

Providing Relatively Simple Access To Complex Public Data Files

John Blodgett, Urban Information Center
University of Missouri—St. Louis

OVERVIEW

The Missouri State Census Data Center (MSCDC) has as its primary mission providing easy and efficient access to a large collection of public data files. The MSCDC uses the SAS (R)* software package almost exclusively to achieve this goal. In this paper we shall illustrate the manner in which we do this by focusing on a single type of file—the Census Bureau's Summary Tape File 1A (STF1A)—and describing how we provide access to this complex and widely used file.

The system we'll be describing runs primarily under SAS 6.06 on a large IBM mainframe computer running MVS/TSO. However, many of the ultimate "consumers" of the data products produced will be doing their work using VM/CMS, PC-DOS and other platforms. Many—but not all—will be using SAS for their final analyses.

COPYING AND RESTRUCTURING

Normally the "copy phase" of processing a new file is relatively trivial. The primary purpose is to replicate the data file as received from the data producer. The only difficult part of a copy run is the assignment of the output data set name (DSN) under which the file will be catalogued. In the case of STF1A, however, we made a decision to split the file in two: one file containing complete geographic area summaries (the "inventory" file) and the other summaries for smaller areas split by township and place (the "hierarchical" file). The "H" records comprise 2/3 of the file but in our experience account for less than 1/4 of the usage, so segregating them would result in considerable savings in the time and I/O required to satisfy most inquiries.

The job that accomplishes the copy-and-split processing is stored in a special job library (COPYJOBS.CNTL) and the results of the run (JCL log, SAS log) are captured in a public library

(COPYLOGS) which can be browsed by anyone needing to see details of the run. Hard copy of the run is kept in the traditional file folder. The job name (STF91AMO) and the member names for the libraries are all identical and will also be used for the job name and member names used for the conversion phase.

We alluded above to the "difficulty" of assigning a DSN to the output file when doing a copy. Naming data sets is important in helping us easily identify and locate files. We take advantage of the multiple levels of DSN's in MVS, using each level of the name to provide information about the data set. The DSN of the sequential, inventory-records file for the state of Missouri is \$6386.PUBLIC.STF901A.MOI.L0. The first level, \$6386, identifies the account that "owns" the data set; the PUBLIC level is used in conjunction with RACF conventions on our system to identify files that can be read (but not written) by anybody who can access the system. The third level is the "file type" and identifies what kind of data file it is. The first two levels are virtual constants for the MSCDC public archive but this third level is where it gets interesting. The TSO LISTCAT command provides information comparable to the DIR command in PC-DOS. Entering `listcat l($6386.PUBLIC.STF901A)` would produce a listing of all catalogued files with these first three levels—which means all 1990 STF1A files. In a directory-based system like DOS this would be equivalent to using subdirectories and the command `dir \ $6386\PUBLIC\STF901A`.

The fourth level of the DSN, "MOI" in the example, identifies a specific occurrence of the filetype. In this case it indicates the Missouri inventory file ("ILH" was used for the Illinois hierarchical file). The last level of the DSN, L0, is called the "mode". By convention L0 ("level zero") means a sequential file. If we create a single-data set version of this raw file (which we will) the DSN of the SAS data set will be identical to the sequential file except that the mode will be "SSD". The internal name of

the SAS data set will--again by a rigorously enforced convention--be MOI, the same as the DSN level just preceding .SSD. If we convert a file into a SAS data library with more than one data set then we use SDL as the last level of the DSN.

An automatically scheduled weekly job is used to produce a sorted, annotated listing of all files catalogued under \$6386.PUBLIC. This report is stored in file \$6386.PUBLIC.DATASETS.CATALOG.

CONVERTING TO SAS

A fundamental principle of the MSCDC system is that any file that we plan to use more than three times should be converted and stored in the form of one or more SAS data sets. Writing INPUT statements to read selected data from a sequential file is error-prone, time consuming, inefficient (INPUT statements are much slower and resource intensive than SET statements), and make developing generalized applications much more difficult. A carefully coded conversion will also tend to be much more efficient in terms of the space used to store the resulting data sets, and will avoid doing unnecessary numeric conversions.

