1980 Census of Population and Housing Census Software Package (CENSPAC) User Manual

D1-D80-CENS-14-TECHu

Note: This technical documentation was issued at same time as the corresponding data file and does not necessarily reflect changes to the data file that may have been made later. We offer it as a service to our customers who have purchased the file, but we do not in general offer technical support for past data files. Note that although the documentation may refer to data on tape, the U.S. Census Bureau no longer releases data on magnetic tape; customers purchasing files previously released on tape will receive the file in ASCII on CD-ROM. For information about this and other technical documentation, please contact the Customer Services Center at 301-763-INFO (4636).

Additionally, Census Bureau organization, telephone numbers, and staff cited in this technical documentation are not current. For updated information, see our Telephone Contacts site at <http://www.census.gov/contacts/www/contacts.html> or Staff Search at <http://www.census.gov/cgi-bin/main/email.cgi>.

U.S. Census Bureau
Marketing Services
Office

USCENSUSBUREAU
Helping You Make Informed Decisions • 1902-2002

U.S. Department of
Commerce
Economic and Statistics
Administration
U.S. Census Bureau

CENSUS SOFTWARE PACKAGE

(CENSPAC)

USER MANUAL

Data User Services Division
Systems and Programming Branch
Bureau of the Census
Washington, D.C. 20233
March 1982

U.S. DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary
Joseph R. Wright, Jr., Deputy Secretary
Robert G. Dederick, Assistant Secretary for Economic Affairs

BUREAU OF THE CENSUS
Bruce Chapman, Director
C. Louis Kincannon, Deputy Director

DATA USER SERVICES DIVISION
Michael G. Garland, Chief

ACKNOWLEDGMENTS

The Census Software Package (CENSPAC) was designed, written, and documented under the direction of Lawrence Finnegan, Chief Systems and Programming Branch. Eddie Webber and Lawrence Iskow assisted in the design of CENSPAC. The system uses a generative program concept adapted from the CO-CENTS programs developed by the International Statistical Programs Center of the Census Bureau. CENSPAC was programmed by Eddie Webber, Michael Langan, Robert White, Bruce Walker, and Lawrence Iskow.

CENSPAC USER MANUAL

Table Of Contents

Page

CHAPTER 1

Introduction to CENSPAC

CENSPAC is a computer software package written by the staff of the Census Bureau to access data from census and other summary tape files. CENSPAC's primary purpose is to retrieve data, produce printed reports or generate new tape files from the 1980 Census Summary Tape Files (STFs). However, the package is designed to work not only with Decennial Census summary tape files, but also with any sequential data file, including other files produced by the Census Bureau and other organizations.

What CENSPAC Can Do

CENSPAC is designed primarily for data retrieval and report generation. It has the following capabilities:

-- Prints reports, with descriptive titles.

-- Calculates new variables through addition, subtraction, multiplication and division. It can be used to calculate percentages and medians.

-- Selects specific records for printing further processing, i.e. cities or counties with certain characteristics, such as high or low income, can be selected for printing or processing.

-- Extracts specific variables from records for printing or further processing. For example, the population over age 65 can be extracted from county records to produce data showing the distribution by county of a state's elderly population.

-- Sorts and ranks records by the value of a variable.

-- Aggregates variables for smaller geographic areas into larger areas, i.e. several census tracts can be aggregated into a larger neighborhood.

-- Merges two files, allowing data from different files, such as 1980 census data and 1970 census data, to be combined in a single report or tape file.

-- Creates new tape files from existing files.

The user determines the format of printed reports. The design of a report is specified by formatting the locations of data items on each page. Users can decide, for example, whether certain variables should be listed in rows or in columns. The content and location of headings and labels in a report are also determined by the user in a CENSPAC program.

'Because CENSPAC is primarily a report generation system, it has limited capabilities for statistical analysis. A user can specify computations to be performed but there are no statistical routines. CENSPAC can be used to extract data from files for input to other statistical software packages.

How To Make CENSPAC Work

CENSPAC is not a complex program to use: It works with only sixteen different commands. Once the user has a basic understanding of these commands, census data or other data files will be readily accessible.

CENSPAC commands are of three types: 1) the input commands identify the files and data dictionaries to be used; 2) the output commands identify the data items the user wants to put in a printed report or on a new data file; 3) the processing commands identify what is to be done with the data before they are printed or put onto a new file. With the processing commands, the data can be manipulated -- for example, means, subtotals, and percentages can be calculated.

The CENSPAC input commands are:

FILEIN -- identifies the input file and data dictionary to be used.

ITEM -- when a data dictionary is unavailable or incomplete, this optional input command provides definitions of the items the user wants to retrieve from a file.

MATCH -- this command provides information when merging two files.

ENDM -- required to end the input commands in a match.

The CENSPAC output commands are:

REPORT -- creates a printed report.

HEAD -- identifies the content of headings in a printed report.

ITEM -- identifies the data items to be included in a report or new output file.

FILEOUT -- creates a new data file.

The CENSPAC processing commands are:

ARRAY -- establishes a temporary working array.

COMPUTE -- assigns value of a mathematical expression using addition, subtraction, multiplication, and/or division. Can also be used to work with non-numeric literals and non-numeric variables.

EDIT -- checks a CENSPAC program for errors before processing.

HELP -- provides users with information about the CENSPAC commands.

IF -- allows for the conditional use of other CENSPAC commands.

LABEL -- allows labels to be provided for newly derived data items.

SORT -- reorders the records on a file.

SOURCE -- allows users to supply their own COBOL instructions to extend CENSPAC's capabilities.

TOTAL -- aggregates data for small geographic areas to produce information for larger geographic areas.

This is only a brief overview of the CENSPAC commands. Chapter II of this manual gives detailed instructions on the rules of the CENSPAC system and the use of CENSPAC commands.

CENSPAC Logic

CENSPAC command sets are loosely structured. There is no set order to a CENSPAC run, except that the FILEIN command must be first, and each FILEOUT or REPORT command must be followed by an ITEM command. This means that as few as three commands, one input and two output, will retrieve census data from the summary tape files:

FILEIN
REPORT and/or FILEOUT
ITEM

A FILEOUT or REPORT command, identifying the form of output, is required in every CENSPAC run. If a new data file is to be created, the user must specify FILEOUT. If the user wants a printed report, the REPORT command must appear. FILEOUT and REPORT commands may both appear in a run if a user wants a new file and a printed report.

The ITEM command, which identifies the data to be retrieved for output, is the third required command in all CENSPAC runs. The output ITEM command must follow a REPORT and FILEOUT commands.

CENSPAC processing commands are required only when data must be manipulated before being placed in a new file or report. CENSPAC processes a command set command by command, starting with the first command and processing each command thru to the end of the command set. Each record from the input file is processed through the entire command set before the next record is retrieved. This means that the sequence of CENSPAC commands will determine how the input data will be processed and whether they will be included in a printed report or on a new file. For example, certain data items may be printed in a report, while other items are entered on a new data file.

The following sample run illustrates a very basic report generated by CENSPAC using the 1980 Census STF1A data file. Population and housing counts for counties are listed in descending sequence by population size. County level summary records are selected (SUMRYLVL = '11') and the data are sorted by population size before printing the report.

The CENSPAC control cards needed to prepare the report are as follows:

```
FILEIN STF1
IF SUMRYLVL = '11' INCLUDE
COMPUTE TOTALPOP = TAB1 (1)
SORT TOTALPOP D
REPORT
HEAD '              County Name
     'Population Housing Units'
ITEM A30.0 AREANAME TABI (1) TAB4 (1)
```

The report generated by CENSPAC would be as follows:

```
              County Name                    Population Housing Units
ADA COUNTY                                     173036     67835
CANYON COUNTY                                   83756     30616
BONNEVILLE COUNTY                               65980     23492
BANNOCK COUNTY                                  65421     24819
KOOTENAI COUNTY                                 59770     26959
TWIN FALLS COUNTY                               52927     20528
BINGHAM COUNTY                                  36489     12084
NEZ PERCE COUNTY                                33220     13507
LATAH COUNTY                                    28749     11013
BONNER COUNTY                                   24163     13055
ELMORE COUNTY                                   21565      8055
MINIDOKA COUNTY                                 19718      6876
MADISON COUNTY                                  19480      5537
CASSIA COUNTY                                   19427      7010
SHOSHONE COUNTY                                 19226      7673
PAYETTE COUNTY                                  15722      6114
JEFFERSON COUNTY                                15304      4994
JEROME COUNTY                                   14840      5531
IDAHO COUNTY                                    14769      6344
GEM COUNTY                                      11972      4578
GOODING COUNTY                                  11874      4591
FREMONT COUNTY                                  10813      5376
CLEARWATER COUNTY                               10390      4112
BLAINE COUNTY                                    9841      7319
FRANKLIN COUNTY                                  8895      3047
WASHINGTON COUNTY                                8803      3605
CARIBOU COUNTY                                   8695      3105
BENEWAH COUNTY                                   8292      3499
OWYHEE COUNTY                                    8272      3015
LEMHI COUNTY                                     7460      3452
BOUNDARY COUNTY                                  7289      2755
BEAR LAKE COUNTY                                 6931      2792
POWER COUNTY                                     6844      2558
VALLEY COUNTY                                    5604      5107
LEWIS COUNTY                                     4118      1807
LINCOLN COUNTY                                   3436      1343
CUSTER COUNTY                                    3385      2100
ADAMS COUNTY                                     3347      1580
BUTTE COUNTY                                     3342      1280
ONEIDA COUNTY                                    3258      1475
BOISE COUNTY                                     2999      2372
TETON COUNTY                                     2897      1245
CAMAS COUNTY                                      818       527
CLARK COUNTY                                      798       445
```

The Data Dictionary

CENSPAC causes a computerized documentation system, called a data dictionary, to assist to retrieving data items. The data dictionary file that the CENSPAC documentor builds and CENSPAC uses in processing data is a machine-readable counterpart to printed documentation. The machine-readable data dictionary defines data items or variables by specifying their location, length, format, and name. When a user references the name of a data item in a run, CENSPAC uses the data dictionary to extract that item from a data set and processes it. The content and structure of CENSPAC data dictionaries are discussed in more detail in Chapter IV.

Each 1980 Census Summary Tape File has a separate data dictionary associated with it, and these dictionaries are being included on the CENSPAC system tape as they are completed. In addition, the CENSPAC system tape contains other data dictionaries, such as First, Second, Fourth and Fifth counts from the 1970 Census.

However, CENSPAC does not require a data dictionary to retrieve data from a file. By immediately following a FILEIN command with an input ITEM command, the user can identify the location and characteristics of data items on a tape for which no data dictionary is available. For more information on using CENSPAC without a data dictionary, see the input ITEM command in Chapter II.

CENSPAC Applications

One of CENSPAC's purposes is to retrieve data from the five 1980 Census Summary Tape Files, called STFs. STFs 1 and 2 contain 100 percent census data, such as age, sex, housing tenure, rent and housing value. STFs 3, 4 and 5 contain sample data, such as statistics on income, occupation, travel to work, and so on, collected from approximately one-sixth of the population.

The information on the census summary tapes is in tabular form. Each file contains a number of tables. STF 1A, for example, contains 59 tables of data -- such as population by age, housing tenure by race -- for each geographic area. Complete information about each summary tape file is included in the technical documentation that a user receives upon purchasing a census summary tape from the Census Bureau. The record layout documentation is provided by the data dictionary and may be reprinted from the data dictionary file by the CENSPAC DOCUMENTOR program (see Chapter IV).

The applications of CENSPAC are not confined to 1980 census summary tapes. CENSPAC also can retrieve and process data from the 1970 Census tapes, the Annual Housing Survey, the General Revenue Sharing files, County Business Pattern files, the Census of Retail Trade, and other Census Bureau data files. Data dictionaries are available with CENSPAC for some of these files. (See Appendix 5.)

CENSPAC can be used with any data file if it is a sequential, fixed length record format file. This means that CENSPAC can be used to retrieve and process data from many local data files, such as a city's police precinct or school district files. Non-sequential, non-fixed length type files can also be processed using a CENSPAC I/O module interface. See Chapter 5 for additional information on I/O modules.

Computer System Requirements

CENSPAC is written in American National Standard Institute (ANSI) COBOL. The software requires up to 170K characters of main storage, direct access storage for the data dictionary files, and input and output devices to support the data files needed for particular runs. CENSPAC was developed on an IBM 370/168 under VS and also is operational on the UNIVAC EXEC-8 operating system, IBM OS, IBM DOS, Burroughs 7700, CDC 7000, UNIVAC 90/80, HONEYWELL 6600, DEC 10, DEC 20, DEC VAX, and APPLE II. The system is intended to be machine independent and users are encouraged to convert CENSPAC to other computer systems and to report these conversions to the Census Bureau.

CENSPAC Support Program

Several times each year the Census Bureau offers training seminars to help users learn about CENSPAC and its operation. The Bureau also will provide limited assistance to users in installing the system. Anyone with installation or use problems should write or call the Systems and Programming Branch, Data User Services Division, Bureau of the Census, Department of Commerce, Washington, D.C. 20233. Telephone: (301) 763-5242

CENSPAC User Registration

Those interested in receiving current information on CENSPAC corrections should return the CENSPAC registration form on the following page to Chief, Data User Services Division, Bureau of the Census, Washington, D.C. 20233.

CENSPAC USER REGISTRATION FORM

1. Name of CENSPAC user ____________________________________

   Organization ____________________________________

   Address ____________________________________

   ____

   Telephone ________________

2. Computer system used:

   Manufacturer Model ____________________________________

   Operating system ____________________________________

3. If CENSPAC was converted to operate on a system other than that on which it was previously operational, will you make the converted version available to others? ____________________________________

   If yes, who is the person to contact for information about this conversion?

   ____

   Were CENSPAC and DOCUMENTOR installed?

   CENSPAC ________________ DOCUMENTOR ________________

4. Please describe any problems you encountered in the installation of CENSPAC.

5. Did you contact the CENSPAC staff at the Census Bureau for installation assistance? ____________________________________

7. Where did you obtain your copy of CENSPAC?

   Census Bureau ________ State Date Center ______ Other __________

8. Describe any problems you encountered in using CENSPAC.

CHAPTER II

How to Use CENSPAC

This chapter describes how to use CENSPAC. The system works with only sixteen commands, and the rules regarding the use of each command and the way commands are linked together provide flexibility in processing data, preparing tables, and creating new files from the census summary tapes and other data files.

The first section of this chapter discusses CENSPAC rules. Sections 2, 3, and 4 describe the input, output, and processing commands. Section 5 describes CENSPAC system variables that are available to users.

Section 1: Rules of the System

CENSPAC Command Sets

CENSPAC commands are free form. Commands may begin in any column of a computer card or in any space of a card-image line on a terminal. Commas and spaces may be used freely because CENSPAC ignores spacing between commands.

The CENSPAC system expects to read cards or lines with no more than 80 characters. One or more commands are allowed per line, and commands may continue over several lines. Command names, keywords, data names, and literals cannot be split between lines, however.

A user can insert comments in a CENSPAC program by placing an asterisk (*) in column 1. The following 79 characters are then treated as comments.

CENSPAC Control Statements

Every CENSPAC control statement consists of a command name and one or more operands. The operands may be of the following types.

> Keywords: Keywords are special names which have predefined meaning in the CENSPAC system. Keywords include CENSPAC command names (FILEIN, IF, COMPUTE) and special CENSPAC operand names (SUM, BLKSIZE, etc.) See Appendix 3 for a complete list of CENSPAC keywords.

Data Names: Data names are descriptive tags assigned to variables in a data file. They may be two to eight characters in length, and may consist of alphabetic and numeric characters. The first character of all user defined data names must be an alphabetic character. Users will define their own data names when new variables are created or when a data dictionary is not available.

Tables of data such as those found in the STFs in which multiple data cells are associated with a single data name are treated as multidimensional arrays. The first letters of the data name must be 'TAB' followed by one to five alphanumeric characters. Subscript notation, enclosed in parentheses, indicates the cell or cells of the array to be used in an operation. See the input and output ITEM commands for more information on subscripting arrays.

Literals: Literals are strings of alphabetic and/or numeric characters that are used in CENSPAC ITEM, COMPUTE, and IF statements to modify data or to make comparisons. Literals are of two types, numeric or alphanumeric. An example of a numeric literal is "1.4" in the following COMPUTE statement:

```
COMPUTE FACTOR = FACTOR * 1.4
```

Examples of a character or alphanumeric literal are '0049' and 'A' in the following IF statement:

```
IF EDNUMBER = '0049' AND EDSUFFIX = 'A' EXCLUDE
```

Numeric literals are used in a numeric context. A leading sign (+ or -) can be used, and a decimal point can be coded where necessary. Only numeric characters (0 through 9), a sign (+ or -) and a decimal point (.) may appear in numeric literals. Decimal fractions cannot have the decimal point as the first character of a literal: .12 is incorrect; 0.12 is correct. The maximum length of numeric literals is 18 characters, including the sign and decimal point, if present. The maximum value allowed in a numeric literal is 12 significant digits to the left of the decimal and six digits to the right.

Alphanumeric literals or character literals may contain any character except the quote, and the entire literal must be enclosed by quotes. They must be contained on one card or card-image line; therefore, the maximum size is 78 characters allowing two positions for single quotes. Labels are processed as 40 character literal strings and consequently literal strings in the LABEL command must contain a maximum of 40 characters.

NOTE: CENSPAC requires the use of quotes to begin and end character literal strings. Throughout this manual the words quote (singular) and quotes (plural) are used when referring to the delimiters for alphanumeric literals. For IBM OS/VS, IBM DOS, UNIVAC EXEC-8 systems the word quote (or quotes) is to mean a single quote (character '). On other operating systems, Burroughs, CDC and DEC (at least), the double quote is used (character "). The usage of this character is consistent with each operating systems usage of the quote in COBOL.

CENSPAC Operators

CENSPAC performs arithmetic and character operations with the COMPUTE command. Logical and comparative operations are performed with the IF command.

Arithmetic operators: The arithmetic operators +, -, *, and / may be used in arithmetic expressions.

Logical operators: The operators NOT, AND, and OR may be used in logical expressions.

Comparative operators: The following comparative operators may be used in logical expressions:

<, LESS THAN
=, EQUAL
>, GREATER THAN

Arithmetic and logical operations are processed with the following order of priority:

```
* /  highest priority
+ -
<=>
AND
OR   lowest priority
```

If two or more arithmetic or logical operators of the highest priority appear in an expression, CENSPAC processes the operators from left to right. The left-most operation is performed first, and each succeeding highest priority operator to the right is processed next. Expressions enclosed in parentheses are always processed first, before they are linked to operations outside the parentheses.

Character data, as well as numeric data, can be COMPUTED. When using character data or character literals only the equal sign (=) can be used as an operator. It assigns the value of a variable or literal string to the new variable. For example, the following COMPUTE commands use alphanumeric data:

```
COMPUTE  COUNAME  =  AREANAME
COMPUTE  DASHES   =  '------------'
```

Limitations on the Use of Numeric and Alphanumeric Data

The maximum numeric value that CENSPAC can work with must be contained in a field of 18 decimal digits, with 6 decimal places within the 18 digit field. In the IBM 360/370 compatible CENSPAC system, all user-defined CENSPAC numeric variables (created by the ARRAY and COMPUTE commands) are defined as 10 byte, signed, packed decimal (computational), items with 6 implied decimal places and 12 decimal digits to the left of the decimal point. User-defined variables on the UNIVAC 1100 system are 7 character (63 ASCII bits) fields of 18 digits with 12 decimal digits to the left of the decimal point and 6 digits to the right. All user-defined numeric data items are converted into one of these forms before processing. Therefore, the maximum value that CENSPAC can work with in user-defined variables is ± 999999999999.999999. CENSPAC predefined system variables (see Chapter II, Section 5) are defined in computational form with 9 decimal digits and no decimal places.

The CENSPAC system needs two attributes of data items identified in order to be able to process the data correctly. The first attribute is data type which simply identifies whether a data item is alphanumeric (may contain any character), numeric (contains only numeric characters), or is in a special computational numeric format such as binary. This attribute is used for all data items and controls the definitions of the data fields within CENSPAC. The second attribute is data category and it identifies the use of the data item. The data category attribute is used only by the TOTAL command. Specifically, data items may be 'code' items such as geographic codes and area names which cannot be aggregated, or they may be numeric, which can be aggregated and arithmetically manipulated. In the data dictionaries provided by the Bureau with CENSPAC, all geographic data items are identified as 'code' and all tables are identified as 'numeric'. The documentor options of 'DT' (data type) and 'DC' (data category) can be added to the DOCUMENT command to produce a fuller document report which identifies the data type and data category attributes of documented data items. See Chapter IV for more information on the documentor.

Data Files and Multiple Types of Records

CENSPAC can process data files with multiple record types. Though the system has no commands oriented specifically towards multiple record files, its file processing logic allows data from one record to be held for processing with subsequent records. All records are read into a single work area defined within CENSPAC. The data items required for current processing are moved from the input area to a common work area where they are available for reference by the user. All user-defined items are also placed in this common work area. User-defined data items in the work area are only modified when they are referenced as the result of a COMPUTE command. These data items are not automatically cleared, except for initialization at the beginning of a run.

Suppression on 1980 Census Summary Tapes (STFs)

On the 1980 Census STFs, when population or housing counts for a geographic area are so small that identification of persons or households would be possible, the information is suppressed.

The cells containing suppressed data are filled with zeros, and a series of flags on the record indicate suppression. CENSPAC does not process suppressed cells differently than other cells. The user must decide how to handle suppression, either ignoring it by processing the zero-filled cells as if there were no suppression, or by processing tables with suppressed cells separately. For example, users may exclude tables with suppressed data from a report, or they may print tables with suppressed cells, labeling the cells that are affected.

Users can differentiate suppression zeros from data zeros on the 1980 Summary Tape Files by checking suppression flags for the table universe. When the potential for suppression exists, the user can have the suppression flag printed on a report or specify special processing for records containing suppressed data. See Example 7 in Chapter III for a CENSPAC run that uses suppression flags.

A few data items on the summary tapes are never suppressed, such as total population and housing unit counts, the number of persons by race and Spanish origin, the number of year-round housing units, and the number of occupied and vacant housing units. Complete information on suppression can be found in the technical documentation that accompanies each summary tape.

Suppression is flagged differently on the 1970 census tapes. CENSPAC analyzes the individual tables that are subject to suppression and sets a CENSPAC system variable equal to 0 or 1 to indicate suppression. For more information on this system variable, see "$SUPXXX" in Chapter II, Section 5, "CENSPAC System Variables".

CENSPAC Error Messages

The first operation that CENSPAC performs in a run is to check control statements for correct syntax. When an error is found, a numbered error message is printed and the checking continues. When all commands have been read and analyzed, a summary is printed showing the number of commands processed and the number of errors found. If any errors are detected, a description of the error is printed. If no serious errors are found, CENSPAC will continue processing the run. All errors with message numbers below 100 are serious and will result in program termination. Error message numbers 100 and above are warning messages only. See Appendix 1 for a complete list of CENSPAC error messages.

<u>CENSPAC Command Library</u> (Not available in May or August 1980 versions of CENSPAC.)

Individual CENSPAC commands or groups of CENSPAC commands can be stored in the CENSPAC command library (actually part of the master data dictionary) for later use in any number of runs. The CENSPAC Documentor program must be used to create or update individual library members (see Chapter IV, the CENSPAC Documentor).

The CENSPAC commands are stored in the library as label information for a data name associated with the 'MACLIB' data dictionary. The data name is used as the library member name in a CENSPAC run requesting the library member. The character '&' (ampersand) is prefixed to the library member name when it is referenced in a CENSPAC run. A reference at any point in a CENSPAC run to a library member will have all the information from that member placed into the CENSPAC run at that point. For example, the following CENSPAC commands:

```
FILEIN STF1
REPORT
&LIB1
```

Actually produce the following CENSPAC run.

```
FILEIN STF1
REPORT
*  TEST OF DATA DICTIONARY
*  LIBRARY FUNCTION
ITEM 'TABLE 1    ' H40.0 TAB1
ITEM U40.0 TAB1 ITEM
'The Stratifier is:'  S40.0 TAB1 (1)
ITEM
ITEM C25.0 TAB1 (C) TAB1 (C) ITEM
&LIB1
```

The first 2 commands, the FILEIN and REPORT are as were provided as card-image input. The following 7 lines of CENSPAC comments and commands are the result of the reference to '&LIB1'. The data name 'LIB1' is in the data dictionary 'MACLIB' and consists of the following CENSPAC Documentor information:

```
D LIB1      .* TEST OF DATA DICTIONARY
           :* LIBRARY FUNCTION
           :  ITEM  Table 1  'H40.0 TAB1
           :  ITEM U40.0 TAB1 ITEM
           :  'The Stratifier is :' S40.0 TAB1 (1)
           :  ITEM
           :  ITEM C25.0 TAB1 (C) TAB1 (C) ITEM
```

The last line of the CENSPAC run, the library request itself, '&LIB1' is provided following the library information that was retrieved as a result of it. Up to 40 lines of commands can be placed into a single library member. Each of the command lines can be up to 40 characters long. For further information on the library function and the Documentor program; see Chapter IV, Special Data Dictionaries.

