very simple operation for the programmer based on values for certain variables. The transmission of historical information to AMEX is done automatically.

SYSTEM TESTING presented a problem because the new system was different in concept as well as methodology. Thus, parallel test results were not identical. However, since the differences were explainable, and careful checking of output confirmed that the desired records were being selected, the development was judged a success.

Individual validation of the new macros was deemed critical. Two test facilities were created. The first allows individual testing of each macro. Test data and standardized results were provided. Each macro is completely documented and any macro can be tested at any time.

The second test facility is for the benefit of the marketing users. It allows runs against test data to see if macro calls are appropriate for a particular solution. The programmer can make the macro calls look like the specifications, run against a set of test data bases, then show the results to the user right away. The entire test runs quickly and provides immediate output for review.

MANAGEMENT: This project was supported by first level management, but the effort was ad-hoc and not budgeted. Team meetings lasted about 2 hours once a week and most of the development was done by one of the programmers during the week. The entire project took about eight months including the associated reforms noted above.

PROBLEMS: When the finished product was put to use we noticed that more CPU resources were needed in one lump than with the old method. During off-peak periods, about 55 minutes of CPU time and 180 minutes of elapsed time are required for the first program to run on our mainframe. On a busy day, this job may run all day, tying up an interator and interfering with other users. At night it conflicts with production and is subject to cancellation by the operator. This problem was solved by submitting the job in the late afternoon or early morning.

When a job fails, it is sometimes difficult to see why, because many other macros and tables are called by the main macro and options Source and Source2 are not normally used. This was solved by use of the test facility and turning on a macro parameter called Test which causes all of the generated macro code and error messages to be displayed.

BENEFITS: The same three programs are now used each time. Only the macro call, cell assignment logic, and JCL needs to change.

-Global suppression is more complete. For example, if a fraudulent claim exists on a policy, all records for that person will be excluded from consideration.
-Program turn-around is quicker, so jobs can be run closer to the due date, allowing the use of current data.
-Programmer productivity is improved. A cross-sell now takes two days part time instead of 5 days full time.
-JCL changes are much simpler and less error prone.
-It is easier to verify the results. Users appreciate this.
-Backed to AMEX is done automatically.
-Response time is available to check results, and any error is corrected by a simple change to a macro call.
-Programmer is more confident of supplying correct results, on time.
-Response times are no longer an intimidating prospect.
-User productivity is improved because the new specification form is simpler to complete.
-Results are standardized and have become familiar to everyone involved. Thus, the users can understand the output and are able to work easily with the programmer.
-Response analysis is better because mailing data is now stored on SAS files for easier access and because backdoors to AMEX are automatic and reliable.

CONCLUSION: A substantial effort was needed to complete this project since it included not only the development of the new application but the standardization of variables, formats, and macros. The improved programs result in more efficient use of programmer time, less rework, and more accurate file creation. Users are able to reduce costs and improve response by targeting their mailings more accurately. The standardization of variables, formats, and macros is already making subsequent development easier and more efficient.

Footnotes:
1. SAS is a registered trademark of SAS Institute Inc. Cary NC.
2. AMEX is a registered trademark of the American Express Company, New York NY.
3. AMEX Life Assurance Company is a wholly owned subsidiary of American Express Company.
4. Programs were designed and written by Steve Wilson, AMEX Life Assurance Company.

Author: Preston A. McCoy
AMEX Life Assurance Company
1650 Los Gatos Drive
San Rafael, CA, 94903-1989