We frequently will use the conversion step to enhance the structure and/or content of a file. For example, a survey file such as PUMS or the Current Population Survey which has multiple record types will be converted into a series of easily linked SAS data sets. In the case of STF1A, however, the conversion process is rather straightforward. The SAS INPUT statement for reading all 1000+ variables from an STF1 record is only about sixty lines of code and the entire conversion step without LABEL/ATTRIB statements is only about 100 lines. Assigning useful labels requires many lines, of course, and presents a special challenge because of the 40-character limit. What label do you use for a variable such as P25I3--the number of non-family households with two or more persons with one or more persons aged 65 or older? (We used "NON-FAMILY HOUSEHOLDS", the leading colons signaling that you need to see labels of preceding variables for qualifiers).

The Census Bureau provides a machine readable data dictionary with its 1990 data files and we did

use it to generate "skeleton" labels for the data cells, but it was only marginally helpful. The Bureau's data dictionaries make no allowance for the SAS 40-character label limit.

The conversion code is stored in the form of a SAS macro in a public macro library. This is particularly convenient since we shall be converting additional STF1 files for other states. The macro supports a "user exit macro" to allow for customizing the conversion process without having to modify the conversion macro. We used this exit macro to assign Missouri Regional Planning Commission codes to all records with a county code.

CREATING A CODEBOOK

Because of the complexity of files such as STF1 and the limitations of the SAS system for providing adequate meta-data on a SAS data set, the MSCDC creates documentation modules called "codebooks" or "CDBKS" (for the DSN of the library where we store them). These are structured documents with (as yet untapped) potential for serving as code-generating modules. Their use to date has been entirely as eye-readable data dictionaries. We have experimented with generating CDBKS modules directly from Census Bureau data dictionaries, but with limited success. Once again, the label length incompatibilities have proven to be the biggest obstacle. We have found it to be easier to first create the SAS data set(s) and then use a SAS macro to generate the skeleton of the CDBKS module. This was how we generated our initial STF901 codebook.

An important structural note is that "STF901" is what we call a "file type". There is only one codebook module associated with each file type, while there can be many SAS data sets.

A detailed description of the structure and potential applications of CDBKS modules is beyond the scope of this paper but can be supplied upon request.

THE HELP MODULE

While the CDBKS library provides a very detailed down-to-the-variable type of documentation for a

file type, we find that it is useful to also provide a HELP library with briefer descriptions of each file type. The HELP module for STF901 is fewer than one hundred lines of descriptive text. Users can browse these modules to help determine if the files they describe may be relevant to their data needs. If they need more information they can then refer to the CDBKS library.

APPLICATIONS

The system applications library (APPLCTNS) consists of setups that utilize the files in the public archive to produce specific reports or other data products. They frequently reflect applying software tools to perform common manipulations and/or displays of the data.

For STF901 we created three sample applications (so far).

The two most important application modules related to the STF901 data both involve displaying the data in special reports. One of these involves the use of our %TABRGEN macro/system which will be described in Jim Struthers' paper. The other involves using the STF1 profiles software which we helped to develop for the national State Data Center organization. This is a fairly simple program to run but the source code for it is over 2500 lines of code. It typically produces a series of 8-page "profile reports" for each STF1 observation on the input SAS data set.

Producing these profiles is greatly simplified and made more efficient by being able to reference a converted SAS data set for the input and by being able to reference a stored DATA step program instead of having to recompile the huge source modules. That is, it is "simplified" if you have a model setup that points you to the appropriate SAS catalog where the stored program resides and shows you the relatively simple syntax necessary to invoke it. Complete job setups to generate these profile reports usually run about 30-40 lines of code, including JCL.

UPDATING METADATA FILES

In a data archive metadata files--files that describe and help you find your data--can be as useful (and as time consuming to create and maintain) as the actual data files. We spend a lot of time

creating log and listings libraries, our on-line DATASETS.CATALOG file, etc. We also periodically publish a *MSCDC Data Sets Reference Manual* which provides users (even those without access to or interest in directly accessing our computer system) with an overview of our data holdings. Although we do not use SAS to produce this publication, we do use two very important metadata files stored as SAS data sets to maintain the information that goes into the *Reference Manual*.