Section 2: CENSPAC Input Commands

There are four CENSPAC input commands: FILEIN, ITEM, MATCH, and ENDM. Only the FILEIN command is required in every CENSPAC run.

CENSPAC may be used with or without a data dictionary. When a data dictionary is available, the only input command required is FILEIN, followed by the name of the dictionary. When no data dictionary is available, the input ITEM command is required to identify data items for input. It must immediately follow the FILEIN command.

The commands MATCH and ENDM are required only when two files are to be merged.

The input commands are described here in alphabetical order.

ENDM

The ENDM command is required in every MATCH run. It terminates the File B command set when a user is matching two files.

Syntax: ENDM

The ENDM command has no operands. (See the MATCH command discussion and Example Run 10 for more information.)

FILEIN

The FILEIN command identifies the input data file to be processed by CENSPAC, as well as the data dictionary to be used. FILEIN must be the first command in every CENSPAC run. If one file is to be processed, only one FILEIN command is allowed. If two data files are to be processed with MATCH commands, two FILEIN commands, as well as the corresponding MATCH commands, are required.

Syntax: FILEIN DICTNAME dictname IOMODULE iomodulename RECSIZE recsize
BLKSIZE blksize DEVICE device FILENAME filename filetype WORKLMT nnnn

DICTNAME: The name of the input file data dictionary is required in all CENSPAC runs. It identifies the data file and data dictionary to be used in processing. The optional keyword, DICTNAME, can precede the data dictionary name. See Appendix 5 for the list of data dictionaries available from the Census Bureau. Even if no dictionary is available, a dictionary name must be specified. CENSPAC uses dictname to determine file processing logic. If CENSPAC recognizes dictname, i.e. if it corresponds to a predefined name available in the master data dictionary, such as those for the census summary tapes, the appropriate processing logic is selected. If CENSPAC does not recognize dictname, or if it is not a file that requires special input processing logic, standard fixed length record format processing is assumed. If no data dictionary is available, warning message 101, will be issued and processing will continue.

IOMODULE: This operand is optional. The keyword IOMODULE followed by the I/O module subroutine name, such as IOMODULE STF1 or IOMODULE STANDARD, allows the user to override the processing logic implied by the DICTNAME operand. If IOMODULE is not provided, the input processing logic to be used will be determined from the dictname operand. IOMODULE STANDARD causes CENSPAC to read the file without using an IOMODULE subroutine even if one is implied by the dictname provided. That is, DICTNAME STF1 implies calling the STF1 I/O module to process the record segments in STF1. The use of IOMODULE STANDARD would override this and read the data as though each segment were a complete record. See Appendix 2 for additional information on the I/O modules.

RECSIZE: This operand is optional. It is required only when a data dictionary is not available. The keyword RECSIZE, followed by a 2 to 5 digit numeric value, indicates the logical record length of the input data file. By specifying RECSIZE, a user can override record size information in a data dictionary. If no data dictionary is available and RECSIZE is not provided, a record length of 80 characters will be assumed. Specifying RECSIZE has no effect on a CENSPAC run when processing census segmented files (summary tape files) or when IOMODULE (other than IOMODULE STANDARD) was specified. The I/O modules for these files contain the record size information.

BLKSIZE: This operand is optional. It is not required when a data dictionary is available or the information can be supplied through job control language or some other system interface. It also is not required if the blocking factor equals one and RECSIZE was supplied. The keyword BLKSIZE, plus a 2 to 5 digit numeric value, indicates the blocksize, in characters, of the input data file. By specifying BLKSIZE, a user can override blocksize information in the data dictionary and in job control language or other system interface. Specifying blksize has no effect on a CENSPAC run when processing census segmented files (summary tape files). The I/O modules for these files contain the blocksize information.

DEVICE: This operand is optional. The keyword DEVICE is followed by the code for the type of input device used. If omitted, TAPE device is assumed. Device codes are:

TAPE	standard labeled magnetic tape
TAPU	unlabeled magnetic tape
DISK	sequential mass storage
CARD	card or card-image file

On an IBM OS/VS compatible system DEVICE can be specified, but because the operating system, handles all device-dependent file attributes, the operand has no effect on a CENSPAC run.

FILENAME: This operand is optional. The keyword FILENAME is followed by the external filename (DD-name for IBM OS/VS) so that CENSPAC can retrieve an input data file. If FILENAME is not specified, the default filename used by CENSPAC is 'AINP' for the first FILEIN command and 'BINP' for the second (two FILEIN commands are used only in a MATCH). On census summary tape files and other files using I/O modules, the default filename 'AINP' is always used, and a filename operand cannot override the default name.

filetype: A filetype code of 'IBM' can be included on UNIVAC EXEC-8 systems to identify the input file as an IBM EBCDIC file. Use of this code allows IBM EBCDIC standard labeled or unlabeled tapes to be processed. This operand has no effect when processing a file that uses an I/O module (other than IOMODULE STANDARD).

WORKLMT: This parameter controls the size of a CENSPAC system workfile that holds data name and label references during the run. The keyword is followed by a 2 to 5 digit number indicating the number of 80 character records that are to be allocated and initialized for this run. The default value is 100, which including the 50 entries stored internally in CENSPAC, allows for up to 150 data names and/or table names in a run. If this number of names is exceeded, a CENSPAC error message will be issued and the run will have to be re-done with a larger WORKLMT value. Since each record in this work file is initialized at the beginning of a run, the value of WORKLMT should not be excessively large or unnecessary processing will be done just to initialize work space that will not be utilized.

(Note: The IOMODULE, iomodule, filetype, DEVICE, DICTNAME, and WORKLMT operands and keywords are not available in May or August 1980 versions of CENSPAC. The dictname parameter, however, is required for all versions of CENSPAC.)

Example

FILEIN STF1

The 1980 Census Summary Tape File 1 (STF1) is to be used. Since a data dictionary is available for this file, no additional information is required.

FILEIN DICTNAME CITY RECSIZE 100 BLKSIZE 800 FILENAME CITYDATA

A data file is referred to using an external file name (DD-name) CITYDATA. The file contains records 100 characters in length, with a blocksize of 800 characters. DEVICE is not specified, so TAPE is assumed. The DICTNAME CITY is not recognized by CENSPAC as a file that requires special processing so a standard processing format is assumed.

Because some of the keywords on the FILEIN commands are optional, the same results as above can be achieved by providing the following FILEIN command.

FILEIN CITY RECSIZE 100 BLKSIZE 800 CITYDATA

The keywords DICTNAME and FILENAME are not required, the information can be conveyed to the FILEIN command by supplying the operand value without the keyword.

ITEM

The input ITEM command is optional, required only when a data dictionary is not available or when further data definitions are needed. The input ITEM command defines input data items as they appear in the input file. The command must immediately follow a FILEIN command to identify the input data items.

When the ITEM command is used, only the data items that need additional definitions need to be identified. If a data name in an ITEM command duplicates a data dictionary data name, the data dictionary information is overriden by the ITEM command definition and data dictionary information for that item name is not available for that run.

Each ITEM command and its continuation lines, if any, identifies one input record. Multiple ITEM commands redefine the same input record. Each ITEM command that follows the FILEIN command provides a different format for reading the same record. This is useful if the user has a hierarchical file. The user can define a record in several ways, and then use IF statements to decide which format should be used for further processing.

Syntax: ITEM ntw.d dataname

ntw.d: Data items defined by the input ITEM command must be formatted, with computational form (t), size (w), and number of decimal places (d) specified.

'n' is the repetition factor, used only if several items with identical formats are to be defined.

't' indicates computational form. It identifies an item an numeric, alphanumeric, or as an unused space. There are six computational forms allowed in the input ITEM command:

F: identifies numeric data.
A: identifies alphanumeric data.
O: identifies signed numeric data and recognizes an overpunch sign on the rightmost character of an item.
X: identifies unused spaces.
B: identifies signed binary format.
@: provides a 'tab' function to skip to a specific character location. (@ is not available in May or August 1980 versions of CENSPAC).

'w' identifies the width of the data item, from 1 to 32,767 characters (only alphanumeric items can be longer than 18 characters).

'd' identifies the number of decimal places in a data item. The 'd' ranges from 0, for no decimal places, to 6, the maximum decimal range on CENSPAC. If decimal places are not needed, such as with alphabetic items or whole numbers, the 'd' must be specified by a .0. Except when defining binary data, the value of 'd' must always be less than the value of 'w'.

dataname: The data name in an input ITEM command identifies individual data items or tables of data (arrays). Arrays may be up to six dimensions. A two dimensional array for example, is sex by marital status. A three dimensional array is sex by marital status by age. The data name of an array must begin with the letters 'TAB', followed by up to 5 alphanumeric characters identifying the particular array, i.e. TABXXXXX. The array name in the input ITEM command must be followed by subscripts which shows the maximum dimensions of the array. For example, a three dimensional array, TAB001, sex (2 categories), by marital status (5 categories), by age (10 categories), would be subscripted as follows: TAB001 (2,5,10). This array has 100 data cells. If each cell has 8 characters, the ITEM command would be formatted as follows:

```
ITEM 100F8.0 TAB001 (2,5,10)
```

Once this table is identified by the input ITEM command, the user can reference TAB001 in subsequent CENSPAC commands, retrieving one or more cells from the array. Spacing within subscripts is necessary to identify the size and number of dimensions in an array. A comma, and/or blank space, may be used to separate the subscripts within the parentheses.

Data cells in CENSPAC arrays are indexed as they are in the 1980 Summary Tape File data dictionaries (see the STF technical documentation). Data cells in arrays, or tables, are stored and processed successively, with the last subscript varying most rapidly, and the first subscript varying least rapidly. For example, the array TAB10 (2,26) is stored in the following order: TAB10 (1,1), TAB10 (1,2),...TAB10 (1,26), TAB10 (2,1), TAB10 (2,2),...TAB10 (2,26).

Example

```
FILEIN DICTNAME INPUT1 RECSIZE 240 BLKSIZE 4800 FILENAME STDATA
ITEM X1.0 A2.0 STATE A3.0 COUNTY @121.0 22F8.0 TOTPOP TOTHSG TABAGES (2, 10)
```

Here the state and county codes, two numeric items, TOTPOP and TOTHSG, and an array, TABAGES, are defined. These data items are part of an input record of 240 characters. Because they are the only data items needed for this run, only they need to be defined. The X1.0 specification causes character location 1 to be skipped, following this, 'A2.0 STATE', defines a 2 character alphanumeric field that is to be called 'STATE'. 'COUNTY' is defined as a 3 character field immediately following the state data item. The numeric items are total population and total housing units. Each is 8 characters long, with no decimals. The '@121.0' specification positions the definition of the next data item to character location 121. The first numeric item begins in location 121 and the second item begin in location 129. The array that follows consists of 10 age groups for both categories of sex. The 20 cells are 8 characters each and contain no decimals, so that TOTPOP, TOTHSG, and TABAGES all have the same format.

MATCH

The MATCH command is required when two files are to be merged.

Syntax: MATCH sort-option key-1 key-2...key-7

sort-option: This operand is optional. By specifying SORT, the input file can be sorted in ascending order on the identified keys before matching takes place. This operand can only be used with files that require no special I/O modules; therefore, the census segmented files cannot be sorted using the sort-option.

The use of the SORT option in a MATCH command does not affect the later use of the SORT command. The SORT option may be used with each MATCH command, and the SORT command may appear later in a CENSPAC run, for a maximum of three sorts. See the SORT processing command in Section 4 of this chapter for information on work space requirements for sorting.

key-n: This operand is required in a match. The keys specify the fields on the records to be matched. If the user has specified the sort option, the same keys will be used for sorting. The keys specified are combined into a single match key by CENSPAC. The combined length of the keys specified for one input file should equal those specified for the other input file. The keys must be listed in hierarchical sequence, with the major key listed first, followed by intermediate and then minor keys. CENSPAC assumes that input data are in ascending sequence on the key fields. If data are not in ascending sequence, the SORT option must be invoked.

The keys on the two MATCH commands should be identical in structure, even if the data names are different. For example, on file A the tract code might be identified as 2 items -- the 4 character basic tract code and the 2 character suffix code. On file B the tract code might be identified as a single 6 character data item. In this situation, the first MATCH command would reference TRACT1 and TRACT2, which combine to form the 6 character tract code. The second MATCH command would reference TRACT which would form the equivalent 6 character tract code.

MATCH and FILEIN commands are required for both of the input files to be merged. The first MATCH and FILEIN commands define file A. The second MATCH and FILEIN commands define file B.

There are three groups of commands in each MATCH run. The first group defines file A; the second group defines file B, and the third group controls the processing of the matched files including REPORT and/or FILEOUT output. When the FILEIN command for file B appears, the file A command group is terminated. The ENDM command is required to end the file B command group.

```
GROUP 1   { FILEIN A
          { MATCH

          { FILEIN B
GROUP 2   { MATCH
          { ENDM .

GROUP 3   { FILEOUT or REPORT
          { ITEM
```

The file A and B command groups each must begin with the FILEIN command, immediately followed by ITEM and MATCH commands, which identify the records to be matched. The command ENDM follows the file B commands and signals the end of the file B group.

The IF and COMPUTE commands are the only commands, except for FILEIN and MATCH, allowed in the file A and B command groups, and they must appear after the MATCH command. The function of the IF command is limited in these command groups. The INCLUDE and EXCLUDE operators cannot be used to affect the matching process and are not allowed before the ENDM command.

In processing the third command group, CENSPAC may encounter a data name that appears in both files. For example, 'STATE' may be the data name for the state code in each file. Likewise, Table 1, 'TAB1', may occur in each file, but contain different data. In order to distinguish data items that have identical names, a file-id may be used in the commands that follow the ENDM. Data items from file A may be prefixed with 'A.', and data items from file B *must* be prefixed with 'B.'. The 'A.' prefix is assumed when no prefix appears, and it may be omitted when a user is referencing file A items. The 'B.' file-id must be used to identify data items from the B file. The file-id is not required in the input command sets for file A and B since those commands are clearly associated with just one file.

The predefined system variable $MATCH (see $MATCH in Chapter II, Section 5) is used to indicate whether a match was successful. While its use is not required, $MATCH is the means by which a user can tell that a match was successful. The possible values for $MATCH are:

0 A match was successful. All variables from files A and B are available.

1 The key for the current A record is lower than for the current B record. The data for both the current A record and the current B record are available can be referenced in further processing.

2. The key for the current B record is lower than for the current A record. The data for both the current A record and the current B record are available can be referenced in further processing.

The MATCH function assumes that both input files are in ascending sequence and that records in both files have roughly a one-to-one correspondence. They key from the file A record is compared to the key from the file B record. CENSPAC sets the predefined system variable, $MATCH, to 0, 1, or 2 depending on the results of the attempted match.

After processing the current pair of matched records, the $MATCH value is used by CENSPAC to determine from which file(s) to read the next record(s). If $MATCH is 0, indicating a successful match, a record is read from both the A and B files. If the value of $MATCH is 1, indicating a low value key from the A record, a record will be read only from the A file and CENSPAC will attempt to match it with the currently available B record. If $MATCH is equal to a value of 2, then a record from the B file only will be read before attempting a match.

The user can adjust the value of $MATCH in order to control which file is to be read and in doing this can have CENSPAC properly execute a one-to-many match. Basically, the record to be matched with multiple records is held for further matching by setting the value of $MATCH to read only a record from the other file. (See MATCH Example 2 below and the flowchart on page 187).

Example 1

```
FILEIN STF1
MATCH COUNTY BG EDNUMBER
FILEIN STF3
MATCH COUNTY BG ED
ENDM
IF A.SUMRYLVL <'15' OR A.SUMRYLVL> '16' OR B.SUMRYLVL <'15' OR B.SUMRYLVL>
  '16' EXCLUDE
IF $MATCH = 0 FILEOUT
ITEM COUNTY BG B.ED TAB004 A.TAB016 B.TAB038
```

The user is merging the 1980 Census Summary Tape File 1 (STF 1) with data from Sumary Tape File 3 (STF 3) for all block groups and enumeration districts in California. The IF test on the SUMRYLVL field will allow only block group ('15') and enumeration district ('16') records to be processed regardless of the value of $MATCH. The user wants to create a file containing selected complete count (STF 1) and sample (STF 3) data in each record. In this run, CENSPAC assumes that STF 1 and STF 3 are in ascending order. In processing the data, a record is read from file A (STF 1) and a match is attempted with the current record from file B (STF 3).

If the match is successful ($MATCH = 0), a new record is created containing the county code, block group code, enumeration district code, Table 4 and Table 16 from STF 1, and Table 38 from STF 3. The output data items are specified by the ITEM command. Note that COUNTY, BG, and TAB004 do not have a file source prefix whereas TAB016 and TAB038 are prefixed. If no prefix is used, the source of the data is assumed to be file A.

Example 2

```
FILEIN CNT1
MATCH COUNTY70
FILEIN GEOREF RECSIZE 80 BLKSIZE 800
ITEM A3.0 COUNTY A30.0 NAME
MATCH COUNTY
ENDM
IF $MATCH = 0 INCLUDE
REPORT
ITEM COUNTY B.NAME BLKGRP TAB18 (*,C)
IF $MATCH =0
COMPUTE $MATCH =1;
```

In this example of a one-to-many match, a file of county records will be used to attach county names to a file of block group records. The 'A' input file is 1970 First Count containing block group records. 'GEOREF' is the 'B' input file containing county records with area names. The records are being matched on county codes and, if successful ($MATCH=0), the county code, name, block group code, and Table 18, will be printed. To hold the 'B' record with the county name, $MATCH is then set to 1, indicating to CENSPAC that the 'A' record should be released and another match should be attempted with the current 'B' record and the next record from the 'A' file. This is done only when a valid match is being processed so that when the 'A' file advances to the next county, a non-match condition will occur and allow the 'B' file to be advanced also.

Section 3: CENSPAC Output Commands

The CENSPAC output commands are FILEOUT, REPORT, ITEM and HEAD. In a CENSPAC run, only the input and output commands are required. CENSPAC processing commands are used when data are to be manipulated (SORT, COMPUTE, etc.) before being placed into an output file or report.

In every CENSPAC run the user must specify the type of output, either FILEOUT for a new data file, or REPORT for a printed report. More than one REPORT and/or FILEOUT may be requested in a single run. Up to 999 separate reports and output files can be produced in a run.

The output ITEM command is required in every CENSPAC run. One or more ITEM commands must follow each FILEOUT and REPORT command. The ITEM command identifies the data items the user wants to place on an output data file or report. In the format statements associated with the ITEM command, the user can specify the design of a report, including the labeling of tables.

The HEAD command which provides page headings for a report, is not required.

The Census Bureau has compiled a list of commands that will retrieve, with labels, every table on 1980 Census Summary Tape File 1. To obtain this list of commands, write to the Systems and Programming Branch, Data User Services Division, Bureau of the Census, Department of Commerce, Washington, D.C. 20233.

The output commands are described here in alphabetical order.

FILEOUT

The FILEOUT command produces an output data file. At least one FILEOUT command, or one REPORT command, or both, must appear in every CENSPAC run. Multiple FILEOUT commands are allowed in one run.

Every FILEOUT command must be followed by one or more ITEM commands. Each ITEM command defines a separate record format to be placed in the output file.

Syntax: FILEOUT BLFACTOR block-factor DOCUMENT dictname RENAME
DEVICE device FILENAME filename

BLFACTOR: This operand is not required if the block size for an output data file is to be equal to the record size, that is if the output file is to contain one record per block. The keyword BLFACTOR, followed by a one to five digit numeric value, indicates the number of records per block for the output data file.

DOCUMENT: This operand is optional. The keyword DOCUMENT requests the creation of documentation for the output file produced by the FILEOUT command. The DOCUMENT operand must be followed by a data dictionary name (dictname). This links the CENSPAC run with the CENSPAC DOCUMENT command. The DOCUMENT operand creates documentation in the form of data definition statements. These statements must be processed by the CENSPAC Documentor program to create the machine-readable data dictionary and the printed data dictionary report. The data names, data attributes (length, type, etc.) and descriptive labels will be taken directly from the data dictionary associated with the input file. Additional labeling information may be provided by LABEL commands if desired. The documentation will reflect the correct beginning positions on the new output record(s) and the sizes and computational format of data items based on the format specifications provided in the output ITEM command associated with the FILEOUT command.

In some cases duplicate data names may appear on the output file when the same data name is included in two input files that are being matched, or when a data item is referred to more than once in the same output file. Duplicate data names are not accepted by the Documentor program. The CENSPAC FILEOUT document function will change data names in order to ensure that they are unique.

Two options are available to solve the problem of duplicate names. First, CENSPAC automatically checks for duplicate names and, when they are encountered, a new name is generated by CENSPAC. The name generated for individual data items is ITEMnnnn, where the letters 'ITEM' are the first part of the name and a sequential 4-digit number follows.

For tabular data items, the three letter prefix 'TAB' is used, followed by a sequential 5-digit number (TABnnnnn). When data items or tables are renamed, CENSPAC produces a cross-reference listing of the original data names and the resulting new names. When renaming is done without specifying RENAME, only the duplicate data names are renamed.

The second option to process duplicate data names is to use the RENAME operand. When the RENAME operand is specified (see below), all data names, not just duplicates, will be renamed atuomatically according to the above rules.

RENAME: This operand is optional and is used in conjunction with the DOCUMENT operand. The RENAME option instructs CENSPAC to rename all data items, not just those with duplicate names. Individual data items will be renamed in sequence starting with ITEM0001, ITEM0002, etc,; tabular data items will be renamed TAB0001, TAB0002, etc.

DEVICE: This operand is optional. The keyword DEVICE is followed by the code for the type of output device used. If omitted, TAPE is assumed. The valid device codes are:

TAPE	standard labeled magnetic tape
TAPU	unlabeled magnetic tape
DISK	sequential mass storage
CARD	card or card-image file

On IBM OS/VS compatible operating system DEVICE can be specified, but because the operating system handles all device dependent file attributes, the operand has no effect on the CENSPAC run.

FILENAME: This operand is optional. The keyword FILENAME is followed by the filename of the output file. If omitted, the default filename is 'F001'. If a filename is specified, and a name other than 'F001' is used, the job control statements or runstream used to execute the CENSPAC must reference the filename. For IBM JCL, the following must be included:

```
//GO.filename DD (other required JCL parameters)
```

If several output files are being created, with multiple FILEOUT commands, the subsequent default filenames are 'F002', 'F003', 'F004', etc.

(Note: DOCUMENT, dictname, RENAME, and DEVICE keywords and operands are not available in May or August 1980 versions of CENSPAC.)

By using IF and COMPUTE commands with a FILEOUT command, a user can conditionally process and write particular data items onto the output file. A user also can request, in one CENSPAC program, that some data items be placed on a new file while others are placed on a report.

When multiple ITEM commands follow a FILEOUT command, multiple output records will be produced for each input record processed. When a user provides multiple ITEM commands in a CENSPAC run, the ITEM command defining the largest output record must be listed first.

The predefined system variable $LCTF is available for counting the number of records placed on a new file. See Chapter II, Section 5, for more information on this variable.

Example

```
FILEIN STF1
FILEOUT BLFACTOR 10 DOCUMENT EXAMPLE1
ITEM STATE COUNTY TAB4 TAB8
```

This FILEOUT command will produce an output file with 10 records per block. The new file will contain records with state code, the county code, and all data cells from Table 4 and Table 8. Output file documentation is being requested, the document name provided is EXAMPLE1. The data definition statements created by CENSPAC as result of the DOCUMENT request can be processed by the CENSPAC Documentor program to have the 'EXAMPLE 1' data dictionary added to the master data dictionary file. The data dictionary created by this process can then be used in later CENSPAC runs.

