USING THE SAS SYSTEM TO PROCESS 1980 CENSUS SUMMARY TAPES

John Blodgett, University of Missouri-St. Louis

ABSTRACT

The series of Summary Tape Files produced by the U.S. Census Bureau represent the primary machine readable products of the 1980 decennial census. Because of their massive size and complexity these files present a challenge to any shop that needs to use the information in a timely and efficient manner. The staff of the UMSL Urban Information Center has created a SAS based software package that we feel can significantly reduce the costs and programmer effort required to use these files effectively and reliably.

INTRODUCTION

In a paper titled PROCESSING 1980 CENSUS DATA: SAS VS. CENSPAC, presented at the 1981 SUGI conference, Kenneth Hardy et. al. did a preliminary analysis of the relative merits of using SAS versus the Census Bureau's CENSPAC software for processing 1980 census summary tapes(STF's.) Based on some rather experimental runs (using test versions of both CENSPAC and the summary tapes) they concluded that CENSPAC had some advantages over SAS in terms of computer resource efficiency and in its ability to make direct use of machine readable data dictionaries. On the other hand they noted several critical advantages of using SAS, such as superior analytical capabilities, interface with SAS/GRAPH, general ease of use without having to learn a new specialized language, and not having to deal with a relatively new software package and the host of bugs and glitches often found in such packages. Their final conclusion was that they would pay the cost of reduced efficiency and use SAS for accessing the files in their shop. While we quite agreed with their general analysis and conclusions, it seemed clear to us that the limitations attributed to SAS in dealing with these files were by no means inevitable. The advantages enjoyed by the COBOL-based CENSPAC software were primarily those associated with the relative efficiencies of record I/O in COBOL vs. formatted I/O in SAS (which of course is much slower and hence costly), together with CENSPAC's ability to make direct use of its data dictionary.

The obvious solution to the expensive I/O problem was to convert the files to SAS datasets. The solution to the data dictionary problem was partly handled by converting to SAS datasets, since this would now allow access to data by name without need to worry about logical records and column positions. By converting the files to SAS datasets, we would create a database that was efficient to access and easy to utilize within a SAS DATA or PROC step. There would be no more need to be concerned about record layouts, blocking factors or--with a little bit of extra processing added to our conversion step--suppression flags.

While the conversion to SAS datasets would significantly improve ease and efficiency of general access, it would still not solve another very important problem associated with processing these files. The single most common use of the files--at least in the early months after the data was first released--would be to produce reports consisting of simple displays of the data. "Simple" conceptually perhaps, but tedious and awkward from a programming point of view. What we wanted to do was to make the programming task as simple for the programmer as it was for the user. In other words, he should be able to simply request a report consisting of nicely formatted table displays for any or all geographic areas of any type.

SCADS: AN OVERVIEW

We began the task of developing our software for processing the 1980 Census Tapes shortly after returning from that 1981 SUGI conference. Although it did not get its name (SCADS: SAS Census Access and Display System) until we decided to "go public" with the software in August, 1983, we began using the system with the release of the first STF for Missouri in the early summer of 1981. The system had three primary components:

1. A collection of macros and OS JCL setups that handled the conversion of the raw data files as released by the Census Bureau to SAS datasets/data libraries.
2. A collection of table display macros that provided us with the ability to display complete tables with full labeling and even with additional information in the form of percentages that were not present explicitly in the data.
3. A report generator system that allowed us to access the SAS dataset we had created and to display all or selected tables for specific geographic areas.

Frankly, the amount of effort that we expended to create our software could probably not be justified by a shop that was not planning to use the system on an almost daily basis over a number of years. But now that these almost 15,000 lines of SAS source code and model JCL have been created, we hope to be able to share our efforts with other agencies with similar needs to access the data.

CONVERTING STF'S TO SAS DATASETS

The first step in making these data files easily accessible via SAS is to convert the sequential files as released by the Census Bureau to SAS datasets and data libraries. This involves a good deal more than a routine set of DATA-INFILE-INPUT setups. Considerable knowledge regarding the structure and idiosyncrasies of the files is required. Naming conventions are critical and LENGTH statements make a big difference in how much tape will be used for the datasets. The sheer size of the files creates logistical problems. Each observation (containing the summary data for a single geographic area such as a state or county or census tract) on STF3 (to cite the most widely-used and moderately-sized example) consists of up to 39 geographic identifier and flag variables, 27 suppression flags, and 1026 counts (table cells) organized into 150 "tables". The sequential version of this record occupies 12,096 bytes, and after converting to a SAS observation it is still 5,984 bytes long. Consider the work space required to sort the California file which consists of over 34,000 such observations. A special "pointer sort" macro which strips off only sort keys and then reconstructs a sorted SAS dataset using "SET POINT=" statements was devised to get around this resource problem.

