COMPUTER AIDED NAME MATCHING
James Sidlo, American Medical Association

ABSTRACT
Name matching hospital names is a programming challenge since hospital names are more complex than one's own name. Hospital names tend to be written in any number of ways with name qualifiers added in or left out, word transpositions or misspellings, and arbitrary abbreviations, all the result of the writer's whim. This is apparent in the examples given below.

<table>
<thead>
<tr>
<th>Personal Name</th>
<th>Hospital Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joan Mary Ford</td>
<td>Baptist Hospital East</td>
</tr>
<tr>
<td>J. M. Ford</td>
<td>Bapt Med Center East</td>
</tr>
<tr>
<td>J. Ford</td>
<td>Baptist East Hospital</td>
</tr>
</tbody>
</table>

Matching, whether it is done manually or electronically, requires keywords (highlighted in the examples above) to search by. The more unique a keyword, the greater the possibility of finding a single match. This most basic goal of computer aided name matching has been enhanced by an integrated solution presented in this paper. This was accomplished by combining human and computer resources using the tools supplied by Base SAS® 6.06.

INTRODUCTION
The American Medical Association's (AMA) Physician Masterfile is the most comprehensive source of physician information available in the U.S. The Masterfile contains demographic, educational, and current practice information. The primary source of information is from the Physicians' Professional Activities (PPA) Census sent to the 650,000 active physicians in the U.S.. Information obtained from the PPA Census are the physician's designated specialty, type of practice, present employment, time spent in various professional activities, and hospital affiliation.

The focus of this paper will be on hospital affiliation processing, namely on our hospital look-up process, where a written hospital name is matched with an ID number on our hospital database. Let us begin by stepping back to the pre SAS® 6.06 days.

OLD PROCESS
Mainframe screen applications tend to lag behind advances like those already seen in personal computers. At the AMA, mainframe applications have screen applications that are reminiscent of the early 1980's. The lag was more evident in PPA Census processing which consisted of a collection of batch programs and a lot of paper shuffling. The major contributor to the paper shuffle was the hospital look-up aspect of PPA Census processing.

Looking at Figure 1, the old process of hospital look-up can be broken down into four steps; (1) generate a hard copy of hospital names and ID's from the hospital database; (2) look up the hospital ID of the written hospital name from the PPA Census; (3) have data entry key the census data to tape; (4) run the tape in for update to the PPA Database.

The census form asks for the physician's primary hospital affiliation, which consists of hospital name, city, and state. Through hospital look-up, this information is converted into a hospital ID number and is used to link the physicians record to our hospital database. The first step in the old process was to generate a listing of hospital ID numbers along with hospital name, city, and state. This hard copy reference was sorted by hospital name and was generated every time the hospital database was updated.

The second step was the actual process of hospital look-up. This part of the census process caused a major bottleneck from the time of receiving a census return to the time of updating census information to the PPA Database. This was due to the manually intensive process of looking up a census hospital name and searching for its ID number in the hospital ID reference. The search yielded a perfect match rate of 25%. The difficulty lay in trying to match the physician's not so perfect version of the hospital name, requiring that the processing clerk rearrange the name's keywords until a match was found. Upon a successful match, the hospital ID number was pencilled in next to the hospital name.

The last two steps completed the processing of the census. In the third step, the census forms were sent in batches to data entry for keying to tape. Lastly, in the fourth step, the tape was updated to the PPA Database.

It was apparent that the hospital matching process was ripe for total renovation, and would prove to be an excellent test bed for demonstrating the new features of Base SAS® 6.06.

NEW PROCESS
The new process of hospital look-up was one part of an overall
revamping of the processing of the PPA Census. In this revamping, the PPA Database was given online capability in the summer of 91. This was the first in-house application to make use of pull-down menus, thanks to the new features of Base SAS® 6.06. There are seven windows which can be selected using a pull down menu. The Hospital Matching window is a recent addition to the PPA Online Database.

Looking at Figure 2, the new process of hospital look-up can be broken down into three steps; (1) input census data, which has been keyed by data entry, into the name match program; (2) using the PPA Online Database application, work the hospital selection list generated by the name match program; (3) update matched hospital ID's to the PPA Database.

In the first step, a name match program is used to create a file with a list of hospitals with their accompanying 10 numbers. These are the best guesses made by the name matching program. This program is run routinely once a week, and is transparent to the user.