HEAD

The HEAD command is optional. It specifies the headings that are to appear at the top of each page of a report. The HEAD commands must immediately follow a REPORT command. Each HEAD command identifies one line of heading on a printed report. Several HEAD commands may be used to provide more than one line of headings. Heading lines are formed starting at the left margin and continuing to the right. Literals and data names may be used in headings. A HEAD command cannot be subordinate to an IF command, it will be printed in a REPORT as needed at the top of each new page.

Syntax: HEAD format dataname THRU datanamel literal

format: This operand is optional. All rules applicable to the formatting of the output ITEM command also apply to the HEAD command. See the output ITEM command for the format specifications for HEAD.

dataname: This operand is optional. It identifies the name(s) of the variable(s) whose value is to be placed in the report heading. Data items occupy space based upon the length of the item as specified in the data dictionary, unless format specifications are used to override the data dictionary. Reserved words $DATE and $PAGE may be used as data names in the HEAD command. $DATE provides the current date printed as an 8 character field in the location of $DATE. $PAGE provides the current page number printed in the location of $PAGE. See Chapter II, Section 5, for more information on these predefined system variables.

THRU datanamel: A series of data items defined in the data dictionary can be specified using the keyword 'THRU'. The data items must be in consecutive sequence, as defined by a data dictionary, with dataname referring to the first item to be output and datanamel referring to the last. Fillers (blank padding in the record) are excluded from the items. Therefore, to duplicate an entire record, fillers must be explicitly referenced. Fillers are those data names in a data dictionary that begin with the letters 'FILLER'. The 'THRU' function is not available in May or August 1980 versions of CENSPAC.

literal: This operand is optional. Alphanumeric literals can be included in the HEAD command. The literal can contain any character except the quote and must be bounded by quotes. A literal in the HEAD command can be up to 78 characters long (one card-image with a leading and a trailing quote).

A HEAD command with no operands will produce a blank line in the REPORT heading.

Example

```
FILEIN STF3
REPORT
HEAD 'EDUCATIONAL ATTAINMENT BY'
HEAD 'SELECTED INCOME LEVELS FOR' A30.0 AREANAME 'COUNTY'
ITEM TAB8
```

In this example, two lines of heading would be printed at the top of each page in the report. AREANAME is the dataname in the STF3 data dictionary for the name of a geographic area. The specification 'A30.0' overrides the data dictionary definition of AREANAME, limiting the space for AREANAME used in the report to 30 characters.

ITEM

The output ITEM command is required after every REPORT or FILEOUT command. It identifies the data items to be placed on an output file or in a report. Multiple ITEM commands may follow each REPORT or FILEOUT command.

Following the REPORT command, each ITEM command identifies a line of output. Following a FILEOUT command, each ITEM command identifies a separate record on the output file.

Data are output from left to right in the print line or file record corresponding to the sequence of operands provided on the ITEM command. A tab function is available using the 'at sign' character (@) followed by the desired position number in the print line or output record. The position indicator must be formatted with a zero to the right of the decimal point. For example, @67.0 would cause the next item to be printed half way across the page, starting in location 67.

The ITEM command can also be used to insert blank lines in a report. The user does this by entering ITEM on a line with no operands following it.

Syntax: ntw.d dataname THRU datanamel literal

ntw.d: This operand, the format operand, is optional. If omitted, items will be output in their data dictionary form, or, for newly computed variables, in the form F18.6. The default format for numeric items is F.

If a user wants to format data items for output, the computational form (t), size (w), and number of decimal places (d) must be specified.

'n' is the repetition factor, used only if several items with identical formats are to be defined.

't' indicates computational form or formatting type. It identifies an item as numeric, alphanumeric, or as an unused space. There are 13 codes for this parameter allowed in the output ITEM command:

F: identifies numeric data; allows for a leading minus sign (-) with leading zero suppression.

A: identifies alphanumeric data.

O: identifies numeric data; allows an overpunch sign to the right of an item to be entered on a new file (FILEOUT only).

X: identifies blank spaces.

Z: identifies numeric data; no signs allowed; leading zeros suppressed.

-: identifies numeric data; leading minus sign if negative; no zero suppression.

$: leading dollar sign and leading zero suppression; no signs allowed.

B: identifies sign binary format. B2.0 is equivalent to I*4 (FILEOUT only).

@: provides the 'tab' function to skip to specific locations ('@' function not available in May or August 1980 versions of CENSPAC). Four additional codes, H,U,S, and C are available for labeling reports. See ITEM command and labeling reports on the following pages.

'w' identifies the width of the data item, from 1 to 32,767 characters (only alphanumeric items can be longer than 18 characters). Where the width ('w') field provided in the output ITEM command specifies a length different from the definition of that item, additional padding or truncation will occur. For numeric data, when being formatted into a larger field CENSPAC will treat the data value as though it has additional leading zeros (which may be blank suppressed if using the 'F' type code). If a numeric item is formatted into a smaller field left-most digits (most significant digits) are truncated as needed to fit the data into the specified fields. When using the 'F' and '-' format codes, the leading character of the data field being printed is always reserved for a leading negative sign. For alphanumeric fields, the source data field is left-justified into the area specified with truncation or blank fill occurring on the right side of the field.

'd' identifies the number of decimal places in a data item. The 'd' ranges from 0, for no decimal places, to 6, the maximum decimal range on CENSPAC. If decimal places are not needed, such as with alphabetic items or whole numbers, the 'd' must be specified by a .0. Except when specifying binary data items in an output file, the value of 'd' must always be less than the value of 'w'.

The spacing of data items on a line of a report can be done with the X formatting code, which provides blank fill, with the '@' formatting code which provides the 'tab' function, or with a literal that contains spaces, i.e. '   '. Users can also space data items by specifying a greater width, 'w', than is required for a data item. If the 't' identifier indicates zero suppression, then leading blanks will appear between data items. If 't' is not a zero suppressor, zeros will appear between items. Specifying extra space with the 'w' is especially important when a repetition factor, 'n', is used and the user wants to space identically formatted items across a page. The maximum value of 'w', width, for numeric data items is 18.

dataname: This operand is optional. 'Dataname' refers to existing names in the data dictionary or user-defined names, and it indicates that the item is to be placed in the output report or file. The dataname may be subscripted when it refers to an array, allowing specific cells from an array or whole dimensions of a multidimensional table to be output.

dataname THRU datanamel: A series of data items, defined in the data dictionary can be specified using the keyword 'THRU'. The data items must be in consecutive sequence in the data dictionary with dataname referring to the first item to be output and datanamel referring to the last. Fillers (data items in the data dictionary whose first six characters are 'FILLER') are excluded from the items. Therefore, when fillers are desired on the output, they cannot be included as part of the THRU range, but must be explicitly referenced.

The 'THRU' function is not available in May or August 1980 versions of CENSPAC.

literal: This operand is optional. Alphanumeric literals may be used. Alphanumeric literals may contain any character, except quotes and the entire literal must be enclosed by quotes. Alphanumeric literals in the ITEM command may be up to 78 characters long.

Example

```
FILEIN STF1
REPORT
ITEM STATE X4.0 COUNTY @14.0 2F10.0 TOTPOP TABAGES (1,1)
ITEM 'HOUSING UNITS' 2F10.0 TOTHSG TABAGES (2,2)
```

This example formats a report of two lines. One line of output corresponds to each ITEM command. The first ITEM command lists the 2 digit state code in the first 2 positions of the print line, followed by 4 spaces. Next is the 3 character county code. The next specification skips or 'tabs' to column 14, the data requested by the parameters that follow will begin in column 14. This is followed by the format code for 2 numeric data items, both with 10 digits and no decimals. The data items are identified after the format specification. The first of these items is 'TOTPOP', the second item is the cell in the first categories of the first and second stratifiers of the table 'TABAGES'.

The second ITEM command formats a literal in positions 1-13 of the second line of output, followed by 2 numeric items, TOTHSG, and cell (2,2) of the array 'TABAGES', the second categories of the first and second stratifiers of the table.

The ITEM command and arrays

The ITEM command can be used to retrieve tabular data for output onto a file or a printed report. Entire tables can be accessed by referencing the table name. For example, a reference to 'TAB24' is a request to extract table 24 data from the input tape and have all data cells associated with table 24 placed into the report or output file. Individual cells within tables can be selected also. The user does this by subscripting the tables referenced in an ITEM command. Numeric operators as well as the two special array operators, the asterisk (*) and C, may be used as subscripts. A subscript is always enclosed by parentheses and immediately follows the table name, i.e. TAB1 (*), or TAB1 (C) or TAB1 (1). A table name reference with no subscripts is equivalent to using asterisk(s) subscripts. The number of subscripts that follow a table name must equal the number of stratifiers in the table If TAB2 is a three dimensional table, it must be subscripted TAB2 (1,1,1). The array operators and numeric subscripts control the printing of table cells. Numeric subscripts retrieve a single cell from a table, while the array operators retrieve every cell within one stratifier. The left-most subscript identifies the first stratifier. The right-most subscript identifies the last stratifier.

When an asterisk (*) is used as a subscript, all data cells from a stratifier of a table are placed in a line across a page, one cell after another. The column array subscript, C, lists all cells from a stratifier down the page of a report in a column. The C operates in the same way as the asterisk -- only the direction of print changes. The asterisk (*) subscript can be used to enter tables on a file or a report. The C subscript may be used to place arrays only on reports.

For example, say TAB9 is a one dimensional table containing 5 cells -- the count of persons by race. A reference to TAB9 (C) would result in the 5 data items (race categories) being printed on 5 lines, in one column, on a report. A reference to TAB9 (*) would result in the 5 data items being printed across the page on one line of a report.

When retrieving multidimensional tables for a report, the asterisk, the C, and numeric subscripts can be used in combination to print tables.

If TAB12 is a two dimensional array containing 20 cells, the count of persons by race (with five categories) and age (with four age groups), a reference to TAB12 (*, C) would result in a 5 line report (5 racial categories) with 4 columns of data (four age groups). A reference to TAB12 (C *) would result in a 4 line report (age) with 5 columns of data (race).

If TAB15 is a three dimensional array containing 40 data cells, race by age (as in TAB12 above), by sex, a reference to TAB15 (*, *, *) would attempt to print 40 cells of data (5 x 4 x 2) on one line. A reference to TAB15 (C, C, C) would print 40 lines of data in a single column. A reference to TAB15 (1, 1, 1) would print only one cell, the first category of race for the first category of age for the first category of sex. A reference to TAB15 (1, C, *) would print data for the first category of race, for all categories of age in 4 lines down a page, with the 2 categories of sex on each line. See Examples 1 and 2 for additional examples of printing tables.

The ITEM command and labeling reports

When producing printed reports, users can retrieve table labels from the data dictionary and have them printed on a report. The labels available are shown in the summary tape file technical documentation. To retrieve labels, the user must request them in an ITEM command using the formatting type code field. The format code for the data dictionary label is followed by the name of the table (TABxxxxx) whose label is to be used.

Labels may be referenced with an ITEM command at whatever point the user wants them to be inserted in a report. The position of the label reference in a CENSPAC run determines the line of a report on which the label will be printed. See Example 2, Chapter III for more on labeling.

The format codes for data dictionary labels are:

H: table heading label
U: universe description label
S: table stratifier label
C: table category label

For example:

```
        ITEM H40.0 TAB8
```

retrieves the heading label (H) for Table 8 and places 40 characters of the label at the left margin of the print line.

The decimal code has a special meaning in formatting heading labels. It specifies whether continuation lines are to be printed. The reference 'H40.041 TAB8' would retrieve 40 characters of the heading label starting at location 41 on the label fields. Since labels are 40 characters long, this request retrieves the first continuation label. As many as 7 continuation labels may be retrieved. 'H40.281' would retrieve the seventh continuation label, or eighth label line. If no continuation labels are available, a request for them is filled with blanks. The values following the decimal point must be either one character or three characters in length, e.g. .1 or .001. A '.0' and '.1' are equivalent for specifying the beginning label position.

Retrieving heading or universe labels is easy because there is only one heading or universe label per table. But there are often several stratifier labels (S) and category labels (C) per table. For stratifier and category labels, the reference must be subscripted so that the particular label or labels are retrieved. For example, a two dimensional table, TAB10, sex by age, will have two stratifier labels. A reference to S40.0 TAB10 (1) will retrieve the entire 40 characters of the first stratifier label, 'SEX'. Similarly, S40.0 TAB10 (2) will retrieve the second stratifier label 'AGE'.

Category labels identify the categories within the stratifiers of a table. For example, the category labels for the stratifier 'AGE' in TAB10 would identify the various age categories. Category labels cannot be retrieved individually, and must be referenced with an asterisk (*) or C. When category labels are referenced, CENSPAC retrieves all category labels within a particular stratifier. Category labels can be retrieved by placing an asterisk (*) or a C in the subscript position of a particular dimension of a table and using numeric subscripts to void references to the other dimensions. The numeric subscripts act as place holders to identify the position of the array operator. When retrieving category labels, an asterisk (*) or a C must appear in at least one dimension of the array. For example, to retrieve the category labels for age, the second dimension in the two dimensional array, TAB10, the following format would appear in an ITEM command:

```
ITEM  C40.0 TAB10 (1, *)
```

This command would retrieve all category labels for age and print them in a line.

```
ITEM C40.0 TAB10 (1, C)
```

This command would retrieve all category labels for age and print them in a column. If TAB10 is a 2 by 26 table, with 2 sex categories and 26 age categories, the above example would retrieve 26 lines of labels, one for each age category.

```
ITEM C20.0 TAB10 (C, C)
```

When more than one array operator appears in a subscript, category labels will be printed for the stratifier referenced by the right-most operator. The labels for that stratifier will be repeated for each category of the other stratifiers. The above example would print 52 lines of labels, with the 26 categories of age listed for each category of sex. No labels for sex would be printed.

Some labels in the data dictionary are not accessible to CENSPAC. These include, GROUP, DATA, FOOTNOTE, Value, and Comment. (See Chapter IV, the CENSPAC Documentor for more information on these labels.)

REPORT

The REPORT command produces a printed report. At least one REPORT or FILEOUT command must appear in a CENSPAC run.

Every REPORT command must be followed by one or more ITEM commands. All ITEM commands that follow a REPORT command, up to the next REPORT or FILEOUT command, will be part of the current report.

Syntax: REPORT PAGESIZE (p)

> pagesize: This operand is optional. It specifies the maximum number of lines per page of output. If PAGESIZE is not specified, it is set at 58 lines. The value of PAGESIZE is unique for each REPORT. The user can control the number of lines printed on each page of a report.

Reports are formatted by the user with the HEAD and ITEM commands that follow the REPORT command. The HEAD and output ITEM command identifies the data items to be printed and their format. Both HEAD and ITEM commands can be used to label reports, retrieve labels from the data dictionary, or identify user-defined labels. Multiple HEAD and ITEM commands may follow a REPORT command. Each HEAD or ITEM command identifies a separate line of output.

By using IF and COMPUTE commands with a REPORT command, a user can conditionally print particular data items. A user can also request that some data items be placed on a new file while others are placed on a report.

Example

```
FILEIN STF1
REPORT PAGESIZE (40)
ITEM TAB1
```

This command instructs CENSPAC to produce a report with 40 lines on each page.

Section 4: CENSPAC Processing Commands

CENSPAC processing commands are necessary only when a user wants to manipulate data before they are placed on a new file or report. Processing commands can be inserted in a CENSPAC run wherever they are needed following the FILEIN command allowing for the conditional manipulation of data items.

The processing commands are ARRAY, COMPUTE, EDIT, HELP, IF, SORT, SOURCE and TOTAL. They are described here in alphabetical order.

ARRAY

The ARRAY command defines a temporary working table in a CENSPAC run. In most runs the ARRAY command is not needed because the COMPUTE command defines the name and dimensions of a working table. The ARRAY command needs to be used only in special circumstances when a COMPUTE command with array operators cannot establish the dimensions.

Multiple tables may be defined in a single ARRAY command. Each reference to a table must include the table name and its dimensions in subscripts. The ARRAY command that defines a table must precede commands that reference the table. If a table is referenced in a COMPUTE command that has not otherwise been defined, however, a default definition will be established. The default characteristics are those of the first table referenced on the right side of the equal sign (=) in the COMPUTE command.

Syntax: ARRAY dataname (n n n ...)

dataname: This operand is required. The dataname identifies the table that is being established. The first three characters must be 'TAB', followed by up to 5 alphanumeric characters.

(n n n ...): This operand is required. The numeric values enclosed in parentheses identify the stratifiers of the table and the number of categories in each dimension. Up to 6 dimensions are allowed.

Example

```
ARRAY TABAGE1 (3, 100)
```

This command creates a working table identified as 'TABAGE1'. The table has two dimensions, with 3 categories in the first stratifier and 100 categories in the second stratifier. A total of 300 cells will be created by this ARRAY command.

COMPUTE

The COMPUTE command manipulates numeric data items with the arithmetic operators, creating new arithmetic variables. COMPUTE also works with character data so that character values or strings can be held for later reference. Datanames can refer to either numeric or alphanumeric data, although mixed usage in a single command is not allowed.

Syntax: COMPUTE dataname = mathematical expression, dataname, or literal data

dataname: This operand is required. The dataname may be user-defined or it can be an item defined in the data dictionary. If the result variable in a COMPUTE command has not been previously defined a default definition will be established. The default characteristics, including size of the item and number of cells, of the item or table will be those of the first dataname referenced on the right side of the equal (=) sign in the COMPUTE command.

mathematical expression: The arithmetic operators +, -, *, and / may be used with the COMPUTE command. Parentheses may be used as needed. All computations are done without rounding. The least significant digits that cannot be accommodated in the result field are dropped.

Numeric data items in CENSPAC are signed, computational fields with 6 implied decimal places and 12 significant digits to the left of the decimal point. When a numeric value exceeds this length, overflow occurs, and the left-most characters, the most significant digits, are lost. No error messages are issued; the occurrence of overflow is ignored by the CENSPAC system.

If a COMPUTE command involves division by zero, the CENSPAC program will terminate. Users must insure that any COMPUTE statements with the divide operator (/) will not result in division by zero. Note: Most COBOL compilers will suppress division by zero, setting the result field to zero and allowing program execution to continue.

The COMPUTE statement does not provide for rounding but rounding can be easily provided for by 'half-adjusting'. Rounding using half-adjusting involves adding a value equivalent to .5 of the least significant digit that is to be retained for printing. For example, to round to the nearest whole number a COMPUTE statement would add 0.5 to the results. When the value is printed using the ITEM command with all decimal places truncated, the value will be correct.
The statement

```
COMPUTE PERCENT = MALES/TOTPOP + 0.5
```

will calculate the percentage of males and half adjust so that when the value of PERCENT is printed as a whole number it will be properly rounded. (An unrounded initial result of .0 thru .4 with .5 added will leave the units position unchanged, an unrounded result of .5 thru .9 with .5 added will increase the units position by one. The effect of this is to round to the nearest whole number).

Similarly the statement

```
    COMPUTE PERCENT = MALES/TOTPOP + 0.05
```

will half-adjust so that the resulting value carried to one decimal place is properly rounded.

The COMPUTE command can operate on a single cell or all cells in one or more stratifiers of a table. The asterisk (*) indicates that all data items in a stratifier are to be processed by the COMPUTE. When referencing entire tables with an asterisk, the number of stratifiers and categories within stratifiers for each of the tables referenced in the COMPUTE command must be identical.

For example:

```
    COMPUTE TABNEW (*) = TAB1 (*) + TAB2 (*)
```

Here TABNEW, TAB1 and TAB2 must have the same number of stratifiers (in this case) and categories within stratifiers for the COMPUTE command to work properly. References to numeric literals, data items, or single cells within tables, imply the use of that item repetitively on every cell in a table when using the array operator (*).

For example:

```
    COMPUTE TABNEW (*) = TAB003 (*) * 3.2
```

If TABNEW and TAB003 are tables with 10 cells in one dimension, this command directs each cell in TAB003 to be multiplied by 3.2 and to place the results in the corresponding cells of TABNEW.

dataname or literal: Alphanumeric data values or literal strings can be assigned as values of new alphanumeric character items.

Also see the SORT command for a discussion of the '#' CENSPAC variables.

Note: The use of alphanumeric data and '#' CENSPAC variables are not available in May or August 1980 versions of CENSPAC.

A special operand, SUM, is available with the COMPUTE command for use with tables. This function allows the user to calculate a total by adding all cells from one or more dimensions of a table. The format of the COMPUTE command with the SUM function is:

```
COMPUTE NEWTOTAL = SUM (TABxxxxx (n, n))
```

where at least one subscript is the asterisk operator. The table TABxxxxx must be referenced with as many subscripts as it has stratifiers, and it must contain an asterisk in each stratifier that is to be included in the sum. Numeric subscripts may be used with the asterisk in order to include only specific categories within stratifiers of a table in a SUM. When using SUM, the result data name should first be computed to zero before the SUM operation. It is not automatically set to zero for each new input record by CENSPAC and if not computed to zero, an aggregation of all records processed will be derived. See Example 4 in Chapter III for a CENSPAC run using the SUM function. No more than 999 COMPUTE statements can be present in a CENSPAC run.

Examples

```
FILEIN STF1
COMPUTE NEWTOTAL = 0
COMPUTE NEWTOTAL = SUM (TAB10 (1, *)).
REPORT
ITEM F10.0 NEWTOTAL
```

The user sets the new variable NEWTOTAL equal to zero before summing across one dimension of TAB10. The variable NEWTOTAL is then listed in a report using the F10.0 format specification.

```
COMPUTE FACTOR = (FACTOR + 2.4) * 100
```

The data item, FACTOR, is assigned the value of (FACTOR + 2.4) multiplied by 100.

```
COMPUTE AGE1 = TABAGES (1) + TABAGES (2)
```

The data item AGE1 is assigned a value equal to the sum of the values of cell 1 and cell 2 of the one dimensional table TABAGES.

```
COMPUTE TABAGES (*) = TABAGES (*) * 0.15
```

The values of all cells in the one dimensional table TABAGES1 are assigned the values of the corresponding cells in the one dimensional table TABAGES multiplied by 0.15

```
COMPUTE TOTMALE = 0
COMPUTE TOTMALE = SUM (TAB14 (1, *))
```

The data item TOTMALE is assigned the value of the sum of all cells subordinate to first category of the first stratifier of TAB14. If TAB14 is a sex (2 categories) by marital status (5 categories) distribution, SUM would calculate a total for all males included in the 5 marital status stratifiers.

```
IF SUMRYLVL = '11' COMPUTE CTYNAME = AREANAME;
IF SUMRYLVL = '14'
REPORT
ITEM CTYNAME TRACT TAB5 (*)
```

AREANAME from county level records (SUMRYLVL = '11') is assigned to CYTNAME. CTYNAME remains unchanged and available for reference until another county record is processed. Here it is printed in a tract report (SUMRYLVL = '14').

EDIT

The EDIT command causes CENSPAC to check a set of commands for validity and syntax without actually accessing the data file. The EDIT command may appear anywhere after the FILEIN command and will edit the entire command sequence of the run.

After all commands in a run have been read, appropriate error messages are printed and processing ends. Error message 105 will be printed as a result of the EDIT command.

Syntax: EDIT

The EDIT command has no operands.

HELP

The HELP command provides information on the format, function, and rules of CENSPAC commands. The HELP command is not available in May or August 1980 versions of CENSPAC.

Syntax: HELP command-name options

command-name: This operand is required. The command name identifies the CENSPAC command for which HELP is being requested. If a semicolon (;) is used in place of a command name, the user will receive information about the HELP command.

options: This operand is optional. The options allow users to request information about particular features. The options apply to general command characteristics (such as SYNTAX or FUNCTION), or to parameters of a particular command (such as BLKSIZE for the FILEIN command). If no options are entered, all the HELP information about a command will be printed.

The options available for particular commands are as follows:

Command	FUNCTION	SYNTAX	EXPRESSION	FORMAT	DEVICE	BLKSIZE	RECSIZE
ARRAY	X	X					
COMPUTE	X	X	X				
EDIT	X	X					
ENDM	X	X					
FILEIN	X	X			X	X	X
FILEOUT	X	X			X	X	X
HEAD	X	X		X			
HELP	X	X					
IF	X	X	X				
ITEM	X	X		X			
LABEL	X	X		X			
MATCH	X	X					
REPORT	X	X					
SORT	X	X					
TOTAL	X	X					

Examples

```
HELP ;
```

In this example, the user has requested information about the HELP command.

```
HELP COMPUTE SYNTAX
```

In this example, the user has requested information about the syntax of the COMPUTE command.