The conversion-to-SAS process in SCADS is actually done at two levels. At "level 1" we do a simple straightforward conversion of logical records to SAS observations. Suppression flags are examined and the corresponding table items are assigned missing values. All table items (counts) are given names of the form TxIy, where x is the table number and y the item number within the table. All other variables are stored as character strings and are given names corresponding to those used in the CENSPAC data dictionaries (and hence appearing in all documentation.) At the second level of conversion we create what we call "level 2 data libraries." In these, different geographic summary levels are stored in separate SAS datasets. County level summaries can be found in the COUNTIES dataset, while census tract summaries are in the TRACTS dataset. In addition, we overcame what we felt was a very annoying "glitch" in the way low level census areas (census tracts and below) were summarized on the STF's. Each time such an area was split by an area such as MCD or place (city), the summary for the geographic area was not found in a single summary record but instead was cut into the intersecting pieces, which were not even together on the file. For example, if tract 10 was partly in Cary and partly in Raleigh, then there would be a summary record for the Cary portion and a separate summary record for the Raleigh portion--but no record summarizing just tract 10, which is what most data users really want to see. As part of our level 2 conversion process we aggregate these pieces to create summaries for complete geographic areas.

The conversion process can be rather expensive since it involves a lot of data conversions (there are 1026*34,000 or almost 35 million EBCDIC-to-binary conversions required to convert the STF3 file for California to a level 1 SAS dataset.) Level 2 conversions get expensive because of the considerable amount of sorting of large datasets involved. In converting the Missouri STF3 file (which is about average in size) we used just over 6 minutes of CPU time on an Amdahl V7 with over 15,000 tape I/O's for the level 1 conversion. Level 2 runs used just over 3 CPU minutes, 25,000 tape I/O's and almost 30,000 disk I/O's (mostly for sorting.) However, we were ahle to run these conversions at night to take advantage of slack-time discounts and we only had to run them twice. Our total "1-time" conversion costs were less that $200 for computer resources. This amount is trivial compared to what we have saved since, by never having to do another EBCDIC-to-binary conversion or aggregation of the two or more pieces of a split census tract. Additionally, where the original sequential file occupied 98% of a 6250-bpi tape the level 2 database we now use almost exclusively occupies only 36% of a tape.

DISPLAYING THE DATA WITH TABLE DISPLAY MACROS

After several months of discussion among UIC staff members about the relative merits of various approaches to creating a table display capability for SCADS, a decision was made not to write a super sophisticated, super complex, super generalized albeit user friendly report generator. Instead we opted for a relatively low-tech straightforward approach that would require only that several student assistants with an elementary knowledge of SAS and census data be available to do most of the work. Thus was born the concept of table display macros. A somewhat abridged example appears in Figure 1. The results of executing it appear as part of Figure 2.