The "Data sets data set" (also referred to as DSDS or DS2) is a SAS data set containing one observation for each "data" data set in the public archive. In the case of SAS data libraries there can be an entry (observation) for each SAS file (data set or view) on the SDL. In addition to the obvious variables such as DSN, FILETYPE, VOLSER, SASDSN and ENGINE, we incorporate variables like UNIVERSE (geographic or other applicable universe for the file) and OBSUNITS (geographic or other unit described by the records/observations). A series of eight 64-byte text variables are used for optional comments or clarifications.

Maintenance on this data set is a 3-phase process that begins with a batch job to create skeleton transaction observations (by identifying new catalog entries since the program was last run), and ends with a job to apply and verify the transactions. The middle phase is where the work of entering in the non-automatable variables has to be done using FSEDIT.

A second metadata file is used to help us keep track of our "reference modules". These are the members of our reference libraries--CNVTLOGS, LISTINGS, CDBKS, APPLCTNS, etc. The update process is similar to that used for the DSDS.

Reports generated by SAS from these two metadata files are downloaded and directly incorporated into the *MSCDS Data Sets Reference Manual*. Our goal is to update this manual quarterly.

CONCLUSIONS

Files like STF901 are a very rich source of information that can be utilized by academic researchers, planners, government agencies and

the private sector. By creating versions of these files in the form of carefully converted, well documented SAS data sets and by providing additional software tools for transforming and manipulating these data, we significantly enhance the ability of researchers to access them and turn them into useful information. It should be apparent from the brief descriptions provided here that there is a considerable amount of effort that goes into creating this infrastructure. There is also a certain degree of risk involved whenever you attempt to enhance data by restructuring it into more useful forms. Errors can and do occur. But we think that such errors are less likely to occur, and are much more likely to be found and connected, if the data transformation takes place centrally and the resulting files and software tools are used by a large number of data analysts.

The goal of this system is to provide a more reliable, user friendly, efficient, and predictable environment for researchers needing to access public data files. By "predictable" we mean that the researcher thinking of accessing a certain data file will be in a much better position to predict the cost of doing so—in terms of dollars, man-hours, computer resources, etc. Perhaps even more important, the researcher needs to predict and quickly evaluate the appropriateness of the file for the problem. One of the enormous side benefits of going to all the trouble of converting, documenting and doing applications prior to having an end user specifically request it, is that you wind up with people who understand something about the files. This expertise can be invaluable when advising potential users regarding the appropriateness of using a file for a particular application. Also, by creating an on-line "sampling environment" where metadata and sample files (SAS data sets) are made readily available for researchers to browse, it presents the possibility not only of determining the nature of a *given* file, but also of finding a file that may be relevant for a *given* data requirement.

What remains to be done with this system, beyond the relatively straightforward matter of keeping up with new files and filetypes, is to build more end user interfaces that will make it much easier for the not-a-sophisticated-SAS-user to find their way around. In the world of SAS 6.06 and beyond, this means more sophisticated menu-driven full screen applications which assist the user in gathering relevant information about the

data base as well as actually retrieving and processing the appropriate data. What the SAS macro language has done for this kind of system in the eighties, the new Screen Control Language needs to do in the nineties.

BITNET FOOTNOTE

If you would like to learn more about the use of SAS for accessing public data files and you have access to BITNET (directly or indirectly through Internet) then you might want to join the SAS Public Access Consortium. SASPAC-L is a list server that facilitates the exchange of ideas, opinions, news, questions and answers related to SAS and public data files. A related file server provides easy access to collections of SAS code modules such as some of those described in this paper. Both the list and file servers are maintained by the author on a VM system at UM-St. Louis. To subscribe to the list you need to send an electronic mail message to `LISTSERV @ UMSLVMA` containing the following request: `SUBSCRIBE SASPAC-L <your name goes here>`. To find out about files available from the SASPAC-L file server you need to send an e-mail message to the same address containing the request

`GET SASPAC-L FILELIST.`

This will result in a file being sent to you which contains the names and brief descriptions of the files available. Then, just as you obtained a copy of the SASPAC-L FILELIST file by sending the above GET command, you can send a request like

`GET STF901C SASMACRO`

to get a SAS macro for converting 1990 STF1A files.

You can also send e-mail to the author (`C1921 @ UMSLVMA`) if you need further assistance.

*SAS is a registered trademark of SAS Institute, Inc., Cary, NC, USA.

John Blodgett
Office of Computing
U. of Missouri St. Louis
8001 Natural Bridge Rd.
St. Louis, MO 63121-4499