IF

The IF command selects records for processing and allows for conditional operations on data items.

Syntax: IF expression THEN command-list terminator

expression: This operand is required. It identifies the conditions that must be met before the commands are carried out. Logical and comparative operators (NOT, AND, OR, greater than, less than, equal to), as well as data names, numeric literals, and parentheses may be used in an IF expression. Alphanumeric data items can be tested only against other alphanumeric items or alphabetic literals. Numeric data items may be tested only against other numeric items or against numeric literals. For a complete list of CENSPAC operators and the rules regarding their use, see Chapter II, Section 1.

With the IF command, users may omit or select certain records for processing. For example, when working with the Census Summary Tape Files, users may want data tables for particular geographic areas printed on a report. This selection is done using the variable SUMRYLVL in the IF expression, followed by the geographic code(s) for the areas to be selected. The technical documentation accompanying the Census Summary Tape Files identify the SUMRYLVL codes.

THEN: This operand is optional. The keyword THEN is used only for readability and does not affect processing.

command-list: This operand is required. The commands, such as COMPUTE or ITEM, specify what operations are to be performed based on the IF expression. Several keywords are also available for use in IF commands. The INCLUDE keyword is used to include certain records in further processing; all records failing the test to be included are dropped from further processing. The EXCLUDE keyword is used to omit certain records from further processing. The STOP keyword stops further processing of the input data file by causing an immediate end-of-file condition.

terminator: The semicolon(;) ends the IF commands. If no semicolon appears, the IF is terminated at the end of the control card set, or when the keywords STOP, EXCLUDE, INCLUDE appear or when the commands FILEIN, ENDM, SORT, or TOTAL appear. When the INCLUDE keyword is used, no other CENSPAC commands can be included between the IF expression and the keyword INCLUDE.

Multiple IF commands may appear in a single run, but each IF statement must be terminated before another IF statement is allowed. For example:

```
IF SUMRYLVL = '14' INCLUDE
COMPUTE NEWVAR = TAB10 (1) + TAB10 (2)
IF NEWVAR < 1000
REPORT
ITEM COUNTY TRACT NEWVAR;
```

The first IF command is terminated with the keyword, INCLUDE, while the second IF statement terminator is the semicolon (;).

No more than 999 IF statements can be present in a CENSPAC run.

The logical connectors AND and OR can be used as well as parentheses when needed. The expression is evaluated left to right with AND taking precedence over OR. Expressions within parentheses are always evaluated first. For example, note the following expression:

```
IF A1 > B1
   AND A1 < C1
   OR A1 > D1
      COMPUTE X1 = 5;
```

Based on the priority rules, the expression would be evaluated as though parentheses were included as follows:

```
IF (A1 > B1
   AND A1 < C1)
   OR A1 > D1
      COMPUTE X1 = 5;
```

The expression would be considered to be true if A1 is greater than B1 and A1 is less than C1, or if A1 is greater than D1 regardless of B1 and C1. The expression could be evaluated differently by changing the position of the parentheses.

```
IF A1 > B1
   AND (A1 < C1
   OR A1 > D1)
      COMPUTE X1 = 5;
```

The expression will be true if A1 is greater than B1 and either of the two remaining conditions is true, that is, A1 is less than C1 or A1 is greater than D1.

When using more than one INCLUDE and/or EXCLUDE statements in a CENSPAC run, care must be taken to avoid excluding all records from a run. Each INCLUDE or EXCLUDE implies deleting records. A sequence of IF commands each with the INCLUDE operand will successively delete more records. For example

```
IF COUNTY = '001' INCLUDE
```

will allow only records belonging to county '001' to be accepted for further processing. All records that do not have a county code of '001' will be excluded. A second IF command,

```
IF COUNTY = '003' INCLUDE
```

following the previous IF, will cause the remaining records to be deleted. (Only '001' records passed the first test, only '003' records will pass the second test.

If data for both counties are to be processed, the command should be

```
IF COUNTY = '001' OR COUNTY = '003' INCLUDE
```

Examples

```
IF SUMRYLVL = '11' INCLUDE
```

When using Census Summary Tape File 1A, this IF statement would include only county records (code 11) in further processing.

```
IF TOTPOP = 10000 COMPUTE FACTOR = FACTOR + 1;
```

For each record that is processed, when the value of TOTPOP is greater than 10000, the value of FACTOR is increased by one.

```
IF COUNTY NOT = '005' EXCLUDE
```

Only those records with a value of '005' for COUNTY (which is the county code) will be processed.

```
IF COUNTY = '009' STOP
```

This IF statement tests for a value greater than '009' in the data item named 'COUNTY'. If the condition is met, end-of-file is set and the remaining CENSPAC commands are executed as though a normal end-of-file was reached.

LABEL

The LABEL command is an optional command which assigns labels to either single data items or tables. Labels can be specified for data items which are newly computed, input data items without a data dictionary, or input data items for which data dictionary labels are to be overriden. Only one LABEL command may appear for each data item or table. The LABEL command is not available in May or August 1980 versions of CENSPAC.

Syntax:

```
LABEL  dataname1  FROM  dataname2
LABEL dataname1 label-list
```

dataname1: This is the name of the item or table being labeled. It may refer to any previously defined item. Table names cannot be subscripted.

dataname2: When labels are to be assigned from an already existing set of labels, dataname2 identifies the item or table from which labels are to be taken. If labels from one table are used to label a second table, both tables must have the same number of dimensions and categories. Table names are not subscripted when referenced in the LABEL command.

label-list: When new labels are to be assigned a label list is provided by the user. Entries in the label list contain two fields - a label ID followed by the label text enclosed in single quotes. The label ID identifies the type of label and, if necessary, the stratifier number and category number. Valid label-IDs are the following:

H	Heading
U	Universe
S	Stratifier
C	Table Category
G	Group
D	Data Item
V	Data Item Category

Stratifier and category label IDs for tables are followed by the stratifier number in parentheses and category label number. Category labels or value labels for independent data items are identified with a 'V' followed by the value code. No stratifier number is necessary. Category codes referring to alphanumeric data must be enclosed in single quotes.

The label text is provided in alphanumeric strings of up to 40 characters enclosed in single quotes. For continuation labels, a colon precedes the first quote enclosing the label.

Label information in the label list is free form. More than one label may appear on a single input line. There are no column constraints, although literals may not be split from one line to the next. Any sequence of label types and stratifiers can occur. The only limitation in sequencing is the ascending order of category codes within a table staratifier or data item.

In labeling a new item or table, CENSPAC attempts to assign all possible types of labels (D and V for data items; H, U, S, and C for tables). When assigning labels from an already existing set, all labels are transferred. When specifying labels as literals, all types of labels can be provided. Labels not provided will default to blanks, except category labels for numeric codes. These labels will default to the category number as the label.

<u>Example</u>

```
LABEL  TABVALUE  C  (1)  1  'Less than $25,000'
                         2  '$25,000 to $49,999'
                         3  '$50,000 to $74,999'
                         4  '$75,000 to $100,000'
                         5  '$1000,000 or more'
                 H  'HOUSING VALUE FOR'
                  : 'CONDOMINIUM UNITS'
```

The table, TABVALUE, will have category labels assigned to the first stratifier for each of its five categories. A heading label is also specified. The universe and stratifier labels are not given and consequently, will be blank.

```
LABEL  TABNEW  FROM  TAB5
```

The heading, universe, stratifier, and category labels from Table 5 will also be the labels for TABNEW.

```
LABEL  GEOLEVEL  D  'Level of Geography'
                 V  '01'  'State'
                    '02'  'County'
                    '03'  'Place'
```

This alphanumeric data item, GEOLEVEL, has each two digit code identified with its appropriate label.

SORT

The SORT command resequences data in ascending or descending order.

Syntax: SORT dataname(s) dataname1(s)

dataname: This operand is required. One or more datanames, up to 8, can be used to identify the keys to be used in sorting. The first dataname will be used as the major sort key with the second and subsequent datanames being additional intermediate and minor keys. Dataname must be a non-subscripted dataname, that is, references to tables are not allowed. If a table value is desired as a sort key, the table item can be computed to an non-subscripted data item and the equivalent non-subscripted data item can then be used as the sort key.

s: This operand is optional. It specifies an ascending, A, or descending, D, order for SORT. If this is not specified, CENSPAC assumes ascending order.

Example

```
SORT COUNTY TOTPOP D
```

This SORT command reorders the data file by county and total population. County is the major key, and total population is the minor key. Data for counties will be sorted in ascending sequence. The records within the counties (places, tracts, etc.) will be sorted in descending sequence by population size.

When processing a SORT, all commands that precede the SORT command are processed first. CENSPAC then builds an intermediate work file consisting of all data items identified in the commands following the SORT command. The specified data are sorted, and then the sorted data are processed by the commands that follow the SORT command. (See flowchart on page 185.)

When a SORT command is used (or if the SORT option of the MATCH command is used), the host computer system SORT is invoked to do the sorting. The space requirements for the sort will depend upon the individual installation. The space allocation provided in the standard CENSPAC runstream of JCL usually is large enough for small and intermediate sized sort applications, so that the user will not have to calculate space requirements and a run will not be terminated due to lack of space. For large sorts, however, it is necessary to calculate space requirements.

In establishing the intermediate sort work records, CENSPAC selects only the data items needed for processing after the sort. Even though large input records are being processed, the actual sort records may be small.

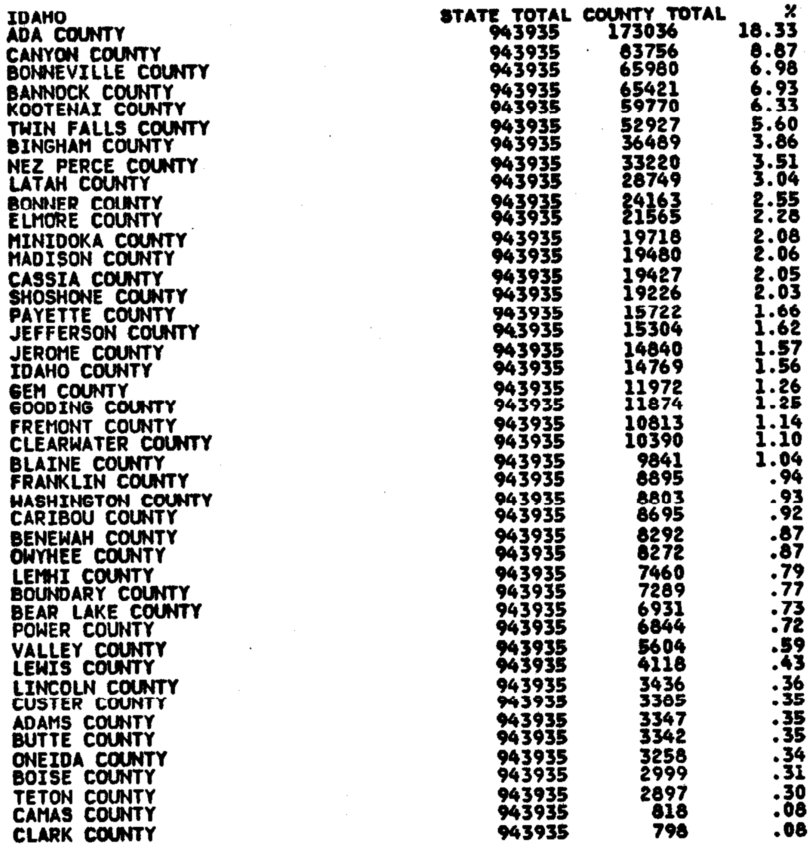
CENSPAC prints the sort record size in its end-of-command summary report, however, the record size calculated does not include the lengths of the sort key fields. Using the sort record size figure and an estimate of the volume of records to be sorted, the user can estimate the work space requirement. The method of calculating this estimate depends upon the system sort and can be found in the operating system's sort manual. The space requirements are likely to be at least 1.5 to 3 times the space needed simply to contain the amount of data to be sorted. For example, if the CENSPAC sort record size is 350 characters and there are 5000 records to be sorted (not necessarily the number of input data records read by CENSPAC) then a sort work space of 2.6 million to 5.3 million characters will be needed. This is approximately 140 to 275 tracks of 3350 equivalent space (19000 characters per track).

Data items may be excluded from the sort work records, yet held for later processing, by defining a '#' variable. Variables defined in a COMPUTE command with a '#' sign as the first character are recognized by CENSPAC to be excluded from sort procedures. The use of a '#' sign prefix on data names can be used to isolate variables from the effects of the SORT and TOTAL command. The '#' prefix on a variable does not establish a unique variable separate from the same variable name without the prefix i.e., '#TABX' and 'TABX' could both be referenced in a run and are equivalent names. If a '#' sign prefix is used on the first reference to a name than that variable is assumed to be a '#' variable for the entire run regardless of whether the '#' sign is used in later references. Conversely, if the first reference to a data name is without the '#' prefix, that data name will be treated as a non '#' variable for the remainder of the run regardless of whether the '#' prefix is later used or not. To avoid confusion on this point, the '#' should be consistently used or consistently omitted for all references to a particular variable in a run.

The use of the '#' prefix establishes the data name as a locally defined variable, that is a reference to '#TAB1' defines a working variable and will not result in accessing data from 'TAB1' of any input files. If TAB1 data is desired and the data is to be isolated from SORT and TOTAL then a new variable must be computed to hold the TAB1 data. For example:

```
FILEIN STF1
IF SUMRYLVL = '04'
   COMPUTE #STNAME = AREANAME
   COMPUTE #TABX1 (1) = TAB1 (1);
IF SUMRYLVL = '11' INCLUDE
COMPUTE CTYPCT = TAB1 (1) / #TABX1 (1) * 100
SORT CTYPCT D
REPORT
HEAD #STNAME 'STATE TOTAL   COUNTY TOTAL %'
ITEM AREANAME 2F10.0 #TABX1 (1) TAB1 (1) F10.2 CTYPCT
```

This run uses '#' variables to hold the state name and total population count for the state so these items can be available for use with the county records. IF SUMRYLVL = '04' tests for a state level summary and if the current record is a state level record, the 'AREANAME' and 'TAB1 (1)' will be saved in '#STNAME' and '#TABX1' respectively. The following 'IF' test selects only county records for further processing. The COMPUTE command computes the percentage population of the county as compared to the state total. A SORT command follows which sorts the county data into descending sequence based on population size within the state. This is followed by a REPORT, HEAD and ITEM command that prints the state name in the heading and prints the county name, state and county population totals, and the county percentage figures. The use of '#' variables is not available in May or August 1980 versions of CENSPAC. The printed output of this run is shown below:



```
IDAHO                                  STATE TOTAL COUNTY TOTAL     %
ADA COUNTY                                943935     173036     18.33
CANYON COUNTY                             943935      83756      8.87
BONNEVILLE COUNTY                         943935      65980      6.98
BANNOCK COUNTY                            943935      65421      6.93
KOOTENAI COUNTY                           943935      59770      6.33
TWIN FALLS COUNTY                         943935      52927      5.60
BINGHAM COUNTY                            943935      36489      3.86
NEZ PERCE COUNTY                          943935      33220      3.51
LATAH COUNTY                              943935      28749      3.04
BONNER COUNTY                             943935      24163      2.55
ELMORE COUNTY                             943935      21565      2.28
MINIDOKA COUNTY                           943935      19718      2.08
MADISON COUNTY                            943935      19480      2.06
CASSIA COUNTY                             943935      19427      2.05
SHOSHONE COUNTY                           943935      19226      2.03
PAYETTE COUNTY                            943935      15722      1.66
JEFFERSON COUNTY                          943935      15304      1.62
JEROME COUNTY                             943935      14840      1.57
IDAHO COUNTY                              943935      14769      1.56
GEM COUNTY                                943935      11972      1.26
GOODING COUNTY                            943935      11874      1.25
FREMONT COUNTY                            943935      10813      1.14
CLEARWATER COUNTY                         943935      10390      1.10
BLAINE COUNTY                             943935       9841      1.04
FRANKLIN COUNTY                           943935       8895       .94
WASHINGTON COUNTY                         943935       8803       .93
CARIBOU COUNTY                            943935       8695       .92
BENEWAH COUNTY                            943935       8292       .87
OWYHEE COUNTY                             943935       8272       .87
LEMHI COUNTY                              943935       7460       .79
BOUNDARY COUNTY                           943935       7289       .77
BEAR LAKE COUNTY                          943935       6931       .73
POWER COUNTY                              943935       6844       .72
VALLEY COUNTY                             943935       5604       .59
LEWIS COUNTY                              943935       4118       .43
LINCOLN COUNTY                            943935       3436       .36
CUSTER COUNTY                             943935       3385       .35
ADAMS COUNTY                              943935       3347       .35
BUTTE COUNTY                              943935       3342       .35
ONEIDA COUNTY                             943935       3258       .34
BOISE COUNTY                              943935       2999       .31
TETON COUNTY                              943935       2897       .30
CAMAS COUNTY                              943935        818       .08
CLARK COUNTY                              943935        798       .08
```

SOURCE

The SOURCE command allows the user to add COBOL routines to a CENSPAC run in the form of source statements. A SOURCE command must appear at the beginning and at the end of the user-supplied COBOL statements. The second SOURCE command returns the system to the CENSPAC command mode.

Syntax: SOURCE

The SOURCE command has no operands.

The SOURCE command alerts CENSPAC that COBOL source statements follow. The SOURCE command may be used to solve problems that are beyond the capabilities of CENSPAC. SOURCE code can include one or more COBOL source statements that perform the desired functions, or a COBOL 'CALL' statement can be supplied which links CENSPAC to a user written program that performs the desired functions. The COBOL statements are not checked by CENSPAC for errors.

COBOL source statements that begin in column one of a card image are treated as paragraph names. Statements beginning after column one are treated as procedure division statements.

See appendix 2 for a description of naming conventions that must be followed for COBOL SOURCE statements.

Example

```
SOURCE
    IF QINA < 10
       ADD 1 to AOOPTR
       COMPUTE AOOXNEW = AOOXNEW + AOOTAB38 (AOOPTR)
    ELSE
       MOVE +78910 to AOOXNEW.
SOURCE
```

If the input record number is less than 10, the value 1 will be added to AOOPTR and the value of AOOXNEW will be computed. Otherwise, AOOXNEW will be assigned the value, 78910. The variables PTR, XNEW, and TAB38 are from the common work area and are prefixed according to naming conventions described in Appendix 2. The variable QINA is equivalent to $INA.

TOTAL

The TOTAL command aggregates data for smaller geographic areas into data for larger areas.

Syntax: TOTAL dataname1...dataname5

dataname: This operand is required. From one to five data names may be specified as control keys for the computation of totals. When the value of a key changes, the aggregated data are available for subsequent processing.

The TOTAL command operates by examining the values of the control keys (data names) listed on the command line. As long as a control key value remains constant, the TOTAL command will aggregate the values of the variables referenced in the remainder of the CENSPAC run (see below for a discussion of what variables may be referenced by TOTAL). When the value of a control key changes, CENSPAC creates a new record that contains the aggregated data for that key and sets $TOTAL (see below) to the appropriate value. For the TOTAL command to operate properly, the data set must be sorted by the control keys. If the data set is not sorted by the control keys, CENSPAC will produce incorrect aggregations, since a new total is provided each time a key value changes.

To determine what level of aggregation is available, the predefined system variable, $TOTAL, may be used with the TOTAL command. The values of $TOTAL range from 0 through 1 greater than the number of key fields specified in the TOTAL command. If the TOTAL command is:

```
TOTAL DN1 DN2 DN3 DN4 DN5
```

then the values of $TOTAL would be:

```
0 - detail (input) record is being processed
1 - rightmost key (DN5) has changed and an
    aggregation record at this lowest level
    is available
.
.
.
5 - leftmost key (DN1) aggregation record
    is complete
6 - aggregation record for the entire file
    is complete and a grand total for entire
    file is available
```

In the following example, with one key on the TOTAL command, $TOTAL will equal to 0, 1, or 2. When $TOTAL equals 0, it indicates that a detail or input record is being processed; when $TOTAL equals 1, the county level aggregation record is complete, and when $TOTAL equals 2, the aggregation record for the entire file is complete. All commands following the TOTAL command are processed for each record unless excluded previously. If only detail records are to be processed for some commands and only summaries for other commands, then $TOTAL must be used with the IF command to qualify the level of summary properly.

Example

```
FILEIN STF3
IF SUMRYLVL = '13 ' AND POP < 2500 INCLUDE
SORT COUNTY
TOTAL COUNTY
IF $TOTAL > 0
FILEOUT BLKSIZE 10
ITEM STATE COUNTY TAB001 TAB2
```

In this example, only records that represent places of less than 2500 population are included in the run. The records are totaled into summaries by county. When the county code changes, an output record containing the data items listed in the ITEM command, aggregated by county, is written.

CENSPAC establishes a working data set for the TOTAL command. This data set consists of all data items specified as keys in the TOTAL command and all data items referenced following the TOTAL command whose data type and data category are numeric. Numeric data items may be excluded from the TOTAL procedure, yet held for later processing, by defining a '#' variable. Variables defined in a COMPUTE command with a '#' sign as the first character of the data name are recognized by CENSPAC to be excluded from the TOTAL procedures.

Variables that are alphanumeric or code should only be referenced following the TOTAL command if they are also referenced as keys in the TOTAL command. If alphanumeric or code variables are referenced following the TOTAL command that are not also used as TOTAL keys and are not '#' variables, the data values obtained will be from the current input record rather than from the last record aggregated. References to code and alphanumeric data following the TOTAL command at the detail level are valid, however, for derived levels of totals such references are usually invalid. Also see the SORT command for a discussion of '#' variables and Chapter II, Section 1, "Limitations on the use of numeric and alphanumeric data" for more information on data type and data category. For data items that are defined by an ITEM command instead of the data dictionary, the category attribute is assumed to be numeric or alphanumeric, depending on the format code, i.e. F, (numeric), A (alphabetic) etc. The TOTAL command will aggregate those items that are numeric. The use of '#' variables is not available in May or August 1980 versions of CENSPAC.

An example of TOTAL command processing with a sample set of data records including numeric and code variables follows. This example processes tract level records and summarizes the data to county and state level totals.

```
TOTAL STATE COUNTY
REPORT
IF $TOTAL > 0
ITEM STATE COUNTY TRACT TAB1 (*)
```

Records in data file:				CENSPAC OUTPUT				
STATE	COUNTY	TRACT	TAB1	STATE	COUNTY	TRACT*	TAB1	$TOTAL VALUE
01	001	0041	0005		(no output)			0
01	001	0042	0010		(no output)			0
01	003	0021	0100		(no output)			0
	(county	001	break)	01	001	0021	15	1
02	001	0005	0053		(no output)			0
	(county	003	break)	01	003	0005	100	1
	(state	01	summary)	01	003	0005	115	2
(end-of-file)								
	(county	001	break)	02	001	0005	53	1
	(state	02	summary)	02	001	0005	53	2
	(grand total)			02	001	0005	168	3

*Note that the tract code (identified in the data dictionary as alphanumeric and code) listed does not pertain to the geographic summary being presented because it is not referenced in the TOTAL command and is a code item. Note also that the state and county codes, identified as alphanumeric and code and referenced as keys in the TOTAL command, are kept current by CENSPAC TOTAL processing.

Section 5: CENSPAC System Variables

The system variables allow users to test or set selected internal control fields. The data names for the system variables are prefixed by a dollar sign ($). All system variables are automatically set by CENSPAC and should be altered by the user only to achieve specific results. Unless otherwise, indicated, all variables are set to 0 at the beginning of a run. The predefined system variables are described here in alphabetical order.

$DATE: This variable labels reports with the current date. An 8 character field in the form mm/dd/yy is created. $DATE may be used only in HEAD commands and may be used only once in each report.