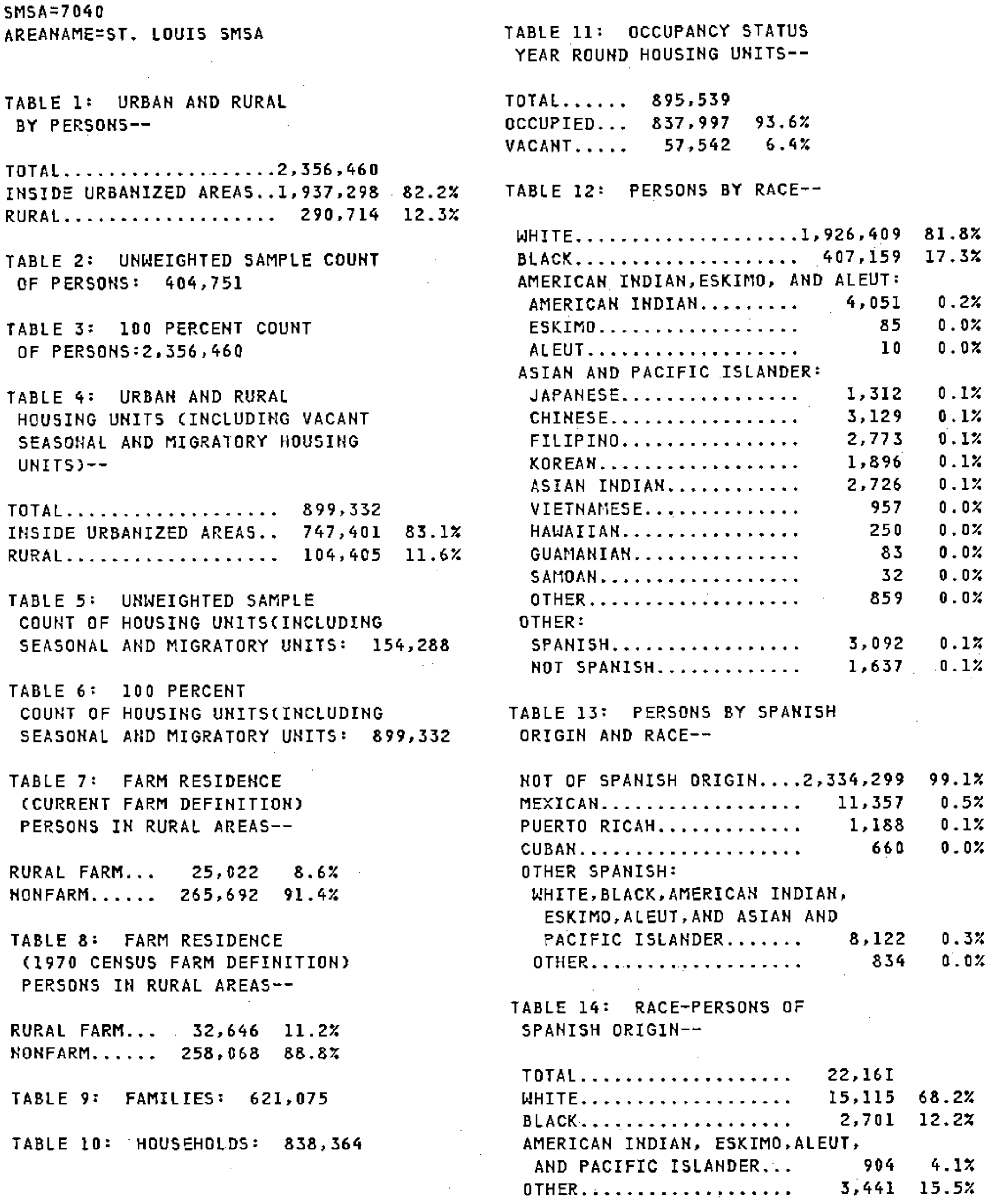
FIG 1: LISTING OF SAMPLE DISPLAY MACRO

```
MACRO _F83T12
_T_=12;
LINK SETLINE;
IF T1I1>0 THEN PCTFACTR=100/T1I1; ELSE PCTFACTR=.;
PB=1;PE=17;LINK PCT_DO;
PUT @C 'TABLE 12:  PERSONS BY RACE--'/
   @C ' '/
   @C ' WHITE.....................' T12I1 COMMA9. +1 P1 5.1 '%%'/
   @C ' BLACK.....................' T12I2 COMMA9. +1 P2 5.1 '%%'/
   @C ' AMERICAN INDIAN,ESKIMO, AND ALEUT:' /
   @C '  AMERICAN INDIAN.........' T12I3 COMMA9. +1 P3 5.1 '%%'/
   @C '  ESKIMO..................' T12I4 COMMA9. +1 P4 5.1 '%%'/
   @C '  ALEUT...................' T12I5 COMMA9. +1 P5 5.1 '%%'/
   @C ' ASIAN AND PACIFIC ISLANDER:' /
   @C '  JAPANESE................' T12I6 COMMA9. +1 P6 5.1 '%%'/
   @C '  CHINESE.................' T12I7 COMMA9. +1 P7 5.1 '%%'/
   @C '  FILIPINO................' T12I8 COMMA9. +1 P8 5.1 '%%'/
   @C '  KOREAN..................' T12I9 COMMA9. +1 P9 5.1 '%%'/
   @C '  ASIAN INDIAN............' T12I10 COMMA9. +1 P10 5.1 '%%'/
   @C '  VIETNAMESE..............' T12I11 COMMA9. +1 P11 5.1 '%%'/
   @C '  HAWAIIAN................' T12I12 COMMA9. +1 P12 5.1 '%%'/
   @C '  GUAMANIAN...............' T12I13 COMMA9. +1 P13 5.1 '%%'/
   @C '  SAMOAN..................' T12I14 COMMA9. +1 P14 5.1 '%%'/
   @C '  OTHER...................' T12I15 COMMA9. +1 P15 5.1 '%%'/
   @C ' OTHER:'/
   @C '  SPANISH.................' T12I16 COMMA9. +1 P16 5.1 '%%'/
   @C '  NOT SPANISH.............' T12I17 COMMA9. +1 P17 5.1 '%%'/;
%
```