It is in the second step in which the innovations applied to the new process may be appreciated. In the old process, the hard copy listing of hospital ID numbers can be thought of as one huge selection list. The new process, however, generates a more practical listing of up to 10 hospitals per census hospital name. The PPA Online Database provides a hospital matching window, activated by a selection on a pull-down menu. In this window,

```
Physician: JOAN M. FORD MD
ID Hospital Name               Cty
or: 7
res: 7 PRESBYTERIAN MED CTR OF PHIL.,
papx-190 TEMPLE UNIVERSITY HOSPITAL  PHILADEL

Phonmat: -------------------------------------------------------- •.. ----
I 10 Hospital Name City

<table>
<thead>
<tr>
<th>Physician: JOAN M. FORD MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID Hospital Name</td>
</tr>
<tr>
<td>res: 7</td>
</tr>
<tr>
<td>papx-190</td>
</tr>
<tr>
<td>TEMPLE UNIVERSITY HOSPITAL</td>
</tr>
<tr>
<td>PHILADEL</td>
</tr>
<tr>
<td>4178 PRES MED CTR-PHILADEL</td>
</tr>
<tr>
<td>4170 MED COLL OF PA AFFIL-HOSP</td>
</tr>
<tr>
<td>4134 HOSP PHILA-COLL OSTEO MED</td>
</tr>
</tbody>
</table>

Figure 3 Selection List
```

shown in Figure 3, the user is presented with a list of the name match program's best guesses. A match is then selected by the user by simply keying an "X" beside one of the selection items.

Finally, the third and last step is the once a week routine of updating changes to the PPA Database.

In the old process of hospital look-up the user was concerned with making sure that the most current hard copy of the hospital ID reference was generated, handling stacks of census forms along with pencilling in ID numbers onto those forms, bringing the stacks to data entry for keying to tape, and informing the programmer to make an update to the database. The new process frees the user from those concerns with the ease of processing selection lists online with access to database information to help in making selection determinations. And for the programmer, the name match program and database updates are routinely run once a week instead of by request.

Now that hospital look-up process has been explained, the focus will be shifted to the technical aspect of computer aided name matching.

NAME MATCHING CONCEPT

```
Hospital Database           Census Hospitals
Keyword Dictionary          Word Match
Component Words
Recombine Names
Selection List

Figure 4 Name Matching Data Flow
```

Figure 4 is a conceptual picture of the data flow through SAS® data steps of the name matching process with the number of observations processed represented by the graphic boxes. The picture may be divided into four parts; (1) creation of a keyword dictionary from the hospital database; (2) breaking down of census hospital names into component words; (3) merging matches of keywords and component words; (4) recombination of component words into names generating a selection list.

This is a batch program where much of the work is done through SAS® sorting and match merging. Though the technique may not seem elegant or efficient, the object was to unburden the programmer from hard coding complicated array sorts. From this standpoint, array elements are converted into observations leaving, it up to SAS® to do all the work.

KEYWORD DICTIONARY

In this first step, a batch program generates the keyword dictionary (see Appendix 1). The dictionary is used in the third step's match scoring, which causes the most likely name matches to bubble their way to the top of the match selection list. The keyword dictionary is generated from a hospital database, upon which scoring is based on a word's uniqueness. Fewer occurrences means a greater uniqueness, and therefore, a greater score. Abbreviations and truncations are also taken into consideration. When a word is a truncated version of a longer word, the longer word's frequency is included in the truncated word, thereby reducing the truncated word's uniqueness through a lower score.

COMPONENT WORDS

Steps 2 through 4 are actually part of the name match program (see Appendix 2). The second step is similar to the first step were hospital names from the census are broken down into component words. A component word is replicated into ever decreasing length until only three characters are left. This process is a brute force way of making a match, in the next step, upon an abbreviation or misspelling. The hospital name, city and state are read in, after which the name is broken down into component words.

WORD MATCH

In the third step, a keyword for each component word is searched for. This accomplished by a binary search algorithm. Upon a match, the keyword's score is obtained, and depending whether
the match is perfect or just a close one, the score is increased or reduced respectively. This the most CPU intensive part of the name matching process.

RECOMBINANT NAME SCORING
In this fourth step, the hospital selection listing is generated. The words are then recombined into hospital names, and the word scores are tallied into name scores. Additionally, the name score is doubled upon a perfect name match, and again upon a city match. If the city doesn't match the score is halved. Additional selection list conditions are as follows:
• Generate name match selection list by descending scores, forgiving any word transpositions and name truncations.
• Combine word scores by hospital ID, thereby obtaining match scoring on the name level.
• Increase scoring upon perfect name matches. In addition, increase scoring upon city match.
• Eliminate matches having relatively poor scores.