$EOFA: This variable is available to invoke special end-of-file processing logic. The value of $EOFA is Ø as the input file is being processed. It is set to 9 when end-of-file is reached and if the user sets the value of $EOFA to a non-zero value, the CENSPAC run will assume that end-of-file has been reached. If $EOFA is referenced in a run, the CENSPAC end-of-file processing logic is modified so that all CENSPAC commands are executed one more time at end-of-file even though no new input record is available. All input variable values are from the last input record read for this end-of-file processing. If $EOFA is not referenced in a run, processing terminates after completing processing with the last record. This special usage of $EOFA is not available when using MATCH commands. This $EOFA processing logic is useful for developing record counts or printing multiple units of geography across a page. In both of these applications, there is data available that has not been output when end-of-file has been reached and one extra command processing cycle is needed to have the data output. See example run 11 for an example using $EOFA.

The STOP function should not be used when using $EOFA processing. The STOP function will cause immediate end-of-file and will not allow the CENSPAC commands to be processed after end-of-file. The use of "COMPUTE $EOFA = 9" in place of STOP will achieve the same effect as STOP (simulate end-of-file condition) but will allow the CENSPAC commands to be processed one time after the end-of-file condition.

Note: While the variable $EOFA is available in all versions of CENSPAC, the special end-of-file processing logic discussed here is not available in May or August 1980 versions of CENSPAC.

$INA This variable provides the current input logical record count for the input file.

$INB: This variable provides the current input logical record count for File B or the second input file when using a MATCH command.

$LCTF001: OR $LCTF This variable provides the current record count for a file generated with the FILEOUT command. $LCTF001 is available for the first FILEOUT command; $LCTF002, $LCTF003, etc., are available for subsequent FILEOUT commands in a run. When the numeric portion of the name is excluded, CENSPAC assumes the record counter refers to the current FILEOUT being processed. References to the record counter cannot precede the FILEOUT command. The use of the abbreviated form $LCTF, is not available in May or August 1980 versions of CENSPAC.

$LCTR001: OR $LCTR This variable, which may be referenced only after a REPORT command, provides the current line count within a report page. $LCTRØØ1 is available for the first REPORT command; $LCTRØØ2, $LCTRØØ3, etc., are available for subsequent REPORT commands in a run. When the numeric portion of the data name is excluded, CENSPAC assumes the line counter refers of the current REPORT being processed. References to the line counter cannot precede the REPORT command. The use of the abbreviated form $LCTR, is not available in May or August 1980 versions of CENSPAC.

This variable allows the user to limit the data on each page of a report. By setting the value of $LCTR, page ejects can be produced wherever they are needed. The page size is set by the operand PAGESIZE in the REPORT command. As a report is printed, each line that is printed adds a value of 1 to the $LCTR variable. When the $LCTR value exceeds the PAGESIZE value, the page is ejected, and the headings are printed on a new page. The value of $LCTR is checked before each ITEM, or line, is printed. By setting the value of $LCTR001 higher than the PAGESIZE value, a page will be ejected. The following example shows how page ejects may be programmed.

```
IF NEWCNTY NOT = OLDCNTY
COMPUTE $LCTR = 100
COMPUTE NEWCNTY = OLDCNTY;
```

This sequence of CENSPAC commands would cause each county to begin on a new page.

$MATCH This variable indicates whether an attempted match between the current input records was successful during a MATCH run. The value of $MATCH is reset each time a match is attempted. The possible values are:

0 A match was successful. All variables from A and B records are available.

1 The key for the current A record is lower than for the current B record. The data for both the current A record and the current B record are available.

2 The key for the current B record is lower than for the current A record. The data for both the current A record and the current B record are available.

$PAGE: This variable numbers pages in reports. Only one $PAGE may be associated with each REPORT. If multiple REPORT commands are used, each REPORT may include a $PAGE. $PAGE may be used only in a HEAD command.

$PAGR001: This is a special $PAGE variable associated with the REPORT command. The use of $PAGE triggers the page numbering mechanism for a REPORT. A related variable, $PAGR001, allows the user to manipulate the initial value of the page counter, setting the starting value greater than 1. $PAGR001, $PAGR002, $PAGR003, etc., are available for successive REPORT commands in a run.

```
IF $INA = 1
COMPUTE $PAGR001  = 24;
```

This sequence of commands causes the first page in a report to be numbered 24.

$SORT: This variable provides the current sorted record count in a SORT.

$SUPxxx: This variable can be used to indicate suppression in 1970 census tables. The reserved word 'SUP' replaces the work 'TAB'. 'SUP' is followed by a numeric value which identifies the table. 'SUP1' would be the suppression flag for 'TAB1'. This variable is equal to 1 when suppression is detected in a table and equal to 0 when no suppression is detected. To indicate suppression on the 1980 Summary Tape Files, see example run 7 in Chapter III or the technical documentation for each 1980 Summary Tape File.

$TOTAL: This variable is used with the TOTAL command and allows the user to selectively process aggregation records. The value of $TOTAL ranges from 0 through 1 greater than the number of keys specified in the TOTAL command. $TOTAL is set to 0 when a detail record is being processed. The value of $TOTAL corresponds to the current level of TOTAL being processed. As the records specified by the TOTAL keys are aggregated, $TOTAL varies from a value of 1 through the value corresponding to the highest key in the TOTAL command. When $TOTAL equals one more that the specified number of keys, the aggregation record for the entire file is complete.

Chapter III

CENSPAC Sample Runs

This chapter presents 12 sample CENSPAC runs. The first few runs provide examples of very basic CENSPAC functions and could be used by someone attempting to learn CENSPAC. Later runs exemplify more complex CENSPAC applications and are intended to provide a framework from which to model CENSPAC applications to meet more complex requirements.

Run 1

Printing a Table

This run shows how to produce a basic report. It prints a two dimensional table, identifies the geographic area, and includes page headings. The structure of this run can be used to print any table with a minimal number of commands.

Commands:

```
FILEIN STF1
IF SUMRYLVL = '11' INCLUDE
REPORT
HEAD 'REPORT GENERATION - RUN 1' X10.0 'PAGE' $PAGE
HEAD 'TABLE 27'
ITEM
ITEM 'COUNTY: ' COUNTY
ITEM
ITEM TAB27 (*,C)
```

Printout:

```
REPORT GENERATION - RUN 1            PAGE      1
TABLE 27

COUNTY: 041

    36475     8121
     3213     1179
       59       22
      197       58
       48       23

COUNTY: 087

    54515    17810
     7564     3151
      118       38
      322      200
      107       54

COUNTY: 760

    48136    23018
    35524    20849
      135       87
      315      225
      159      108
```

FILEIN STF1

Summary Tape File 1 is the input file.

IF SUMRYLVL = '11' INCLUDE

Only records with a summary level code of 11 will be included in the processing. Code 11 identifies county-level records. All records with a summary level code other than '11' will be excluded from the remainder of this run.

REPORT

A printed report will be produced.

HEAD 'REPORT GENERATION - RUN 1' X10.0 'PAGE' $PAGE

The literal, 'REPORT GENERATION - RUN 1' will be printed as a heading at the top of each page. After skipping 10 spaces (X10.0), the word PAGE and the page number ($PAGE) will be printed.

HEAD 'TABLE 27'

A second heading will be printed containing the literal, 'TABLE 27'.

ITEM

A blank line will be printed.

ITEM 'COUNTY: ' COUNTY

A line will be printed beginning with the literal 'COUNTY: ', followed by the county code.

ITEM

A blank line will be printed.

ITEM TAB27 (*,C)

Table 27 will be printed. The values for the first stratifier of Table 27, tenure, will be printed across the page (*), and the categories of the second stratifier, race, will be printed in columns (C).

Run 2

Data Dictionary Labelling and Page Ejection

Data from Table 27 is displayed with data dictionary labelling. Data for each geographic area is printed on a separate page in this example, with the page heading identifying the geography. Although this run illustrates printing of one table per geographic area, the commands could be expanded with additional ITEM commands to print more tables below Table 27 or with additional tables printed to the right of Table 27 by adding parameters into the existing ITEM commands.

Commands:

```
FILEIN STF1
IF SUMRYLVL NOT = '11' EXCLUDE.
REPORT
HEAD 'REPORT GENERATION - RUN 2' X20.0 'PAGE ' $PAGE
HEAD 'TABLE 27: ' H40.0 TAB27
HEAD  @21.0 'FOR ' AREANAME
HEAD   .HEAD U40.0 TAB27
ITEM    ITEM X48.0 2C16.0 TAB27 (*,1)
ITEM C40.0 TAB27 (1,C) 2F15.0 TAB27 (*,C)
ITEM    COMPUTE $LCTR001 = 100
```

```
REPORT GENERATION - RUN 2                           PAGE      1
TABLE 27: TENURE (2) BY RACE OF HOUSEHOLDER (5)
                 FOR CHESTERFIELD

UNIVERSE:  OCCUPIED HOUSING UNITS

                                                  TOTAL:            RENTER OCCUPIED:
WHITE                                               36475             8121
BLACK                                                3213             1179
AMERICAN INDIAN, ESKIMO, AND ALEUT                     59               22
ASIAN AND PACIFIC ISLANDER                            197               58
OTHER                                                  48               23

REPORT GENERATION - RUN 2                           PAGE      2
TABLE 27: TENURE (2) BY RACE OF HOUSEHOLDER (5)
                 FOR HENRICO

UNIVERSE:  OCCUPIED HOUSING UNITS

                                                  TOTAL:            RENTER OCCUPIED:
WHITE                                               54515            17810
BLACK                                                7564             3151
AMERICAN INDIAN, ESKIMO, AND ALEUT                    118               38
ASIAN AND PACIFIC ISLANDER                            322              200
OTHER                                                 107               54

REPORT GENERATION - RUN 2                           PAGE      3
TABLE 27: TENURE (2) BY RACE OF HOUSEHOLDER (5)
                 FOR RICHMOND CITY

UNIVERSE:  OCCUPIED HOUSING UNITS

                                                  TOTAL:            RENTER OCCUPIED:
WHITE                                               48136            23018
BLACK                                               35524            20849
AMERICAN INDIAN, ESKIMO, AND ALEUT                    135               87
ASIAN AND PACIFIC ISLANDER                            315              225
OTHER                                                 159              108
```

```
FILEIN STF1
```

Summary Tape File 1 is the input file.

```
IF SUMRYLVL NOT = '11' EXCLUDE
```

Only records with a summary level code of 11 will be included in the processing. Code 11 identifies county level records. All records that do not have an '11' in the SUMRYLVL field will be dropped from further processing.

```
REPORT
```

A printed report will be produced.

```
HEAD 'REPORT GENERATION - RUN 2' X20.0 'PAGE ' $PAGE
```

The literal, 'REPORT GENERATION - RUN 2', will be printed as a heading at the top of each page. After skipping 20 spaces, the word PAGE and the page number ($PAGE) will be printed.

```
HEAD 'TABLE 27:  'H40.0 TAB27
```

A second heading will be printed containing the literal 'TABLE 27:'. The data dictionary heading label for Table 27 will be printed next to the literal in a field of 40 characters (H40.0).

```
HEAD @21.0  'FOR ' AREANAME
```

Another heading line follows to identify the table, after skipping to position 21, with the literal 'FOR'. This is followed by the value of the item AREANAME.

```
HEAD HEAD U40.0 TAB27
```

A blank line will be printed (specified by the HEAD command with no operands following it). The next heading line will contain the universe label from Table 27 in a field of 40 characters (U40.0).

```
ITEM ITEM X48.0 2C16.0 TAB27 (*,1)
```

A blank line will be printed. On the next line, 48 spaces will be skipped, then 2 category labels will be printed for the first stratifier of Table 27 in fields of 16 characters. The asterisk (*) identifies the stratifier for which category labels will be printed. The first category label, 'Total', will be left-justified and padded with blanks in the field of 16 characters. The field of 16 allows for the length of the second category label, 'renter occupied'. The user cannot access individual labels within a stratifier, but must print all the labels using the linear (*) or columnar (C) operator.

```
ITEM C40.0 TAB27 (1,C) 2F15.0 TAB27 (*,C)
```

Category labels for the second stratifier of Table 27, (1,C), will be printed in fields of 40 characters. The labels will be printed in a column, one label per line. Two fields of 15 characters each will be printed across the page containing the values for the first stratifier of Table 27, (*,C). The values for the second stratifier will be listed in columns. All numeric data, which includes all table cells in STF1, are right-justified. To space the data in columns of comparable width to the column labels, fields of 15 characters are specified.

```
ITEM COMPUTE $LCTR001 = 100
```

A blank line will be printed (specified by the ITEM command with no operands following it). Every time the REPORT command is entered for each new record in this run, the value of the line counter ($LCTR001) is checked. When the value is greater than the report pagesize (here equal to the default value of 58 because PAGESIZE was not specified in the REPORT command), a new page is started. By computing the value of the line counter at the end of the command set to be greater than 58, ($LCTR001 = 100), the data that is printed for each record selected is printed on a separate page.

Page ejection could be more selective, such as beginning each new county on a new page and having data for the counties listed on the same page. A test to page eject when the county code changes might be as follows:

```
IF SUMRYLVL = '04' COMPUTE $LCTR = 100;
```

These CENSPAC commands assume a STF1A file sequence where the county summary is '04' and precedes detail records for the county. These commands (IF and COMPUTE) would precede the ITEM commands for the report, immediately following the REPORT and associated HEAD commands, if any.

Another method for selective page ejection is to test more directly for changes in the codes. For example

```
IF OLDCOUNTY NOT = COUNTY
   COMPUTE $LCTR = 100
   COMPUTE OLDCOUNTY = COUNTY;
```

This sequence of commands compares the value of the COUNTY data item from the current record against the last county code processed. If there has been a change in the codes, $LCTR will be set to 100 and, to allow this test to be performed correctly for the next record, the value of OLDCOUNTY will be set equal to the current county code. The IF test will not be true again until data for the following county is encountered. This sequence of commands would appear in a REPORT command sequence immediately following the REPORT and HEAD commands, and before any ITEM commands for the REPORT. For example

```
REPORT
HEAD 'PAGE EJECT EXAMPLE'
IF OLDCOUNTY NOT = COUNTY
   COMPUTE $LCTR = 100
   COMPUTE OLDCOUNTY = COUNTY;
ITEM 'BODY OF REPORT FOLLOWS'
ITEM COUNTY TAB1
(remainder of CENSPAC commands)
```

Run 3

Calculation of Category Sums

In order to reduce record size and still provide a large amount of data, the 1980 summary tape files do not total categories of a stratifier. The user may want to sum categories, displaying the sums in a table. This run illustrates the calculation of sums, with labeling and formatting.

Commands:

```
FILEIN STF1
IF SUMRYLVL = '11' INCLUDE
COMPUTE TABOWN (1,*) = TAB27 (1,*) - TAB27 (2,*)
COMPUTE TOTALHU = 0
COMPUTE TOTRENT = 0
COMPUTE TOTOWN = 0
COMPUTE TOTALHU = SUM(TAB27(1,*))
COMPUTE TOTRENT = SUM(TAB27(2,*))
COMPUTE TOTOWN = SUM(TABOWN(1,*))
REPORT
HEAD 'REPORT GENERATION - RUN 3' X20.0 'PAGE ' $PAGE
HEAD     HEAD X15.0 H40.0 TAB27
HEAD X15.0 U40.0 TAB27
ITEM ITEM 'COUNTY: ' AREANAME
ITEM @45.0 '           RENTER    OWNER'
ITEM @45.0 'TOTAL:     OCCUPIED: OCCUPIED:'
ITEM C40.0 TAB27 (1,C) 3F10.0 TAB27 (*,C) TABOWN (1,C)
ITEM @45.0 ' -----     -----     -----'
ITEM 'TOTAL' @41.0 3F10.0 TOTALHU TOTRENT TOTOWN
```

Printouts:

```
REPORT GENERATION - RUN 3                    PAGE      1

                         TENURE (2) BY RACE OF HOUSEHOLDER (5)
                         Universe:  Occupied Housing Units
COUNTY: CHESTERFIELD
                                                     RENTER    OWNER
                                           TOTAL:    OCCUPIED: OCCUPIED:
White                                      36475      8121     28354
Black                                       3213      1179      2034
American Indian, Eskimo, and Aleut            59        22        37
Asian and Pacific Islander                   197        58       139
Other                                         48        23        25
                                           -----     -----     -----
TOTAL                                      39992      9403     30589

COUNTY: HENRICO
                                                     RENTER    OWNER
                                           TOTAL:    OCCUPIED: OCCUPIED:
White                                      54515     17810     36705
Black                                       7564      3151      4413
American Indian, Eskimo, and Aleut           118        38        80
Asian and Pacific Islander                   322       200       122
Other                                        107        54        53
                                           -----     -----     -----
TOTAL                                      62626     21253     41373

COUNTY: RICHMOND CITY
                                                     RENTER    OWNER
                                           TOTAL:    OCCUPIED: OCCUPIED:
White                                      48136     23018     25118
Black                                      35524     20849     14675
American Indian, Eskimo, and Aleut           135        87        48
Asian and Pacific Islander                   315       225        90
Other                                        159       108        51
                                           -----     -----     -----
TOTAL                                      84269     44287     39982
```

```
FILEIN STF1
```

Summary Tape File 1 is the input file

```
IF SUMRYLVL = '11' INCLUDE
```

Only records with a summary level code of 11 will be included in the processing. Code 11 identifies county level summary records.

```
COMPUTE TABOWN (1,*) = TAB27 (1,*) - TAB27 (2,*)
```

The number of owners, by race, is calculated by subtracting renters by race (TAB27 (2,*)) from total householders by race (TAB27 (1,*)). The new values are entered in TABOWN (1,*), a user-defined array.

```
COMPUTE TOTALHU = 0
```

The value of total housing units is set to zero. Initialization is necessary whenever the SUM keyword is to be used. The value of a newly computed item is only set to zero at the beginning of a run; thus SUM would continue to add values across records unless it is set to zero for each record.

```
COMPUTE TOTRENT = 0
```

The value of total renter-occupied units is set to zero.

```
COMPUTE TOTOWN = 0
```

The value of total owner-occupied units is set to zero.

```
COMPUTE TOTALHU = SUM (TAB27 (1,*))
```

The total number of housing units is calculated by summing across all categories of the second stratifier, race.

```
COMPUTE TOTRENT = SUM (TAB27 (2,*))
```

The total number of renter-occupied housing units is calculated by summing across all categories of the second stratifier, race.

Run 3

```
COMPUTE TOTOWN = SUM (TABOWN (1,*))
```

The total number of owner-occupied housing units is calculated by summing across all categories of the second stratifier, race.

```
REPORT
```

A printed report will be produced.

```
HEAD 'REPORT GENERATION - RUN 3' X20.0 'PAGE' $PAGE
```

The literal, 'REPORT GENERATION - RUN 3', will be printed as a heading at the top of each page. After skipping 20 spaces, the word 'PAGE', and the page number ($PAGE) will be printed.

```
HEAD HEAD X15.0 H40.0 TAB27
```

A blank line will be printed (specified by the HEAD command followed by no operands). After skipping 15 spaces (X15.0), the next line will print the heading for Table 27 in a field of 40 characters (H40.0).

```
HEAD X15.0 U40.0 TAB27
```

After skipping 15 spaces (X15.0), a heading containing the universe label for Table 27 in a field of 40 characters (U40.0) will be printed.

```
ITEM ITEM 'COUNTY:  ' AREANAME
```

A blank line will be printed (specified by the ITEM command followed by no operands). The next line will contain the literal, 'COUNTY: ', followed by the value of the item, AREANAME.

```
ITEM @45.0 '          RENTER OWNER'
```

After 'tabbing' to position 45 spaces, a line will be printed with the literal, '   RENTER OWNER', as column labels. Because the data cells will be printed in 10 character fields (see below), the long "renter occupied" and "owner occupied" labels are split onto two print lines to keep the headings aligned over the columns of data. The 'tab' to position 45 is used to allow later ITEM commands to use the same 'tab' position to facilitate lining up data items.

```
ITEM @45.0 'TOTAL:  OCCUPIED:  OCCUPIED: '
```

After 'tabbing' to position 45, this literal provides the second line of column heading.

```
ITEM C40.0 TAB27 (1,C) 3F10.0 TAB27 (*,C) TABOWN (1,C)
```

Category labels for the second stratifier of Table 27 (1,C) will be printed in fields of 40 characters (C40.0). The labels will be printed in a column, one label per line. The values for Table 27 will be printed across the page in three 10 character fields. The first two fields will be the tenure categories of Table 27, with the race categories listed in a column, TAB27 (*,C). The last field will be the newly calculated owner counts, with race categories again appearing in a column, TABOWN (1,C).

```
ITEM @45.0 ' -----     -----     -----'
```

After tabbing to position 45, this literal will be printed to show that categories are being summed.

```
ITEM 'TOTAL' @41.0 3F10.0 TOTALHU TOTRENT TOTOWN
```

The literal 'TOTAL' will be printed followed by blanks to position 41. Three 10 character fields will be printed containing the sums of all housing units, renter-occupied and owner-occupied housing units.

Run 4

Calculation of Subtotals and Additional Labelling Features

This run shows how subgroups are labelled. It will print universe labels of more than 40 characters on one line. Labels over 40 characters in length require special formatting because CENSPAC stores labels in 40 character fields. This run also shows how to calculate and print subtotals and a grand total, with labels.

Commands:

```
FILEIN STF1
IF SUMRYLVL = '11' INCLUDE
COMPUTE TOT2PLUS = TAB21 (2) + TAB21 (3)
COMPUTE TOTALALL = 0
COMPUTE TOTALALL = SUM(TAB21(*))
REPORT
HEAD 'REPORT GENERATION - RUN 4' X20.0 'PAGE ' $PAGE
ITEM
ITEM 'TABLE 21 FOR COUNTY: ' AREANAME  ITEM
ITEM H40.0 TAB21
ITEM U39.0 TAB21 U30.053 TAB21 ITEM
ITEM '1 PERSON:' X16.0 TAB21 (1)
ITEM '2 OR MORE PERSONS:' X7.0 F9.0 TOT2PLUS
ITEM '  FAMILY HOUSEHOLD' X7.0 TAB21 (2)
ITEM '  NONFAMILY HOUSEHOLD' X4.0 TAB21 (3)
ITFM 'TOTAL:' X19.0 F9.0 TOTALALL
```

Printout:

```
REPORT GENERATION - RUN 4                          PAGE       1

TABLE 21 FOR COUNTY: CHESTERFIELD

PERSONS IN HOUSEHOLD AND HOUSEHOLD TYPE
UNIVERSE:  HOUSEHOLDS WITH ONE OR MORE PERSONS 60 YEARS AND OVER

1 PERSON:                        1346
2 OR MORE PERSONS:               4929
  FAMILY HOUSEHOLD               4832
  NONFAMILY HOUSEHOLD              97
TOTAL:                           6275

TABLE 21 FOR COUNTY: HENRICO

PERSONS IN HOUSEHOLD AND HOUSEHOLD TYPE
UNIVERSE:  HOUSEHOLDS WITH ONE OR MORE PERSONS 60 YEARS AND OVER

1 PERSON:                        4259
2 OR MORE PERSONS:              11377
  FAMILY HOUSEHOLD              11115
  NONFAMILY HOUSEHOLD             262
TOTAL:                          15636

TABLE 21 FOR COUNTY: RICHMOND CITY

PERSONS IN HOUSEHOLD AND HOUSEHOLD TYPE
UNIVERSE:  HOUSEHOLDS WITH ONE OR MORE PERSONS 60 YEARS AND OVER

1 PERSON:                       10401
2 OR MORE PERSONS:              18169
  FAMILY HOUSEHOLD              16838
  NONFAMILY HOUSEHOLD            1331
TOTAL:                          28570
```

```
FILEIN STF1
```

Summary Tape File 1 is the input file.

```
IF SUMRYLVL = '11' INCLUDE
```

Only records with a summary level code of 11 will be included in the processing. Code 11 identifies county level records.

```
COMPUTE TOT2PLUS = TAB21 (2) + TAB21 (3)
```

The number of households with 2 or more persons will be calculated by adding cells 2 and 3 of Table 21.

```
COMPUTE TOTALALL = 0
```

Before using the SUM function, the value of total households in Table 21 is set to zero. See discussion of initialization in Run 3.

```
COMPUTE TOTALALL = SUM (TAB21 (*))
```

The total number of households will be calculated by summing across all categories of Table 21.

```
REPORT
```

A printed report will be produced.

```
HEAD 'REPORT GENERATION - RUN 4' X20.0 'PAGE ' $PAGE
```

The literal, 'REPORT GENERATION - RUN 4' will be printed as a heading at the top of each page. After skipping 20 spaces, the word "PAGE", and the page number ($PAGE) will be printed.

```
ITEM
```

A blank line will be printed.