Note the following characteristics of tahle display macros:

1. Although not trivial to code, after you have done 3 or 4 the process resembles typing as much as it does programming. Students with only a very elementary knowledge of SAS coded most of these macros.
2. Although typing errors are easy to make in these display macros, they are also

FIG. 2: SAMPLE REPORT GENERATED USING TABLE DISPLAY MACROS



```
SMSA=7040
AREANAME=ST. LOUIS SMSA


TABLE 1:  URBAN AND RURAL
 BY PERSONS--

TOTAL...................2,356,460
INSIDE URBANIZED AREAS..1,937,298  82.2%
RURAL..................  290,714  12.3%

TABLE 2:  UNWEIGHTED SAMPLE COUNT
 OF PERSONS:  404,751

TABLE 3:  100 PERCENT COUNT
 OF PERSONS:2,356,460

TABLE 4:  URBAN AND RURAL
 HOUSING UNITS (INCLUDING VACANT
 SEASONAL AND MIGRATORY HOUSING
 UNITS)--

TOTAL..................  899,332
INSIDE URBANIZED AREAS..  747,401  83.1%
RURAL..................  104,405  11.6%

TABLE 5:  UNWEIGHTED SAMPLE
 COUNT OF HOUSING UNITS(INCLUDING
 SEASONAL AND MIGRATORY UNITS:  154,288

TABLE 6:  100 PERCENT
 COUNT OF HOUSING UNITS(INCLUDING
 SEASONAL AND MIGRATORY UNITS:  899,332

TABLE 7:  FARM RESIDENCE
 (CURRENT FARM DEFINITION)
 PERSONS IN RURAL AREAS--

RURAL FARM...   25,022   8.6%
NONFARM......  265,692  91.4%

TABLE 8:  FARM RESIDENCE
 (1970 CENSUS FARM DEFINITION)
 PERSONS IN RURAL AREAS--

RURAL FARM...   32,646  11.2%
NONFARM......  258,068  88.8%

TABLE 9:  FAMILIES:  621,075

TABLE 10:  HOUSEHOLDS:  838,364
```

```
TABLE 11:  OCCUPANCY STATUS
 YEAR ROUND HOUSING UNITS--

TOTAL......  895,539
OCCUPIED...  837,997  93.6%
VACANT.....   57,542   6.4%

TABLE 12:  PERSONS BY RACE--

 WHITE....................1,926,409  81.8%
 BLACK....................  407,159  17.3%
 AMERICAN INDIAN,ESKIMO, AND ALEUT:
  AMERICAN INDIAN.........     4,051   0.2%
  ESKIMO..................        85   0.0%
  ALEUT...................        10   0.0%
 ASIAN AND PACIFIC ISLANDER:
  JAPANESE................     1,312   0.1%
  CHINESE.................     3,129   0.1%
  FILIPINO................     2,773   0.1%
  KOREAN..................     1,896   0.1%
  ASIAN INDIAN............     2,726   0.1%
  VIETNAMESE..............       957   0.0%
  HAWAIIAN................       250   0.0%
  GUAMANIAN...............        83   0.0%
  SAMOAN..................        32   0.0%
  OTHER...................       859   0.0%
 OTHER:
  SPANISH.................     3,092   0.1%
  NOT SPANISH.............     1,637   0.1%

TABLE 13:  PERSONS BY SPANISH
 ORIGIN AND RACE--

 NOT OF SPANISH ORIGIN....2,334,299  99.1%
 MEXICAN..................    11,357   0.5%
 PUERTO RICAN.............     1,188   0.1%
 CUBAN....................       660   0.0%
 OTHER SPANISH:
  WHITE,BLACK,AMERICAN INDIAN,
   ESKIMO,ALEUT,AND ASIAN AND
   PACIFIC ISLANDER.......     8,122   0.3%
  OTHER...................       834   0.0%

TABLE 14:  RACE-PERSONS OF
 SPANISH ORIGIN--

 TOTAL...................    22,161
 WHITE...................    15,115  68.2%
 BLACK...................     2,701  12.2%
 AMERICAN INDIAN, ESKIMO,ALEUT,
  AND PACIFIC ISLANDER...       904   4.1%
 OTHER...................     3,441  15.5%
```

generally very easy to find and correct.