IMPROVEMENTS ON THE HORIZON
The automated keyword selection process still needs improvements. Operators are surprised by some of the programs illogical choices for the selection list. Though Base SAS® 6.06 made the PPA Online Database possible, SAS/AF® and SAS/FSP® would have provided sophisticated data entry and menu driven tools easing program development.

CONCLUSION
The goal of this project was to combine computer and human resources into a fast and efficient matching system. Base SAS® 6.06 made available the mainframe resources and application tools providing pull down menu driven mainframe applications already seen in personal computers today. The quick prototyping through SAS® along with the its application tools made this project possible.
APPENDIX A  Keyword Dictionary Program

**APPENDIX B  Name Match Program**
DATA DETAIL;
LENGTH MENO @ N 4;
KEEPIV MENO DATE BATCH SOURCE DHOSNAME DHOSCITY DHOSST:
INFILE IN = END:
INPUT MENO @111.
@12 RECTYPE $6.
@2S BATCH $S.
@29 SOURCE $SO.
@99 DHOSNAME $40.
@99 DHOSCITY $23.
@122 DHOSST $2.
IF RECTYPE = 4:
N + 1;
IF N < &START THEN DELETE;
IF N > &START + &LIMIT - 1 THEN STOP;
IF DHOSNAME = "" & DHOSCITY = "" & DHOSST = "" THEN RETURN;
OMENO = MENO;
BINARY HOSPITAL.PPA SEARCH = OMENO KEY = MENO;
IF MENO = OMENO THEN:
MENO = OMENO;
RETURN;
END;
DAMACD = AMACD;
BINARY HOSP.HOSPITAL SEARCH = DAMACD KEY = AMACD;
IF AMACD = DAMACD THEN RETURN;
%INCLUDE 'SDA.DSOURCE.S(MENO)';
IF DHOSNAME = "" THEN DHOSNAME = 'TRANSLATE( HOSNAME, &LOWCASE, &UPCASE)';
IF DHOSCITY = "" THEN DHOSCITY = 'TRANSLATE( HOSCITY, &LOWCASE, &UPCASE)';
IF DHOSST = "" THEN DHOSST = 'TRANSLATE( HOSST, &LOWCASE, &UPCASE)';
PROC SORT;
BY N;
%MACRO DATAT';
%MACRO DATA;
%MACRO D0 FIRSTOBS = 1 %TO &LIMIT %BY &AT_ATIME;
OPTIONS FIRSTOBS = &FIRSTOBS OBS = %EVAL( &FIRSTOBS + &AT_ATIME - 1) :
DATA SUBSET:
SET DETAIL;
OPTIONS FIRSTOBS = 1 OBS = MAX:
%MATCH( DATABASE = HOSPITAL, DETAIL = SUBSET);
DATA DATA( KEEP = LIST PERFECT );
MERGE DATA SUBSET:
BY N;
LENGTH DHOSMAT $180;
RETAIN DHOSMAT;
IF FIRST.N THEN DO;
DHOSMAT = "";
LIST = 0;
END;
LIST + 1;
IF LIST <= 20 THEN DHOSMAT = TRIM( DHOSMAT ) || " " || LEFT( PUT( AMACD,8. ), );
ELSE DHOSMAT = "";
IF LAST.N;
%INCLUDE 'SDA.DSOURCE.S(MENO)';
CHECK = SUBSTR( MENO, 7, 8 );
LENGTH DATETIME $18;
ARRAY HEADER(1) $16
DATETIME CHECK BATCH SOURCE ID COMMAND SECONDS:
ARRAY HOS(1) $ DAMACD2 DHOSNAME DHOSCY DHOSST DHOSMAT
IF DATE THEN
DATETIME = PUT( DATE, DATE7. ) || "00:00:00 ";
SOURCE = COMPRESS( SUBSTR( SOURCE, 13 ) );
IF LIST = 1 & PERFECT THEN
DAMACD2 = DHOSMAT;
DO 1 = 1 TO DIM( HEADER );
IF HEADER(1) = " THEN HEADER(1) = " ;
END;
FILE OUT MOD,
PUT MENO HEADER(')
DO 1 = 1 TO DIM( HOS )
IF HOS(1) = " THEN
PUT HOS(1) = ';
END;
PUT;
PROC FREQ;
TABLES LIST PERFECT LIST * PERFECT
I LIST MISSING;
%MEND;
%MEND;
%MEND;