ITEM 'TABLE 21 FOR COUNTY: ' AREANAME ITEM

The literal identifying the table will be printed, followed by the value of AREANAME. A blank line will then be printed.

ITEM H40.0 TAB21

The heading label for Table 21, PERSONS IN HOUSEHOLD AND HOUSEHOLD TYPE, will be printed in a field of 40 characters (H40.0)

ITEM U39.0 TAB21 U30.053 TAB21 ITEM

The universe label will be printed on one line. The first line of the universe label for Table 21 in the data dictionary, "Universe: Households with One or More", will be printed (U39.0). A field of 39 allows room for this label and one blank for spacing. The next format code (U30.053) will print the next line of universe label in a field of 30 characters, beginning in position 53 of the label. Because the universe label and its continuation line are to be printed on the same line, the 12 leading blanks which are part of the continuation label (the blanks indent the label past the word "Universe: ") are eliminated, and printing will begin with character 53.

Note: When printing labels, if a beginning position (the position to the right of the decimal point, i.e. 053 in U30.053) other than zero is specified, a three digit number must be used. Even though the universe label and its continuation line are two separate labels, when a beginning position is specified, CENSPAC treats them as a single 80 character label.

ITEM '1 PERSON: ' X16.0 TAB21 (1)

Category labels for this table are provided with literals so that the group label, '2 or more persons: ' (see below) can be inserted as a label. If users want group labels included among category labels, they must specify all the category labels with literals because category labels from the data dictionary cannot be printed individually, but must be printed for all categories. This ITEM command will print the first category label, skip 16 spaces, and then print cell 1 of Table 21.

```
ITEM '2 OR MORE PERSONS: ' X7.0 F9.0 TOT2PLUS
```

This is the group label that will appear among the category labels. After skipping seven spaces, the value of the subtotal, TOT2PLUS, will be printed in a field of 9 characters (F9.0).

```
ITEM '  FAMILY HOUSEHOLD' X7.0 TAB21 (2)
```

This ITEM command will print the second category label, indenting two spaces because the label is a subgroup of '2 OR MORE PERSONS: '. After skipping seven spaces, cell 2 of Table 21 will be printed.

```
ITEM '  NONFAMILY HOUSEHOLD' X4.0 TAB21 (3)
```

This ITEM command will print the third category label, indented to indicate a subgroup. After skipping four spaces, cell 3 of Table 21 will be printed.

```
ITEM 'TOTAL: ' X19.0 F9.0 TOTALALL
```

The literal will label the newly computed total. After skipping 19 spaces, the total will be printed in a field of 9 characters (F9.0).

Run 5

Geographic and Component Area Summaries

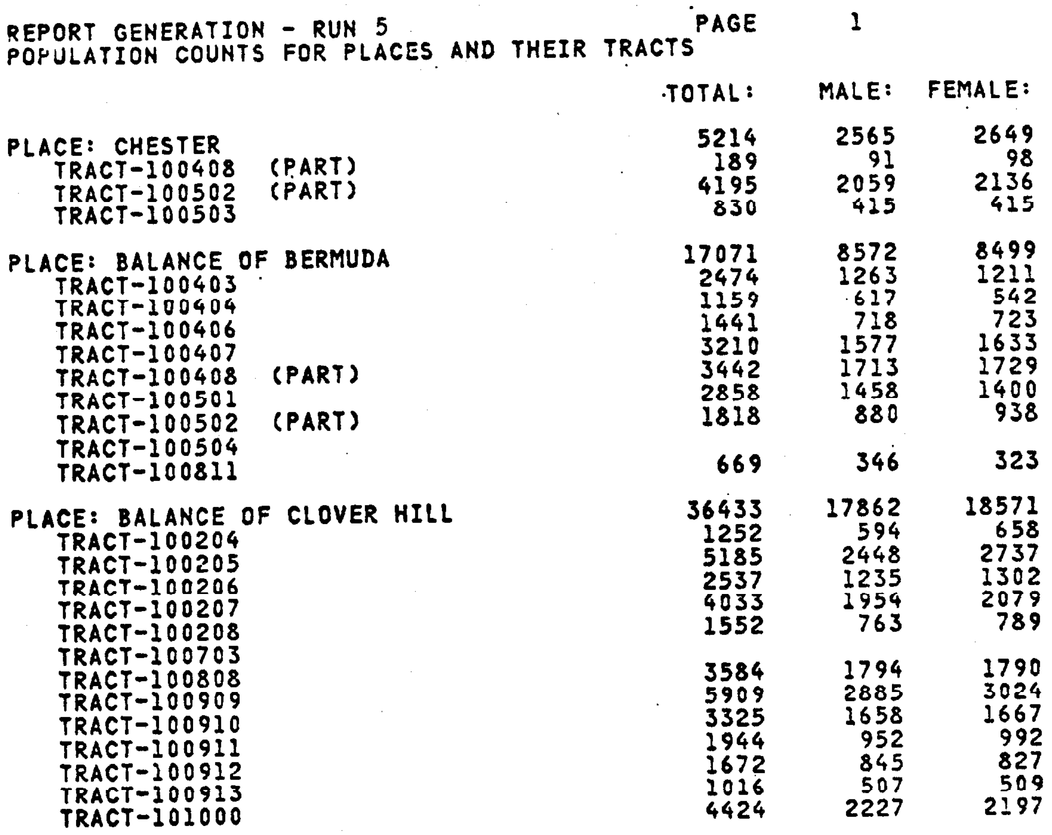
Population counts are printed for places, followed by summaries for the component areas within places. Both the places and their components are identified. This run could be modified to print any records nested by geographic area. The ordering of the output is dependent on the sort sequence of records in a file. Because of the hierarchical sequencing of the census summary tape files, a place summary record is followed by summary records for the samller units of geography, including tracts, within the place. Since summaries respect the boundaries of the larger units in which they are nested, partial tracts are shown when a tract is split by place boundaries. These partial tract summaries are identified in the printout.

Commands:

```
FILEIN STF1
IF SUMRYLVL = '13' OR SUMRYLVL = '14' INCLUDE
*
* CALCULATE TOTAL POPULATION USING SUM
COMPUTE TOTPOP = 0
COMPUTE TOTPOP = SUM(TAB6(*))
REPORT HEAD 'REPORT GENERATION - RUN 5' X20.0 'PAGE ' $PAGE
HEAD 'POPULATION COUNTS FOR PLACES AND THEIR TRACTS'
HEAD
HEAD X40.0 '   TOTAL:    MALE:  FEMALE:'
*
* IF PLACE SUMMARY, LABEL PLACE
IF SUMRYLVL = '13'  ITEM
ITEM 'PLACE: ' A33.0 AREANAME 3F9.0 TOTPOP TAB6(*);
*
* IF TOTAL TRACT SUMMARY, LABEL TRACT
IF SUMRYLVL = '14' AND PARTTRCT = '2'
ITEM '   TRACT-' TRACT X25.0 3F9.0 TOTPOP TAB6(*);
*
* IF PART TRACT SUMMARY, LABEL AS PART TRACT
IF SUMRYLVL = '14' AND PARTTRCT = '1'
ITEM '   TRACT-' TRACT '  (PART)  ' X15.0
      3F9.0 TOTPOP TAB6 (*)
```

Run5

Printouts:



```
REPORT GENERATION - RUN 5                              PAGE      1
POPULATION COUNTS FOR PLACES AND THEIR TRACTS
                                                    TOTAL:     MALE:   FEMALE:

PLACE: CHESTER                                       5214      2565      2649
   TRACT-100408  (PART)                               189        91        98
   TRACT-100502  (PART)                              4195      2059      2136
   TRACT-100503                                       830       415       415

PLACE: BALANCE OF BERMUDA                           17071      8572      8499
   TRACT-100403                                      2474      1263      1211
   TRACT-100404                                      1159       617       542
   TRACT-100406                                      1441       718       723
   TRACT-100407                                      3210      1577      1633
   TRACT-100408  (PART)                              3442      1713      1729
   TRACT-100501                                      2858      1458      1400
   TRACT-100502  (PART)                              1818       880       938
   TRACT-100504
   TRACT-100811                                       669       346       323

PLACE: BALANCE OF CLOVER HILL                       36433     17862     18571
   TRACT-100204                                      1252       594       658
   TRACT-100205                                      5185      2448      2737
   TRACT-100206                                      2537      1235      1302
   TRACT-100207                                      4033      1954      2079
   TRACT-100208                                      1552       763       789
   TRACT-100703
   TRACT-100808                                      3584      1794      1790
   TRACT-100909                                      5909      2885      3024
   TRACT-100910                                      3325      1658      1667
   TRACT-100911                                      1944       952       992
   TRACT-100912                                      1672       845       827
   TRACT-100913                                      1016       507       509
   TRACT-101000                                      4424      2227      2197
```

```
FILEIN STF1
```

Summary Tape File 1 is the input file.

```
IF SUMRYLVL = '13' OR SUMRYLVL = '14' INCLUDE
```

Only records with a summary level code of 13 or 14 will be included in the processing. Code 13 identifies place or partial place records. Tract or partial tract records are identified by code 14.

```
*
* CALCULATE TOTAL POPULATION USING SUM
```

These comments describe the purpose of the commands that follow.

```
COMPUTE TOTPOP = 0
```

The value of total population is set to zero, before deriving a sum total. See discussion of initialization in Run 3.

```
COMPUTE TOTPOP = SUM (TAB6 (*))
```

The total population is calculated by summing across the stratifier of TAB6, sex.

```
REPORT HEAD 'REPORT GENERATION - RUN 5' X20.0 'PAGE ' $PAGE
```

A printed report will be produced. At the top of each page, a heading will be printed with the literal, followed by 20 blank spaces, the word "PAGE", and the page number.

```
HEAD 'POPULATION COUNTS FOR PLACES AND THEIR TRACTS'
```

Another heading will be printed containing the literal shown here. The heading is split into two literals because it exceeds 40 characters.

```
HEAD
```

A blank line will be printed.

```
HEAD X40.0 '   TOTAL:    MALE  FEMALE:'
```

After skipping 40 spaces, the literal will be printed. It will label the data columns.

```
*
* IF PLACE SUMMARY, LABEL PLACE
```

These comment cards describe the purpose of the record selection and labeling that follows.

```
IF SUMRYLVL = '13' ITEM
ITEM 'PLACE: ' A33.0 AREANAME 3F9.0 TOTPOP TAB6 (*);
```

If the record is a place summary, a blank line will first be printed. On the next line, the literal 'PLACE: ' will be printed, followed by the value of AREANAME in a field of 33 characters (A33.0), and three data values in 9 character fields -- total population, and the two data cells of Table 6. The 33 character width of AREANAME was calculated to allow space for the largest name associated with a record. The IF command is terminated by the semi-colon (;).

```
*
* IF TOTAL TRACT SUMMARY, LABEL TRACT
```

These comment cards describe the purpose of the record selection and labeling that follows.

```
IF SUMRYLVL = '14' AND PARTTRACT = '2'
ITEM '  TRACT-'  TRACT X25.0 3F9.0 TOTPOP TAB6(*);
```

If the record is a tract summary and the tract is not split, then a line will be printed containing the literal, followed by the tract code. After skipping 25 spaces, three data values in 9 character fields will be printed -- total population and the two data cells of Table 6. The IF command is terminated by the semicolon (;).

```
*
* IF PART TRACT SUMMARY, LABEL AS PART TRACT
```

These comment cards describe the purpose of the record selection and labeling that follows.

Run 5

```
IF SUMRYLVL = '14' AND PARTTRACT = '1'
ITEM '  TRACT-' TRACT ' (PART) ' X15.0
     3F9.0 TOTPOP TAB6 (*)
```

If the record is a tract summary and the tract is split, then a line will be printed containing the literal, followed by the tract code and another literal identifying the summaries as partial totals. After skipping 15 spaces, three data values in 9 character fields will be printed -- total population and the two data cells of Table 6. The IF command is terminated by the end of the run.

Run 6

Calculating Cumulative Percentages

A table is produced in this run displaying population counts, percentages, and cumulative percentages for four populations. It illustrates recoding categories, here "other races," and describes how to compute cumulative percentages.

Commands:

```
FILEIN STF1
*
* CALCULATE "OTHER" RACES COUNTS BY AGE GROUP AND TOTAL ACROSS ALL AGES
*
COMPUTE TABOTHER (1,*) = TAB12 (1,*) - (TAB12 (2,*) + TAB12 (3,*))
COMPUTE OTHERSUM = 0
COMPUTE OTHERSUM = SUM(TABOTHER (1,*))
*
* CALCULATE RACE PERCENTAGES BY RACE
*
COMPUTE TABRACE (1,*) = TAB12 (1,*) / TAB1 (1) * 100 + 0.05
COMPUTE TABRACE (2,*) = TAB12 (2,*) / TAB7 (1) * 100 + 0.05
COMPUTE TABRACE (3,*) = TAB12 (3,*) / TAB7 (2) * 100 + 0.05
COMPUTE TABRACE (4,*) = TABOTHER (1,*) / OTHERSUM * 100 + 0.05
*
* CUMULATE PERCENTAGES BY RACE
*
COMPUTE TABCUMR (*,1) = TABRACE (*,1)
COMPUTE TABCUMR (*,2) = TABRACE (*,2) + TABCUMR (*,1)
COMPUTE TABCUMR (*,3) = TABRACE (*,3) + TABCUMR (*,2)
COMPUTE TABCUMR (*,4) = 100
*
REPORT
HEAD 'REPORT GENERATION - RUN 6'
HEAD X50.0 'AGE DISTRIBUTION BY RACE'
ITEM     ITEM
ITEM X39.0 'TOTAL' X15.0 'WHITE' X15.0 'BLACK' X15.0 'OTHER'
ITEM X33.0 '------------------- -------------------'
     ' ------------------- --------------------'
ITEM X61.0 '% OF' X16.0 '% OF' X16.0 '% OF'
ITEM X41.0 '% OF   CUM          WHITE  CUM' X10.0
      'BLACK  CUM          OTHER  CUM'
ITEM X22.0 'AGE           #    TOTAL   %      #'
      '     POP    %      #     POP    %      #' '     POP    %'
ITEM X15.0 '----------------- ------- -----'
     ' ----- ------- ----- -----'
     ' ------- ----- ----- ------- ----- -----'
ITEM
ITEM X15.0 C18.0 TAB12 (1,C)
      F7.0 TAB12 (1,C)  F6.1 TABRACE (1,C)  F6.1 TABCUMR (1,C)
      F7.0 TAB12 (2,C)  F6.1 TABRACE (2,C)  F6.1 TABCUMR (2,C)
      F7.0 TAB12 (3,C)  F6.1 TABRACE (3,C)  F6.1 TABCUMR (3,C)
      F7.0 TABOTHER (1,C)  F6.1 TABRACE (4,C)  F6.1 TABCUMR (4,C)
```

```
REPORT GENERATION - RUN 6
                                          AGE DISTRIBUTION BY RACE

                         TOTAL                  WHITE                  BLACK                  OTHER
                  ------------------    ------------------    ------------------    ------------------
                                               % OF                   % OF                   % OF
                           % OF   CUM            WHITE  CUM            BLACK  CUM            OTHER  CUM
       AGE           #     TOTAL   %        #     POP    %        #     POP    %        #     POP    %
----------------- ------- ----- -----  ------- ----- -----  ------- ----- -----  ------- ----- -----

Under 5 years       32594   6.3   6.3   21543   5.8   5.8   10565   7.3   7.3     486  10.1  10.1
5 to 17 years      108034  20.8  27.1   71708  19.4  25.3   35244  24.4  31.7    1082  22.5  32.6
18 to 64 years     326359  62.9  90.1  236166  63.9  89.2   87137  60.3  92.0    3056  63.5  96.1
65 years and over   51952  10.0 100.0   40100  10.9 100.0   11660   8.1 100.0     192   4.0 100.0

                         TOTAL                  WHITE                  BLACK                  OTHER
                  ------------------    ------------------    ------------------    ------------------
                                               % OF                   % OF                   % OF
                           % OF   CUM            WHITE  CUM            BLACK  CUM            OTHER  CUM
       AGE           #     TOTAL   %        #     POP    %        #     POP    %        #     POP    %
----------------- ------- ----- -----  ------- ----- -----  ------- ----- -----  ------- ----- -----

Under 5 years        9182   7.3   7.3    8316   7.4   7.4     739   6.0   6.0     127  10.7  10.7
5 to 17 years       32831  26.0  33.4   29890  26.5  34.0    2626  21.4  27.5     315  26.4  37.2
18 to 64 years      78409  62.2  95.6   69572  61.8  95.8    8114  66.1  93.6     723  60.7  97.9
65 years and over    5712   4.5 100.0    4889   4.3 100.0     797   6.5 100.0      26   2.2 100.0

                         TOTAL                  WHITE                  BLACK                  OTHER
                  ------------------    ------------------    ------------------    ------------------
                                               % OF                   % OF                   % OF
                           % OF   CUM            WHITE  CUM            BLACK  CUM            OTHER  CUM
       AGE           #     TOTAL   %        #     POP    %        #     POP    %        #     POP    %
----------------- ------- ----- -----  ------- ----- -----  ------- ----- -----  ------- ----- -----

Under 5 years        1526   6.8   6.8    1366   6.7   6.7     138   8.2   8.2      22  10.0  10.0
5 to 17 years        5756  25.8  32.7    5306  26.0  32.8     390  23.0  31.3      60  27.3  37.3
18 to 64 years      13845  62.1  94.9   12698  62.3  95.2    1013  59.9  91.2     134  60.9  98.3
65 years and over    1158   5.2 100.0    1003   4.9 100.0     151   8.9 100.0       4   1.8 100.0

                         TOTAL                  WHITE                  BLACK                  OTHER
                  ------------------    ------------------    ------------------    ------------------
                                               % OF                   % OF                   % OF
                           % OF   CUM            WHITE  CUM            BLACK  CUM            OTHER  CUM
       AGE           #     TOTAL   %        #     POP    %        #     POP    %        #     POP    %
----------------- ------- ----- -----  ------- ----- -----  ------- ----- -----  ------- ----- -----

Under 5 years         311   6.0   6.0     293   5.9   5.9      11   5.2   5.2       7  13.2  13.2
5 to 17 years        1388  26.6  32.6    1325  26.8  32.7      43  20.3  25.5      20  37.7  51.0
18 to 64 years       3196  61.3  94.0    3037  61.4  94.2     134  63.2  88.8      25  47.2  98.2
65 years and over     319   6.1 100.0     294   5.9 100.0      24  11.3 100.0       1   1.9 100.0

                         TOTAL                  WHITE                  BLACK                  OTHER
```

```
FILEIN STF1
```

Summary Tape File 1 is the input file.

```
*
* CALCULATE "OTHER" RACES COUNTS BY AGE GROUP AND TOTAL
*           ACROSS ALL AGES
```

These comment cards describe the purpose of the computations that follow.

```
COMPUTE TABOTHER (1,*) = TAB12 (1,*) - (TAB12 (2,*) + TAB12 (3,*))
```

The count of "other" persons for four age groups will be calculated by subtracting the sum of Whites and Blacks from the total for each age group. TABOTHER will have the same dimensions, (5 x 4), as TAB12. This command is an example of how subgroup counts are derived through subtraction. The total population is comprised of White; Black; American Indian; Eskimo and Aleut; Asian and Pacific Islander; and "other" races. The "other" races category that is being computed is a recode that includes American Indian, Eskimo, Aleut, Asian, Pacific Islander, and "other". Consequently, it is derived by subtracting the White and Black population from the total population.

```
COMPUTE OTHERSUM = 0
```

The number of persons of "other races" is set to zero before deriving a sum total. See discussion of initialization in Run 4.

```
COMPUTE OTHERSUM = SUM (TABOTHER (1,*))
```

The number of persons of "other races" is calculated by summing across age groups (1,*).

```
*
* CALCULATE RACE PERCENTAGES BY AGE
*
```

These comment cards describe the purpose of the computations that follow.

```
COMPUTE TABRACE (1,*) = TAB12 (1,*) / TAB1  (1) * 100 + 0.05
```

This statement, as well as the three that follow, will calculate the percentage of the population in each age group for the total population and for each racial group (White, Black, and other). The above command will calculate the percentage of persons in each age group for the total population. Each age group ((TAB12 (1,*)) is divided by the total population (TAB1 (1)) and multiplied by 100. The 0.05 is added to allow for rounding when the number is truncated to one decimal place.

```
COMPUTE TABRACE (2,*) = TAB12 (2,*) / TAB7 (1) * 100 + 0.05
```

The percentage of persons in each age group for the white population is calculated and placed in TABRACE. Each White age group (TAB12 (2,*)) is divided by the White population (TAB7 (2)) and multiplied by 100. The 0.05 is added to allow for rounding when the number is truncated to one decimal place.

```
COMPUTE TABRACE (3,*) = TAB12 (3,*) / TAB7 (2) * 100 + 0.05
```

The percentage of persons in each age group for the Black population is calculated and placed in TABRACE. Each Black age group (TAB12 (3,*)) is divided by the Black population (TAB7 (3)) and multiplied by 100. The 0.05 is added to allow for rounding when the number is truncated to one decimal place.

```
COMPUTE TABRACE (4,*) = TABOTHER (1,*) / OTHERSUM * 100 + 0.05
```

The percentage of persons in each age group for the "other races" population is calculated and placed in TABRACE. Each "other races" age group (TABOTHER (1,*)) is divided by the "other races" population (OTHERSUM) and multiplied by 100. The 0.05 is added to allow for rounding when the number is truncated to one decimal place.

```
*
* CUMULATE PERCENTAGES BY RACE
*
```

These comment cards describe the purpose of the computations that follow.

```
COMPUTE TABCUMR (*,1) = TABRACE (*,1)
```

Age group percentages will be cumulated and stored, by race, in table TABCUMR, which will have the same dimensions as TABRACE (5 x 4). The first age group becomes the first entry in the cumulative table. The first entry in the cumulative table, (TABCUMR (*,1)), will be the percentage of each racial group that the first age group represents. This will be done for all racial groups.

```
COMPUTE TABCUMR (*,2) = TABRACE (*,2) + TABCUMR (*,1)
```

The second entry in the cumulative percentage table, (TABCUMR (*,2)), will be equal to the first entry (TABCUMR (*,1)) plus the percentage of each racial group in the second age group (TABRACE (*,2)). This calculation will be done for all racial groups.

```
COMPUTE TABCUMR (*,3) = TABRACE (*,3) + TABCUMR (*,2)
```

The third entry in the cumulative percentage table, (TABCUMR (*,3)), is equal to the second entry (TABCUMR (*,2)) plus the percentage of each race group in the third age group (TABRACE (*,3)). This calculation will be done for all racial groups.

```
COMPUTE TABCUMR (*,4) = 100
```

The last cumulated percentage equals 100. While this could be derived as in the above commands, setting the value to 100 avoids any possible deviation due to rounding of cumulated numbers.

```
*
 REPORT
 HEAD 'REPORT GENERATION - RUN 6'
 HEAD X50.0 'AGE DISTRIBUTION BY RACE'
```

A printed report will be produced. At the top of each page, a two line heading will appear. The first line will contain the literal, 'REPORT GENERATION - RUN 8' The next line will title the table, and it will be centered on the page by skipping 40 spaces.

```
ITEM          ITEM
ITEM X39.0 'TOTAL' X15.0 'WHITE' X15.0 'BLACK' X15.0 'OTHER'
ITEM X33.0 '------------------ ------------------'
           '------------------ ------------------'
```

This series of commands provide the population group labels with column headings underlined by a series of dashes.

```
ITEM X61.0 '% OF' X16.0 '% OF' X16.0 '% OF'
ITEM X41.0 '% OF' X3.0 'CUM' X10.0 'WHITE' X2.0 'CUM'
  X10.0 'BLACK' X2.0 'CUM' X10.0 'OTHER' X2.0 'CUM'
ITEM X22.0 'AGE' X11.0 '#' X4.0 'TOTAL' X3.0 '%'
  X6.0 '#' X5.0 'POP' X4.0 '%' X6.0 '#' X5.0 'POP'
  X4.0 '%' X6.0 '#' X5.0 'POP' X4.0 '%'
ITEM X15.0 '-----------------' X1.0 '-------' X1.0 '-----'
  X1.0 '-----' X1.0 '-------' X1.0 '-----'.X1.0 '-----' X1.0
'-------' X1.0 '-----' X1.0'-----' X1.0 '-------' X1.0
'-----' X1.0 '-----'

ITEM
```

This series of commands provide column headings that identify the percentages.

```
ITEM X15.0 C18.0 TAB12 (1,C)
F7.0 TAB12 (1,C) F6.1 TABRACE (1,C) F6.1 TABCUMR (1,C)
F7.0 TAB12 (2,C) F6.1 TABRACE (2,C) F6.1 TABCUMR (2,C)
F7.0 TAB12 (3,C) F6.1 TABRACE (3,C) F6.1 TABCUMR (3,C)
F7.0 TABOTHER (1,C) F6.1 TABRACE (4,C) F6.1 TABCUMR (4,C)
```

With one ITEM command, all the tabular data will be printed using the "C" operator. The category labels for the age groups are first printed in columns (C17.0 TAB12 (1,C)). Then, for each racial group, the population counts are printed in numeric fields of 8 characters, and the two percentages are printed in numeric fields of 6 characters each, with one place to the right of the decimal point. The values for the category, age, are listed in columns for all three data items.