3. Clearly these macros are assumed to be part of a larger program. There are common routines (SETLINE and PCT_DO) and an array of table sizes with index variable _T_ which are not seen here.
4. These macros display multi-dimensional matrices of data with the required labeling to make the information intelligible. The calculated percentage variables are a significant enhancement for most users.
5. The macros are named using a convention that allows them to be invoked by a report generator macro.
6. No printer line uses more than 43 characters. This permits a multi-column report of up to 3 columns on a 132 character print line.
7. These macros can be used in "black box" fashion by a user with little or no knowledge of SAS.
8. The display macros for the 150 STF3 tables are stored together as a single member of a macro library. This member is 4,466 lines long.
9. A program that invoked all 150 of the STF3 macros for all 114 census tracts in the city of St. Louis generated a report that was 2,166 pages long in triple-column format. This program required less than .5 minutes of CPU time to compile and execute.

The display macros are the heart of our display system. We also have two table-display-report generators. The first one uses only old-style macros (all that were availahle when the system was being initially developed) and involves entering specs via FSEDIT which are then processed by a SAS data step to generate a joh stream. The second system (which has pretty much replaced the first in our shop) uses a preprocessor macro called STFRGENB which can directly generate a report and does not require use of FSP or even an interactive system.

CREATING MORE USEFUL SUMMARIES USING "LEVEL 3" MACROS

When the census tapes were first released there was a tremendous initial demand for complete table displays as users wanted to look at everything to see what might be of interest. However, as time has passed we have seen users become more selective about what information they want to see. A 2,166 page report might be useful for reference but when all you want to do is get a basic feel for a few crucial demographic characteristics of a series of geographic areas (typically census tract or ZIP codes among our users) most people would prefer something considerably more compact. Our response to this kind of request has been to reduce the census data to still another "lower" level. Whereas level 2 datasets manipulated the census data vertically by combining, sorting and segregating summary observations, the level 3 conversion process manipulates the data horizontally by eliminating and combining the variables within an ohservation. A typical level 3 dataset observation will have a total length of no more than a few hundred bytes as compared to the several thousand byte length of a level 2 observation. The typical level 3 observation contains numeric summary variables with names such as TOTPOP, MEDHHINC, MED_AGE or PCTBLACK. The table variables from the original level 2 dataset are replaced with these summary figures which may be simple renames (as in the case of TOTPOP) or simple percentage figures (PCTBLACK) or values derived by applying estimation algorithms to distribution arrays (MED_AGE.) Care is taken to see that these variables are well labeled, with PROC PRINT SPLIT=/ in mind. To assist us in the creation of such datasets we have once again resorted to a collection of macros. These macros (written in the new preprocessor macro language) are intended to serve as independent modules that do all the work required to define, label and KEEP a specific variable. By convention the macro name corresponds to the name of the variable that it creates. Some examples of level 3 macros are shown in Figure 3.

FIG.3 SAMPLE LEVEL 3 MACROS

```
%MACRO POORFOLK;
  LABEL POORFOLK='PERSONS/BELOW POV/LEVEL';
  FORMAT POORFOLK COMMA7.;
  POORFOLK =
  T9112;
  KEEP POORFOLK;
  /*TOTAL PERSONS BELOW POVERTY LEVEL*/
  /*NO AGGREGATION UNIVERSE*/
%MEND POORFOLK;

%MACRO POVUNIV;
  LABEL POVUNIV  = 'PERSONS: KNOWN/POVERTY LEVEL';
  FORMAT POVUNIV  COMMA7.;
  POVUNIV  =
  SUM (OF T9111 T112);
  KEEP POVUNIV;
  /*PERSONS FOR WHOM POVERTY STATUS WAS DETERMINED*/
  /*NO AGGREGATION UNIVERSE*/
%MEND POVUNIV;

%MACRO PCTPOOR;
  LABEL PCTPOOR='PCT OF/POP BELOW/POV.LEVEL';
  FORMAT PCTPOOR  5.2;
  IF SUM (OF T9111 T112) > 0 THEN PCTPOOR = T9112/SUM (OF T9111 T112);
  KEEP PCTPOOR;
  /*PCT. PERSONS BELOW POVERTY LEVEL*/
  /*AGGREGATION UNIVERSE: POVUNIV*/
%MEND PCTPOOR;
```