Although it may seem that the statements for the first three racial groups could be combined using an asterisk (*) to indicate all categories of the first dimensions of TAB12, TABRACE, and TABCUMR, this will not work, even though all of these tables contain five categories in the first stratifier. Table 12, a 5 x 4 table, contains population counts by race, while TABRACE and TABCUMR, also 5 x 4 tables, contain one category of zero-filled data cells not used in the computations. Therefore, using the asterisk (*) subscript would print five categories instead of the four that are needed.

Run 7

Deriving Category Counts by Subtraction

Identifying Data Suppression

How to derive "missing" categories and identify suppression of data are shown in this run. Both examples apply to many tables in the summary tape files.

Data which are suppressed will be flagged in the report. In the two-dimensional table, category counts are derived by subtraction for each stratifier, and data cells are set equal to zero when affected by suppression.

Below is a diagram of the table to be constructed, indicating the source of the data cells.

	RACE			
TENURE	TOTAL	WHITE	BLACK	OTHER
TOTAL	TAB28 (1,1) No suppression	TAB28 (1,2) No suppression	TAB28 (1,3) No suppression	TABOTHER (1,1) No suppression
RENTER	TAB28 (2,1) Supflg 19	TAB28 (2,2) Supflg 20	TAB28 (2,3) Supflg 21	TABOTHER (2,1) Supflg 19, 20, or 21
OWNER	TABOWNER (1,1) Supflg 19	TABOWNER (1,2) Supflg 20	TABOWNER (1,3) Supflg 21	OTHEROWN Supflg 19, 20, or 21

Run 7

Commands:

```
FILEIN STF1
IF SUMRYLVL = '14' INCLUDE
*
* COMPUTE HOUSEHOLDERS OF OTHER RACES
COMPUTE TABOTHER (*,1) = TAB28 (*,1) - (TAB28 (*,2) + TAB28 (*,3))
REPORT
HEAD 'REPORT GENERATION - SUPPRESSION EXAMPLE'  X10.0 $PAGE
HEAD 'TABLE 28: ' H40.0 TAB28
HEAD U40.0 TAB28
HEAD U34.041 TAB28  U10.092 TAB28
ITEM
ITEM 'TRACT - '  TRACT
*
* COMPUTE NUMBER OF OWNERS
COMPUTE TABOWNER (1,*) = TAB28 (1,*) - TAB28 (2,*)
COMPUTE OTHEROWN = TABOTHER (1,1) - TABOTHER (2,1)
*
* SET OWNER CELL = 0 IF SUPPRESSION OF RENTERS
IF SUPFLG19 = '1'
COMPUTE TABOWNER (1,1) = 0
COMPUTE TABOTHER (2,1) = 0
COMPUTE OTHEROWN = 0;
*
IF SUPFLG20 = '1'
COMPUTE TABOWNER (1,2) = 0
COMPUTE TABOTHER (2,1) = 0
COMPUTE OTHEROWN = 0;
*
IF SUPFLG21 = '1'
COMPUTE TABOWNER (1,3) = 0
COMPUTE TABOTHER (2,1) = 0
COMPUTE OTHEROWN = 0;
*
* PRINT CATEGORY LABELS AND DATA CELLS
ITEM X22.0 3C9.0 TAB28 (1,*)  'OTHER'
ITEM C17.0 TAB28 (C,1) TAB28 (C,*)  F9.0 TABOTHER (C,1)
ITEM 'OWNER OCCUPIED:  '3F9.0 TABOWNER (1,*)  F9.0 OTHEROWN
*
* PRINT SUPPRESSION INDICATION
IF SUPFLG19 = '1'
ITEM 'SPANISH ORIGIN HOUSEHOLDER CHARACTERISTICS SUPPRESSED';
IF SUPFLG20 = '1'
ITEM 'SPANISH-WHITE HOUSEHOLDER CHARACTERISTICS SUPPRESSED';
IF SUPFLG21 = '1'
ITEM 'SPANISH-BLACK HOUSEHOLDER CHARACTERISTICS SUPPRESSED'
```

Printout:

```
REPORT GENERATION - SUPPRESSION EXAMPLE                         1
TABLE 28: TENURE (2) BY RACE OF HOUSEHOLDER (3)
Universe:  Occupied Housing Units With
              Householder Of Spanish Origin

TRACT - 100408
                          Total     White     Black     OTHER
Total:                      2         2
Renter occupied:
OWNER OCCUPIED:
SPANISH ORIGIN HOUSEHOLDER CHARACTERISTICS SUPPRESSED
SPANISH-WHITE HOUSEHOLDER CHARACTERISTICS SUPPRESSED

TRACT - 100502
                          Total     White     Black     OTHER
Total:                     25        24                   1
Renter occupied:            3
OWNER OCCUPIED:            22
SPANISH-WHITE HOUSEHOLDER CHARACTERISTICS SUPPRESSED

TRACT - 100503
                          Total     White     Black     OTHER
Total:                      9         9
Renter occupied:            5         5
OWNER OCCUPIED:             4         4

TRACT - 100403
                          Total     White     Black     OTHER
Total:                     23        19         3         1
Renter occupied:            6         4
OWNER OCCUPIED:            17        15
SPANISH-BLACK HOUSEHOLDER CHARACTERISTICS SUPPRESSED

TRACT - 100404
                          Total     White     Black     OTHER
Total:                     17        17
Renter occupied:            9         9
OWNER OCCUPIED:             8         8

TRACT - 100406
                          Total     White     Black     OTHER
Total:                     19        15         3         1
Renter occupied:           15        11
OWNER OCCUPIED:             4         4
SPANISH-BLACK HOUSEHOLDER CHARACTERISTICS SUPPRESSED

TRACT - 100407
                          Total     White     Black     OTHER
Total:                     27        25         1         1
Renter occupied:            7         6
OWNER OCCUPIED:            20        19
SPANISH-BLACK HOUSEHOLDER CHARACTERISTICS SUPPRESSED
```

```
FILEIN STF1
```

Summary Tape File 1 is the input file.

```
IF SUMRYLVL = '14' INCLUDE
```

Only records with a summary level code of 14 will be included in the processing. Code 14 identifies tract or partial tract records.

```
*
* COMPUTE HOUSEHOLDERS OF OTHER RACES
```

These comment cards describe the purpose of the computations that follow.

```
COMPUTE TABOTHER (*,1) = TAB28 (*,1) - (TAB28 (*,2) + TAB28 (*,3))
```

Total and renter householders of "other" race will be calculated by subtracting the sum of white and black householders from householders of all races. The new values will be placed in TABOTHER. The asterisks (*) indicate that "other" race counts will be calculated for both categories of tenure.

```
REPORT
```

A printed report will be produced.

```
HEAD 'REPORT GENERATION - SUPPRESSION EXAMPLE' X10.0 $PAGE
```

At the top of each page, the literal will be printed, and after skipping 10 spaces, the page number.

```
HEAD 'TABLE 28:  ' H40.0 TAB28
```

This HEAD command will print a line containing the literal followed by the heading for Table 28 in a field of 40 characters (H40.0)

HEAD U40.0 TAB28

This HEAD command will print a line containing the first 40 characters of the universe label (U40.0) for Table 28.

HEAD U34.041 TAB28 U10.092 TAB28

The universe label for Table 28 consists of three 40 character lines. This line of heading will print the last two lines of the universe label. Each of the continuation lines begins with 12 blanks to indent it past the word "Universe". Beginning with position "041" of the universe label, the next 34 characters will be printed, including the 12 leading blanks, terminating with one blank after the last word. From the last line, the word "origin" will be printed without leading spaces, so the beginning position is 80 + 12, or .092.

ITEM

A blank line will be printed.

ITEM 'TRACT - ' TRACT

The literal will be printed followed by the tract code.

```
*
* COMPUTE NUMBER OF OWNERS
```

These comments describe the purpose of the commands that follow.

COMPUTE TABOWNER (1,*) = TAB28 (1,*) - TAB28 (2,*)

Owner occupied households, by race, are calculated by subtracting renters from all occupied, for each of the three categories of race.

COMPUTE OTHEROWN = TABOTHER (1,1) - TABOTHER (2,1)

Owner occupied households for "other" races must be calculated from TABOTHER. Renters are subtracted from all occupied households and the value placed in OTHEROWN.

```
*
* SET OWNER CELLS = 0 IF SUPPRESSION OF RENTERS
```

These comments describe the purpose of the commands that follow. If the suppression flags indicate that renter counts have been suppressed for a group, then owner counts cannot be calculated. If this occurs, owner counts are set to zero.

```
IF SUPFLG19 = '1'
COMPUTE TABOWNER (1,1) = 0
COMPUTE TABOTHER (2,1) = 0
COMPUTE OTHEROWN = 0;
```

If data for Spanish-origin households are suppressed, then values for all owners (TABOWNER (1,1)) and for "other' race renters (TABOTHER (2,1)) and "other" race owners (OTHEROWN) cannot be derived. These values are then set to zero.

```
*
IF SUPFLG20 = '1'
COMPUTE TABOWNER (1,2) = 0
COMPUTE TABOTHER (2,1) = 0
COMPUTE OTHEROWN = 0;
```

If data for Spanish-White households are suppressed, then values for Spanish-White owners (TABOWNER (1,2)), for "other" race renters (TABOTHER (2,1)), and for "other" race owners (OTHEROWN) cannot be derived. These values are then set to zero.

```
IF SUPFLG21 = '1'
COMPUTE TABOWNER (1,3) = 0
COMPUTE TABOTHER (2,1) = 0
COMPUTE OTHEROWN = 0;
```

If data for Spanish-Black households are suppressed, then values for Spanish-Black owners (TABOWNER (1,3)), for "other" race renters (TABOTHER (2,1)), and for "other" race owners (OTHEROWN) cannot be derived. These values are then set to zero.

```
*
* PRINT CATEGORY LABELS AND DATA CELLS
```

These comments describe the purpose of the commands that follow.

```
ITEM X22.0 3C9.0 TAB28 (1,*) 'OTHER'
```

Column headings will be provided with this ITEM command. After skipping 22 spaces, the three category labels for the second stratifier of Table 28 are printed in fields of 9 characters across the page (3C9.0 TAB28 (1,*)). The newly calculated category is labeled with the literal 'OTHER'.

```
ITEM C17.0 TAB28 (C,1) TAB28 (C,*) F9.0 TABOTHER (C,1)
```

Category labels in fields of 17 characters (C17.0) will be printed for the first stratifier, tenure, in a column ((C,1)). This will be followed by data cells, printed using data dictionary default formats. Values from Table 28 will be displayed with tenure categories down the page and race categories across the page ((C,*)). "Other" race counts will be printed in fields of 9 characters with the tenure categories in columns (F9.0 TABOTHER (C,1)).

```
ITEM 'OWNER OCCUPIED:  ' 3F9.0 TABOWNER (1,*) F9.0 OTHEROWN
```

The label and values for owner occupied units will be printed on this line. The label must be supplied as a literal since it is not included in the data dictionary. The owner occupied counts by race, calculated from Table 28, will be displayed in three fields of 9 characters across the page (3F9.0 TABOWNER (1,*)). Owner householders of "other" race (OTHEROWN) will be displayed in a 9 character field (F9.0).

```
*
* PRINT SUPPRESSION INDICATION
```

These comments describe the purpose of the commands that follow.

```
IF SUPFLG19 = '1'
ITEM 'SPANISH ORIGIN HOUSEHOLDER CHARACTERISTICS SUPPRESSED';
```

Data cells that are suppressed will be blank when printed. In order to distinguish suppressed data cells from data cells with a zero count, the flag is checked and a message is printed if data are suppressed.

```
IF SUPFLG20 = '1'
ITEM 'SPANISH-WHITE HOUSEHOLDER CHARACTERISTICS SUPPRESSED';
```

The Spanish-White Householder flag is checked, and, if data are suppressed, a message is printed.

```
IF SUPFLG21 = '1'
ITEM 'SPANISH-BLACK HOUSEHOLDER CHARACTERISTICS SUPPRESSED';
```

The Spanish-Black Householder flag is checked, and, if data are suppressed, a message is printed.

Run 8

Columnar Report By Census Tract

In this report, each tract is presented in a separate column of data. Because of the sequential processing of input repords through the CENSPAC commands, each CENSPAC report line usually contains information for only one geographic area. In this run, data are stored in temporary arrays, including the alphanumeric tract codes, and printed in column format when five tract records have been processed.

Commands:

```
FILEIN STF1
*
* CREATE 5 COLUMNS OF DATA BY TRACT.  A COUNTER
* CONTROLS THE FILLING OF NEW TABLES WITH
* DATA FROM EACH NEW TRACT RECORD.  THE
* END-OF-FILE FLAG IS USED TO INSURE PRINTING
* OF LAST PAGE IF LESS THAN 5 COLUMNS.
* EACH COUNTY WILL BE STARTED ON A NEW PAGE.
IF SUMRYLVL = '04'
   COMPUTE STNAME = AREANAME;
IF SUMRYLVL = '11' AND $INA < 5
   COMPUTE NAME = AREANAME EXCLUDE
IF SUMRYLVL = '14' OR SUMRYLVL = '11' OR $EOFA > 0 INCLUDE
*
*
COMPUTE COL = COL + 1
IF $EOFA > 0 OR SUMRYLVL = '11' COMPUTE COL = COL + 5;
*
* AN ARRAY WILL HOLD ALPHANUMERIC TRACT CODES
* FOR PRINTING AS COLUMN HEADINGS.
*
ARRAY TABTRACT (5)
*
* PLACE TRACT CODE IN TRACT ARRAY.  HOLD TABLE 7
* VALUES IN TEMPORARY ARRAY FOR LATER PRINTING.
*
IF COL = 1
   COMPUTE TABTRACT (1) = TRACT
   COMPUTE TAB7A (*) = TAB7 (*);
IF COL = 2
   COMPUTE TABTRACT (2) = TRACT
   COMPUTE TAB7B (*) = TAB7 (*);
IF COL = 3
   COMPUTE TABTRACT (3) = TRACT
   COMPUTE TAB7C (*) = TAB7 (*);
IF COL = 4
   COMPUTE TABTRACT (4) = TRACT
   COMPUTE TAB7D (*) = TAB7 (*);
IF COL = 5
   COMPUTE TABTRACT (5) = TRACT
   COMPUTE TAB7E (*) = TAB7 (*);
*
* PRINT REPORT IF 5 COLUMNS OF DATA HAVE BEEN FILLED.
*
IF (COL > 5 OR COL = 5)
REPORT
HEAD ' ONE COLUMN TABLE REPORT FOR CENSUS TRACTS     DATE ' $DATE
     ' PAGE ' Z4.0 $PAGE HEAD 'STATE OF   ' STNAME HEAD
ITEM A24.0 NAME 5A10.0 TABTRACT (*) ITEM
ITEM C20.0 TAB7 (C) 5F10.0 TAB7A (C) TAB7B (C)
     TAB7C (C) TAB7D (C) TAB7E (C)
ITEM
*
* INITIALIZES LAST 4 COLUMNS TO COVER CASE OF
* LAST PRINT PAGE CONTAINING LESS THAN 5 TRACTS.
*
COMPUTE TAB7B (*) = 0 COMPUTE TAB7C (*) = 0
COMPUTE TAB7D (*) = 0 COMPUTE TAB7E (*) = 0
COMPUTE TABTRACT (*) = ' '
COMPUTE COL = 0;
IF SUMRYLVL = '11'
   COMPUTE NAME = AREANAME
   COMPUTE $LCTR = 100
```

Run 8

Printout:

```
 ONE COLUMN TABLE REPORT FOR CENSUS TRACTS     DATE 10/30/81 PAGE    4
STATE OF  NEVADA

CLARK COUNTY            000302    0004      000502    000503    000504

White                       40      4779      5585      3729      2596
Black                     5135       456       448       278       381
  American Indian                    164        41        51        40
  Eskimo                                                             1
  Aleut                                                              1
  Japanese                   1        10        27         9        11
  Chinese                             22        52        35         3
  Filipino                            41        26        40        12
  Korean                               6        17         5        49
  Asian Indian                         3         1         7
  Vietnamese                          20        30         8        19
  Hawaiian                   3         7         5         5
  Guamanian                                      1         1
  Samoan                               1
Other                       18       287       376       268       616

CLARK COUNTY            000506    000507    000508    000509    0006

White                     2584      4242      3607      4863      1942
Black                      209       250       134       223       119
  American Indian           11        29        29        23        15
  Eskimo                                         1
  Aleut
  Japanese                  27        18        18        30         2
  Chinese                    2        13        32        33        19
  Filipino                  47        62        87        72        19
  Korean                    11         8         9        32         4
  Asian Indian               1                   4         1         1
  Vietnamese                          24         1         6         6
  Hawaiian                   3         3         2         8         3
  Guamanian                 11         5         1         2         2
  Samoan
Other                      147       188       182       358        41

CLARK COUNTY            0007      0008      0009      001001    001002

White                     1942      2059      1378      5472      5158
Black                      174        90       132        76        78
  American Indian           14        17        12        15        29
  Eskimo
  Aleut                                                              2
  Japanese                  18         6         7         6        18
  Chinese                   21        31        12        18        10
  Filipino                  31         9        55        12        25
  Korean                     8         5         3         5        15
  Asian Indian               5         7         1         9         1
  Vietnamese                 3        25         2         2         9
  Hawaiian                   2         4                   3         7
  Guamanian                  3         1                   1
  Samoan
Other                       88        43        93        68        75
```

```
FILEIN STF1
```

Summary Tape File 1 is the input file.

```
*
* CREATE 5 COLUMNS OF DATA BY TRACT.  A COUNTER
* CONTROLS THE FILLING OF NEW TABLES WITH DATA FROM
* EACH NEW TRACT RECORD.  THE END-OF-FILE FLAG IS USED
* TO INSURE PRINTING OF LAST PAGE IF LESS THAN 5 COLUMNS.
* EACH COUNTY WILL BE STARTED ON A NEW PAGE
```

These comments describe the command procedure to follow.

```
IF SUMRYLVL = '04'
     COMPUTE STNAME = AREANAME;
```

This IF and COMPUTE command sequence saves the state name for later printing in the page headings. The SUMRYLVL = '04' tests for the state level summary and if the condition is true, the data from the AREANAME field will be moved intothe STNAME field.

```
IF SUMRYLVL = '11' AND $INA <5
     COMPUTE NAME = AREANAME EXCLUDE
```

This sequence of commands initialize the county name field. SUMRYLVL = '11' tests for the county level summary and $INA <5 insures that this will be done only within the first 4 records of the data file. In STF1A, the first county record will be the second record. The compute then saves the county name for later use. The EXCLUDE omits further processing for the first county record. These commands only serve only to avoid printing a blank first page, since the county record is used to control printing the last tracts from the previous county and to start a new page of output.

```
IF SUMRYLVL = '14' OR SUMRYLVL = '11' OR $EOFA > 0 INCLUDE
```

This IF command allows only county (11) and tract (14) level summaries to be processed from the input file. It also allows the CENSPAC commands to be processed one last time when end-of-file has been reached to allow the last partial page to be printed.

```
COMPUTE COL = COL +1
```

The COL variable is used to control placing subsequent tracts across the page and when a page is full (5 columns for this example) the page will be printed, cleared to zeroes and processing will continue with the next tract.

Run 8

```
IF $EOFA > 0 OR SUMRYLVL = '11' COMPUTE COL = COL + 5;
```

These IF and COMPUTE commands control printing the last tracts in each county and at end-of-file. A value of 1 thru 4 in the COL variable indicates that the page is being filled with tract data but is not yet full. A value greater than 4 indicates that the page is full and is to be printed. Later CENSPAC commands in this run actually control the printing but this IF and COMPUTE test for the end-of-file condition or the start of the next county and set the COL value to a value greater than 4 to cause the tract data to be printed.

```
* AN ARRAY WILL HOLD ALPHANUMERIC TRACT CODES FOR
* PRINTING AS COLUMN HEADINGS.
*
ARRAY TABTRACT (5)
```

A temporary array is established to hold the tract codes for later printing across the page as column headings.

```
*
* PLACE TRACT CODE IN TRACT ARRAY.  HOLD TABLE 7 VALUES
* IN TEMPORARY ARRAY FOR LATER PRINTING.
*

IF COL = 1

     COMPUTE TABTRACT (1) = TRACT
     COMPUTE TAB7A (*) = TAB7 (*);

IF COL = 2

     COMPUTE TABTRACT (2) = TRACT
     COMPUTE TAB7B (*) = TAB7 (*);

IF COL = 3

     COMPUTE TABTRACT (3) = TRACT
     COMPUTE TAB7C (*) = TAB7 (*);

IF COL = 4

     COMPUTE TABTRACT (4) = TRACT
     COMPUTE TAB7D (*) = TAB7 (*);

IF COL = 5

     COMPUTE TABTRACT (5) = TRACT
     COMPUTE TAB7E (*) = TAB7 (*);
```

For each 5 tract records, the tract code will be placed in a different category of TABTRACT and the data from Table 7 will be held in a different temporary table.

```
*
* PRINT REPORT IF 5 COLUMNS OF DATA HAVE BEEN FILLED.
*
IF (COL > 5 OR COL = 5)
```

If five tract records have been processed and the temporary tables filled, the page must be printed.

```
REPORT
HEAD ' ONE COLUMN TABLE REPORT FOR CENSUS TRACTS     DATE ' $DATE
     ' PAGE ' Z4.0 $PAGE HEAD 'STATE OF ' STNAME HEAD
```

The report will have 3 lines of heading. The first heading will contain the date and a page number. The second heading will list the state name. The third heading line will be blank.

```
ITEM A24.0 NAME 5A10.0 TABTRACT(*) ITEM
```

This ITEM command lists the county name that was saved by COMPUTE command for SUMRYLVL = '11' followed by the tract codes that have been saved in the TABTRACT field. A simple technique for spacing the 6 character tract codes into 10 character columns is used. The 5A10.0 format indicates that the tract codes are to be placed in 10 character fields with the unused spaces blank filled. A blank line will be printed below the tract codes.

```
ITEM C20.0 TAB7 (C) 5F10.0 TAB7A (C) TAB7B (C)
     TAB7C  (C) TAB7D (C) TAB7E (C)
ITEM
```

This ITEM command lists the category labels from Table 7 first. To the right of the labels are 5 columns of data, each 10 characters wide from up to 5 separate census tracts. A blank line is printed following the 15 lines of data that are printed from Table 7.

```
*
* INITIALIZES LAST 4 COLUMNS TO COVER CASE OF LAST PRINT
* PAGE CONTAINING LESS THAN 5 TRACTS
*
COMPUTE TAB7B (*) = 0 COMPUTE TAB7C (*) = 0
COMPUTE TAB7D (*) = 0 COMPUTE TAB7E (*) = 0
COMPUTE TABTRACT (*) = ' '
COMPUTE COL = 0;
```

The last 4 columns are filled with zeroes (TAB7B through TAB7D) to clear the counts from the previous set of 5 tracts processed. The column counter is reset to zero. (The first column is not reset to 0 because the last page printed will have at least one column of good data and a small savings in computer time can be effected by not clearing this field unnecessarily. Correct results would be obtained if 'COMPUTE TAB7A (*) = 0' was also included.