The macros in Figure 3 are related but independent. They can be used to create variables counting the number of persons below the poverty level in an area, the percentage of persons living in poverty and the number of persons for whom poverty status was determined (which is not the same as total population generally.) Note the uniform style of coding. (We have since written a SAS program that allows us to enter the essential information for a level 3 macro and handles the task of generating the structured SAS code.) Also notice the concern with "aggregation universe". This is because of the very common desire of users to aggregate level 3 datasets with variables such as PCTPOOR of MED_AGE. To make the results meaningful these variables need to be weighted by a universe variable, aggregated, and then "unweighted" to yield a weighted average in the final aggregated dataset. It often requires very careful reading of the documentation to determine precisely what the universe variable should be.

By making variables such as estimated medians and percentages be defined by these macros we can insure a certain uniformity of definition within our level 3 datasets while allowing total flexibility as to what variables are to be included in any level 3 dataset. While we have only one level 2 dataset for counties in Missouri, we allow for as many level 3 datasets for Missouri counties as various applications require. Typically they are not even permanently kept. Frequently they are passed directly to a SAS/GRAPH PROC GMAP step to display them or to a SAS procedure for statistical summarization. And not infrequently level 3 datasets are intended to be used as the source of data to be downloaded from the mainframe for subsequent processing on a microcomputer. Clearly their more reasonable size makes them much more suitable for this kind of application.

MAPPING THE DATA USING GMAP

One of the most significant advantages of using SAS to process this data is the ease of interface with the SAS/GRAPH procedures, especially PROC GMAP. The interest in this type of display of census data is apparent in the recent proliferation of mapping software, especially for microcomputers. In addition to the cartographic data bases (i.e. PROC GMAP MAP datasets and their equivalents) supplied by SAS Institute for state and county level geography, there are now a number of companies developing such mapping files for smaller areas such as census tracts and ZIP codes. One such company, Geographic Data Technology of Lyme, New Hampshire, offers such datasets for all major SMSA's in the country. The UIC has obtained a number of these files and has developed a macro to easily convert them to SAS GMAP datasets. We are now working on refining our MAPGEN macro to make well-labeled choropleth map generation even simpler and more reliable.

CONCLUSIONS

To say that we have been generally happy with SCADS as a tool for processing 1980 census data would be a considerable understatement. Having spent a decade processing 1970 summary tapes using PL1 and COBOL we are in a good position to appreciate the difference that SCADS has made. Our turnaround time on most requests has been reduced dramatically. The reliability of the results we produce has also shown a dramatic improvement. Student assistants are able to master the system with relative ease and can handle most routine requests. We have several faculty members and local government people accessing the data themselves in spite of very limited experience with SAS or computers in general. And we even have one group of off campus users with virtually no mainframe computer experience and no knowledge of SAS who are able to generate specific demographic profile reports for any aggregation of census tracts, counties, or MCD's in the entire country. They do this with a custom CLIST under TSO which invokes SAS and loads a very user-friendly set of macros which allow them to specify their requests.

In summary, SCADS as it now exists provides an efficient method for accessing a specific collection of public data files. For users needing to access the data but with little or no experience using SAS or computers in general, the SAS language (especially with the new preprocessor) allows the creation of very customized and easy to use systems. SCADS also serves as a basis for creating much more sophisticated application macros for the more serious census data user.

On a more general level the SCADS system is an example of using SAS as a meta-language to build powerful, flexible, easy to use software tools for solving problems which might at first seem inappropriate for a SAS solution. SCADS illustrates how very large files with complex structures can be made accessible to SAS programmers and end users with little or no sacrifice in efficiency or flexibility. The keys to building such systems are the careful restructuring of the data into SAS datasets and databases, and the resourceful application of macros. While building such SAS-based systems is by no means a trivial or easy task, the long term savings in both computer and manpower resources can more than justify the effort.

REFERENCES

Hardy, Kenneth a, Franck C. DiIorio and Judith M. Poole (1981). "Processing 1980 Census Data: SAS Versus CENSPAC." Proceedings of the Sixth Annual SAS User's Group International Conference.

John Blodgett
University of Missouri-St. Louis
Office of Computing
and Telecommunications
8001 Natural Bridge Road
St. Louis, MO 63121