Run 8

```
IF SUMRYLVL = '11'

     COMPUTE NAME = AREANAME
     COMPUTE $LCTR = 100
```

These commands, processed only for county level summaries (SUMRYLVL = '11') save the county name of the new county and set the line counter ($LCTR) to a value larger than the number of lines allowed on a page in order to force data for the next county to be printed on a new page.

Run 9

Creating An Extract File Of Census Tract Records

The STF1 file is used to extract a file of tract records containing selected tables. Because split tract summaries occur on the file, the tract records are SORTed and TOTALed by tract code before extraction. Data dictionary documentation is created to document the extract file.

Commands:

```
FILEIN STF1
IF SUMRYLVL = '14' INCLUDE
SORT COUNTY TRACT
TOTAL COUNTY TRACT
IF $TOTAL = 1
REPORT HEAD 'DATA TO BE WRITTEN ON FILE' HEAD
ITEM COUNTY X2.0 MCD X2.0 PLACE X2.0 TRACT X2.0
     TAB1 TAB5
FILEOUT BLFACTOR 10 DOCUMENT STF1EX
ITEM COUNTY MCD PLACE TRACT TAB1 TAB5;
```

Printout:

```
DATA TO BE WRITTEN ON FILE

041   005   9999   100403        2474       536      1938       947       862       85
041   005   9999   100404        1159                1159       574       512       62
041   005   9999   100406        1441      1441                 644       552       92
041   005   0245   100407        3210      3210                1115      1053       62
041   005   0245   100408         189                            58        56        2
041   005   0245   100502        4195                          1441      1386       55
041   005   0245   100503         830                           332       305       27
```

Run 9

Printout of Output File Created:

```
0410059999100403000002474000000536000001938000000947000000862000000085
0410059999100404000001159000000000000001159000000574000000512000000062
0410059999100406000001441000001441000000000000000644000000552000000092
0410050245100407000000321000000321000000000000001115000001053000000062
0410050245100408000000189000000000000000000000000058000000056000000002
0410050245100502000004195000000000000000000000001441000001386000000055
0410050245100503000000083000000000000000000000000332000000305000000027
```

Printout of Data Dictionary Statements Created:

```
DOCUMENT STF1EX    DISK          70   700
FORMAT   01
D COUNTY        3 1 A C     .FIPS County Code
D MCD           3,, A C     .Minor Civil Division/Census
                            :County Division (MCD/CCD)
D PLACE         4,, A C     .Census Geographic Place
                            :Code
                            :9999 in this field indicates
                            :balance of higher level entity
                            :outside of place. eg; Remainder
                            :of MCD or of county.
D TRACT         6,, A C     .Tract or Block Numbering
                            :Area (BNA)
                            :Tract is a four-digit basic
                            :code with implied decimal and
                            :2-digit suffix
                            :999999 in this field indicates untracted
                            :remainder of a higher level entity,
                            :e.g., Remainder of MCD.
T TAB1          9,, N N
S               3           .Total urbanized area, and rural
H                           .URBAN AND RURAL (3)
                            :
                            : This table has no suppression
                            :
U                           .Universe:  Persons
                            :
                            :NOTE:  Urban is derived by subtracting
                            :         rural from total.
C               1           .Total
C               1           .Inside urbanized areas
C               1           .Rural
T TAB5          9,, N N
S               3           .Occupancy Status
H                           .OCCUPANCY STATUS (3)
                            :
                            : This table has no suppression
                            :
U                           .Universe:  Year-Round Housing
                            :             Units
C               1           .Total
C               1           .Occupied
C               1           .Vacant
END
```

Run 9

Printout of Data Dictionary Produced by Documentor Program Using Data Dictionary Statements:

```
                    STF1EX   DATA DICTIONARY       11/05/81    PAGE    1
                   FILE CHARACTERISTICS SECTION

      FILE      RECORD   BLOCK  STORAGE   RECORD
      NAME       SIZE    SIZE   DEVICE   SEGMENTS

     STF1EX        70     700    TAPE



                       STF1EX   DATA DICTIONARY       11/05/81    PAGE    2
                                  RECORD 01

                 SIZE/
      NAME       SCALE  BEGIN

     COUNTY        3       1
                        FIPS County Code

     MCD           3       4
                        Minor Civil Division/Census
                        Countv Division (MCD/CCD)

     PLACE         4       7
                        Census Geoqraphic Place
                        Code
                        9999 in this field indicates
                        balance of higher level entity
                        outside of place. eq; Remainder
                        of MCD or of county.

     TRACT         6      11
                        Tract or Block Numbering
                        Area (BNA)
                        Tract is a four-digit basic
                        code with implied decimal and
                        2-diqit suffix
                        999999 in this field indicates untracted
                        remainder of a higher level entity,
                        e.q., Remainder of MCD.
                         STF1EX   DATA DICTIONARY       11/05/81    PAGE    3
                                    RECORD 01

                  SIZE/         NUMBER OF
       NAME       SCALE  BEGIN    CELLS

      TABLE 1       9      17       3
      (TAB1)
                 URBAN AND RURAL (3)

                  This table has no suppression

                   Universe:  Persons

                   NOTE:  Urban is derived bv subtractinq
                              rural from total.

                      Total
                      Inside urbanized areas
                      Rural

      TABLE 5       9      44       3
      (TAB5)
                 OCCUPANCY STATUS (3)

                  This table has no suppression

                   Universe:  Year-Round Housinq
                                Units

                      Total
                      Occupied
                      Vacant
```

Run 9

```
FILEIN STF1
```

The input file is STF1 and all file information is provided in the data dictionary.

```
IF SUMRYLVL = '14' INCLUDE
```

Only tract summaries will be included in further processing.

```
SORT COUNTY TRACT
```

In order to get split tract records in the file consecutively within county, the records are sorted by county and tract.

```
TOTAL COUNTY TRACT
```

Records will be aggregated by tract code within county. In most cases, only one record occurs for a tract and thus only one record will be "summarized" in the new total.

```
IF $TOTAL = 1
```

Only summaries resulting from the TOTAL command will be reported and placed in the extract file. This will include both split tract totals and totals for tracts with only one record occurrence.

```
REPORT HEAD 'DATA TO BE WRITTEN ON FILE' HEAD
ITEM COUNTY X2.0 MCD X2.0 PLACE X2.0 TRACT X2.0
     TAB1 TAB5
```

A report will be printed listing the data that will be written on the extract file records. This is useful for testing and verification purposes, but could result in a large volume of printed output when processing all tracts for a large state.

```
FILEOUT BLFACTOR 10 DOCUMENT STF1EX
ITEM COUNTY MCD PLACE TRACT TAB1 TAB5;
```

An extract file will be created with each record containing the items specified in the ITEM command. Format specifications will default to the input specifications and will be documented in a file of data dictionary statements called STF1EX. The file will contain 10 records per block.

The data dictionary definition statements need to be processed through the Documentor program in order to be available for subsequent CENSPAC runs. The data dictionary definition statements are written by CENSPAC to the 'DOC' file. The data in this file must be input to the 'PAR' file of the Documentor program, and the resulting data dictionary will be output on the 'DDF' file of the Documentor program. In almost all cases the 'DDF' file will reference the existing master CENSPAC data dictionary and the new data dictionary being established for the FILEOUT function will be added to this master data dictionary file.

Run 10

Matching Two Files

The extract file created in Run 9 will be matched with a local data file containing school district data. A report is generated showing data for records successfully matched and identifying records not matched.

Commands:

```
FILEIN STF1EX
MATCH       COUNTY TRACT
FILEIN SCHOOL RECSIZE 40 BLKSIZE 1000
ITEM A3.0 CTY A3.0 MCD A6.0 TRACT A3.0 DISTRICT
     3F4.0 ELCHLDRN JHCHLDRN HSCHLDRN 3A2.0 ELSCHOOL
     JHSCHOOL HISCHOOL
MATCH       CTY TRACT
ENDM
REPORT
IF $MATCH = 1
ITEM 'STF1EX FILE NONMATCH' X2.0 COUNTY X2.0 TRACT;
IF $MATCH = 2
ITEM 'SCHOOL FILE NONMATCH' X2.0 B.CTY B.TRACT;
IF $MATCH = 0
REPORT HEAD 'DATA FOR MATCHED RECORDS' HEAD
ITEM 'COUNTY: ' COUNTY '  TRACT: ' TRACT
ITEM H30.0 TAB1 X10.0 H30.0 TAB5
ITEM U30.0 TAB1 X10.0 U30.0 TAB5
ITEM C25.0 TAB1 (C) TAB1 (C)
     @40.0 C25.0 TAB5 (C) TAB5 (C)
ITEM
ITEM 'Number of Elementary Children = ' B.ELCHLDRN
ITEM 'Elementary School Attended: ' B.ELSCHOOL
ITEM
```

Printout:

```
DATA FOR MATCHED RECORDS

COUNTY: 041  TRACT: 100106
URBAN AND RURAL (3)                         OCCUPANCY STATUS (3)
Universe:  Persons                          Universe:  Year-Round Housing
Total                          2979        Total                        952
Inside urbanized areas         2979        Occupied                     930
Rural                                      Vacant                        22

Number of Elementary Children =  584
Elementary School Attended: 14

COUNTY: 041  TRACT: 100107
URBAN AND RURAL (3)                         OCCUPANCY STATUS (3)
Universe:  Persons                          Universe:  Year-Round Housing
Total                          1591        Total                        847
Inside urbanized areas         1591        Occupied                     782
Rural                                      Vacant                        65

Number of Elementary Children =  386
Elementary School Attended: 14

COUNTY: 041  TRACT: 100204
URBAN AND RURAL (3)                         OCCUPANCY STATUS (3)
Universe:  Persons                          Universe:  Year-Round Housing
Total                          1252        Total                        479
Inside urbanized areas         1252        Occupied                     452
Rural                                      Vacant                        27

Number of Elementary Children =  618
Elementary School Attended: 09
```

Run 10

Printout of Match Errors:

```
SCHOOL FILE NONMATCH  041100301
STF1EX FILE NONMATCH  041   100405
STF1EX FILE NONMATCH  041   100504
SCHOOL FILE NONMATCH  041100505
STF1EX FILE NONMATCH  041   100703
STF1EX FILE NONMATCH  041   100807
STF1EX FILE NONMATCH  041   100811
SCHOOL FILE NONMATCH  041100903
SCHOOL FILE NONMATCH  041100906
SCHOOL FILE NONMATCH  041101000
```

Run 10

```
FILEIN STF1EX
MATCH COUNTY TRACT
```

The extract file is the first input file using the data dictionary created from Run 9 and added to the master dictionary file. The keys on which the records will be matched are county and tract codes.

```
FILEIN SCHOOL RECSIZE 40 BLKSIZE 1000
ITEM A3.0 CTY A3.0 MCD A4.0 PLACE A6.0 TRACT
     A3.0 DISTRICT 3F4.0 ELCHLDRN JHCHLDRN
     HSCHLDRN 3A2.0 ELSCHOOL JHSCHOOL HISCHOOL
MATCH CTY TRACT
ENDM
```

The local file with school data does not have a data dictionary associated with it. The file specifications are provided, including record layout information. The keys for matching are the county and tract codes. The match procedure is terminated with ENDM.

```
REPORT
IF $MATCH = 1
ITEM 'STF1EX FILE NONMATCH' X2.0 COUNTY X2.0 TRACT;
IF $MATCH = 2
ITEM 'SCHOOL FILE NONMATCH' X2.0 B.CTY B.TRACT;
```

A report will be printed identifying records not matched from the files.

```
IF $MATCH = 0
REPORT HEAD 'DATA FOR MATCHED RECORDS' HEAD
ITEM 'COUNTY: ' COUNTY '  TRACT: ' TRACT
ITEM H30.0 TAB1 X10.0 H30.0 TAB5
ITEM U30.0 TAB1 X10.0 U30.0 TAB5
```

For those records which match, a report will be created. The first 3 lines will contain the goegraphic identification and the heading and universe labels for Table 1 from the STF1EX file.

```
ITEM C25.0 TAB1 (C) TAB1 (C)
     @40.0 C25.0 TAB5 (C) TAB5 (C)
```

Table 1 will be printed with category labels from the data dictionary. Using the tab feature to start in position 40 of the print page (@40.0), Table 5, with dictionary category labels, will be printed.

```
ITEM
ITEM 'Number of Elementary Children= ' B.ELCHLDRN
ITEM 'Elementary School Attended: ' B.ELSCHOOL
```

Data from the second input file will be reported following the literal strings. Items to be taken from the second file must be prefixed by 'B.' in any references after the ENDM command.

```
ITEM
```

Run 11

Aggregating Data for New Geographic Areas

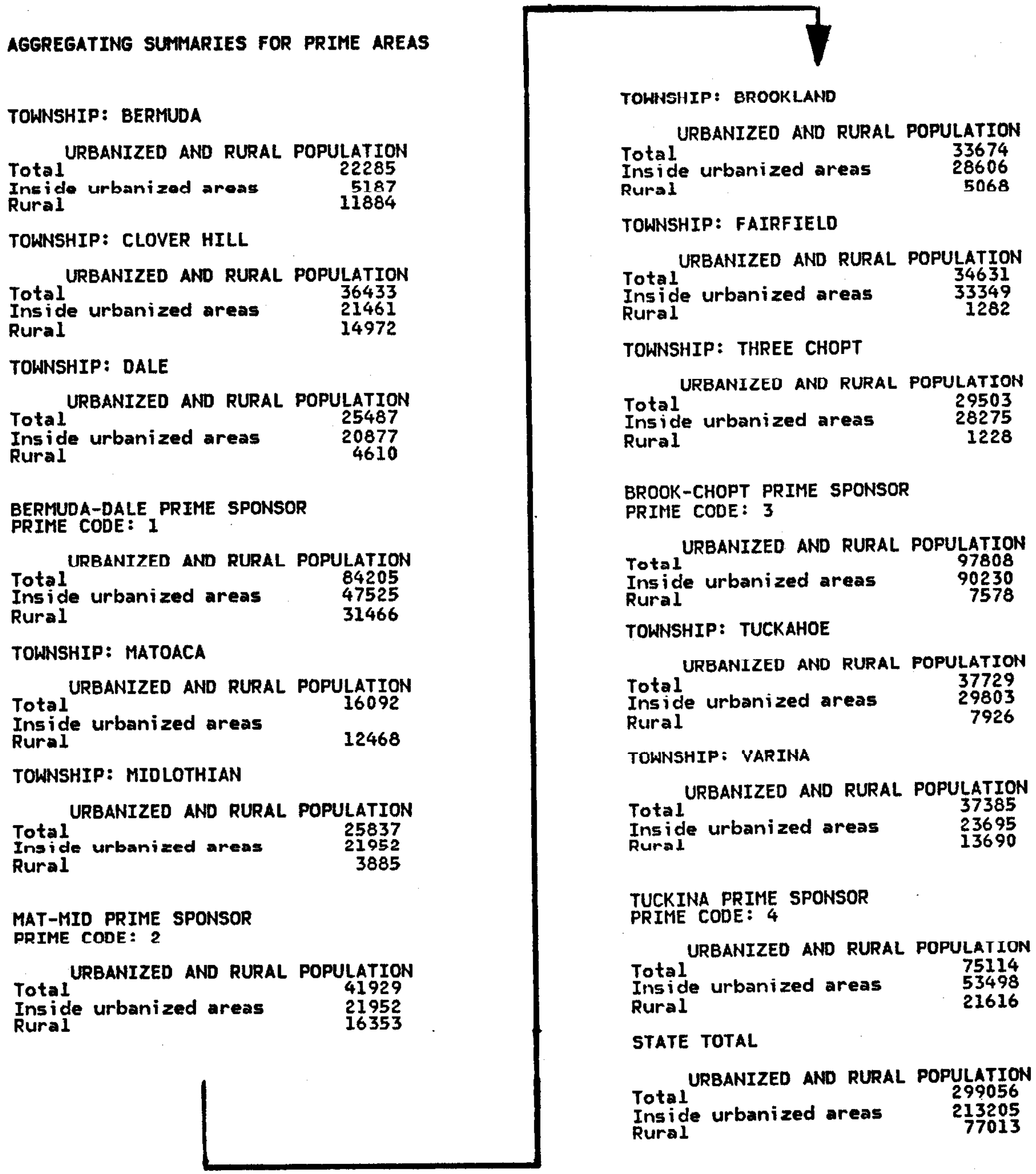
Data are summarized across records to provide new totals for geographic areas comprised of census units of geography. Depending on the county and MCD codes, code values for the new geographic areas, PRIME, are assigned and summaries are aggregated using the TOTAL command.

Commands:

```
FILEIN DICTNAME STF1
IF SUMRYLVL = '12' INCLUDE
COMPUTE PRIME = 0
IF COUNTY = '041' AND (MCD = '005' OR MCD = '010' OR MCD = '015')
  COMPUTE PRIMNAM = 'BERMUDA-DALE PRIME SPONSOR      '
  COMPUTE PRIME = 1;
IF COUNTY = '041' AND (MCD = '025' OR MCD = '030')
  COMPUTE PRIMNAM = 'MAT-MID PRIME SPONSOR'
  COMPUTE PRIME = 2;
IF COUNTY = '087' AND (MCD = '005' OR MCD = '010' OR MCD = '013')
  COMPUTE  PRIMNAM = 'BROOK-CHOPT PRIME SPONSOR'
  COMPUTE PRIME = 3;
IF COUNTY = '087' AND (MCD = '015' OR MCD = '020')
  COMPUTE PRIMNAM = 'TUCKINA PRIME SPONSOR'
  COMPUTE PRIME = 4;
IF PRIME = 0 EXCLUDE
SORT PRIME COUNTY MCD
TOTAL PRIME PRIMNAM
REPORT
HEAD 'AGGREGATING SUMMARIES FOR PRIME AREAS' HEAD HEAD
ITEM
IF $TOTAL = 0
  ITEM 'TOWNSHIP: ' AREANAME ITEM;
IF $TOTAL = 1 EXCLUDE
IF $TOTAL = 2
  ITEM PRIMNAM
  ITEM 'PRIME CODE: ' F1.0 PRIME ITEM;
IF $TOTAL = 3
  ITEM 'STATE TOTAL' ITEM;
ITEM '     URBANIZED AND RURAL POPULATION'
ITEM C25.0 TAB1 (C) TAB1 (C)
```

Run 11

Printout:



```
AGGREGATING SUMMARIES FOR PRIME AREAS

TOWNSHIP: BERMUDA

         URBANIZED AND RURAL POPULATION
Total                           22285
Inside urbanized areas           5187
Rural                           11884

TOWNSHIP: CLOVER HILL

         URBANIZED AND RURAL POPULATION
Total                           36433
Inside urbanized areas          21461
Rural                           14972

TOWNSHIP: DALE

         URBANIZED AND RURAL POPULATION
Total                           25487
Inside urbanized areas          20877
Rural                            4610

BERMUDA-DALE PRIME SPONSOR
PRIME CODE: 1

         URBANIZED AND RURAL POPULATION
Total                           84205
Inside urbanized areas          47525
Rural                           31466

TOWNSHIP: MATOACA

         URBANIZED AND RURAL POPULATION
Total                           16092
Inside urbanized areas
Rural                           12468

TOWNSHIP: MIDLOTHIAN

         URBANIZED AND RURAL POPULATION
Total                           25837
Inside urbanized areas          21952
Rural                            3885

MAT-MID PRIME SPONSOR
PRIME CODE: 2

         URBANIZED AND RURAL POPULATION
Total                           41929
Inside urbanized areas          21952
Rural                           16353

TOWNSHIP: BROOKLAND

         URBANIZED AND RURAL POPULATION
Total                           33674
Inside urbanized areas          28606
Rural                            5068

TOWNSHIP: FAIRFIELD

         URBANIZED AND RURAL POPULATION
Total                           34631
Inside urbanized areas          33349
Rural                            1282

TOWNSHIP: THREE CHOPT

         URBANIZED AND RURAL POPULATION
Total                           29503
Inside urbanized areas          28275
Rural                            1228

BROOK-CHOPT PRIME SPONSOR
PRIME CODE: 3

         URBANIZED AND RURAL POPULATION
Total                           97808
Inside urbanized areas          90230
Rural                            7578

TOWNSHIP: TUCKAHOE

         URBANIZED AND RURAL POPULATION
Total                           37729
Inside urbanized areas          29803
Rural                            7926

TOWNSHIP: VARINA

         URBANIZED AND RURAL POPULATION
Total                           37385
Inside urbanized areas          23695
Rural                           13690

TUCKINA PRIME SPONSOR
PRIME CODE: 4

         URBANIZED AND RURAL POPULATION
Total                           75114
Inside urbanized areas          53498
Rural                           21616

STATE TOTAL

         URBANIZED AND RURAL POPULATION
Total                          299056
Inside urbanized areas         213205
Rural                           77013
```

```
FILEIN DICTNAME STF1
```

Summary Tape File 1 is the input file. The data dictionary provides the file characteristics and is the only parameter necessary with this command.

```
IF SUMRYLVL = '12' INCLUDE
```

Only records for minor civil divisions are included in the processing.

```
COMPUTE PRIME = 0
```

The data item, PRIME, will contain the value for the new geographic area. It is initialized to zero with each input record, so that records outside the PRIME areas can be identified and excluded. Generally, a new variable is set to zero only at the beginning of the run and its value is then replaced with each new computation.

```
IF COUNTY = '041' AND (MCD = '005' OR MCD = '010' OR MCD = '015')
     COMPUTE PRIMNAM = 'BERMUDA-DALE PRIME SPONSOR      '
     COMPUTE PRIME = 1;

IF COUNTY = '041' AND MCD = '025' OR MCD = '030')
     COMPUTE PRIMNAM = 'MAT-MID PRIME SPONSOR'
     COMPUTE PRIME = 2;

IF COUNTY = '087' AND (MCD = ' '005' OR MCD = '010' OR MCD = '013')
     COMPUTE PRIMNAM = 'BROOK-CHOPT PRIME SPONSOR'
     COMPUTE PRIME = 3;

IF COUNTY = '087' AND (MCD = '015' OR MCD = '020')
     COMPUTE PRIMNAM = 'TUCKINA PRIME SPONSOR'
     COMPUTE PRIME = 4;
```

If the record satisfies the geographic codes specified, the appropriate value of PRIME and PRIMNAM will be assigned.

```
IF PRIME = 0 EXCLUDE
```

Those records which fall outside of the PRIME areas will be excluded from further processing.

Run 11

```
SORT PRIME COUNTY MCD
```

The records will be sorted by the three geographic codes. Records will be TOTALed by PRIME and MCD tables are to be printed by counties.

```
TOTAL PRIME PRIMNAM
```

New summaries for each PRIME area will be created by aggregating across the MCD records. Both PRIME and PRIMNAM are referenced in the TOTAL command in order to be able to reference these keys during subtotal processing.

```
REPORT
HEAD 'AGGREGATING SUMMARIES FOR PRIME AREAS' HEAD HEAD
```

A report will be printed with a heading containing the literal string followed by two blank lines.

```
ITEM
IF $TOTAL = 0
ITEM 'TOWNSHIP:  ' AREANAME ITEM:
```

Tables for MCD's will be preceded by this line of print providing the area name and a blank line. The AREANAME field will be taken from the current input record ($TOTAL = 0) and is not associated with any summarized data.

```
IF $TOTAL = 1 EXCLUDE
```

Two keys, PRIME and PRIMNAM were referenced on the TOTAL command in order to be able to properly reference these data items (see TOTAL command discussion for further explanation). Only one of these subtotals is really needed, therefore the first level ($TOTAL = 1) is excluded.

```
IF $TOTAL = 2
ITEM PRIMNAM
ITEM 'PRIME CODE:  ' F1.0 PRIME ITEM;
```

A blank line will be produced followed by the assigned name for the area. Next the PRIME code will be printed for the new summaries, followed by a blank line.

```
IF $TOTAL = 3
ITEM 'STATE TOTAL' ITEM;
```

The state summary of all MCD's in PRIME areas will be identified with the literal string. A blank line will follow.

```
ITEM '     URBANIZED AND RURAL POPULATION'
ITEM C25.0 TAB1 (C) TAB1 (C)
```

A table heading, provided as a literal string, will be printed. Category labels and values for Table 1 will follow. Since these print lines are not qualified by an IF command, each input record ($TOTAL = 0) and each new summary ($TOTAL = 2 or $TOTAL = 3) will be processed through these commands.

Run 12

Report Using STF2 Data

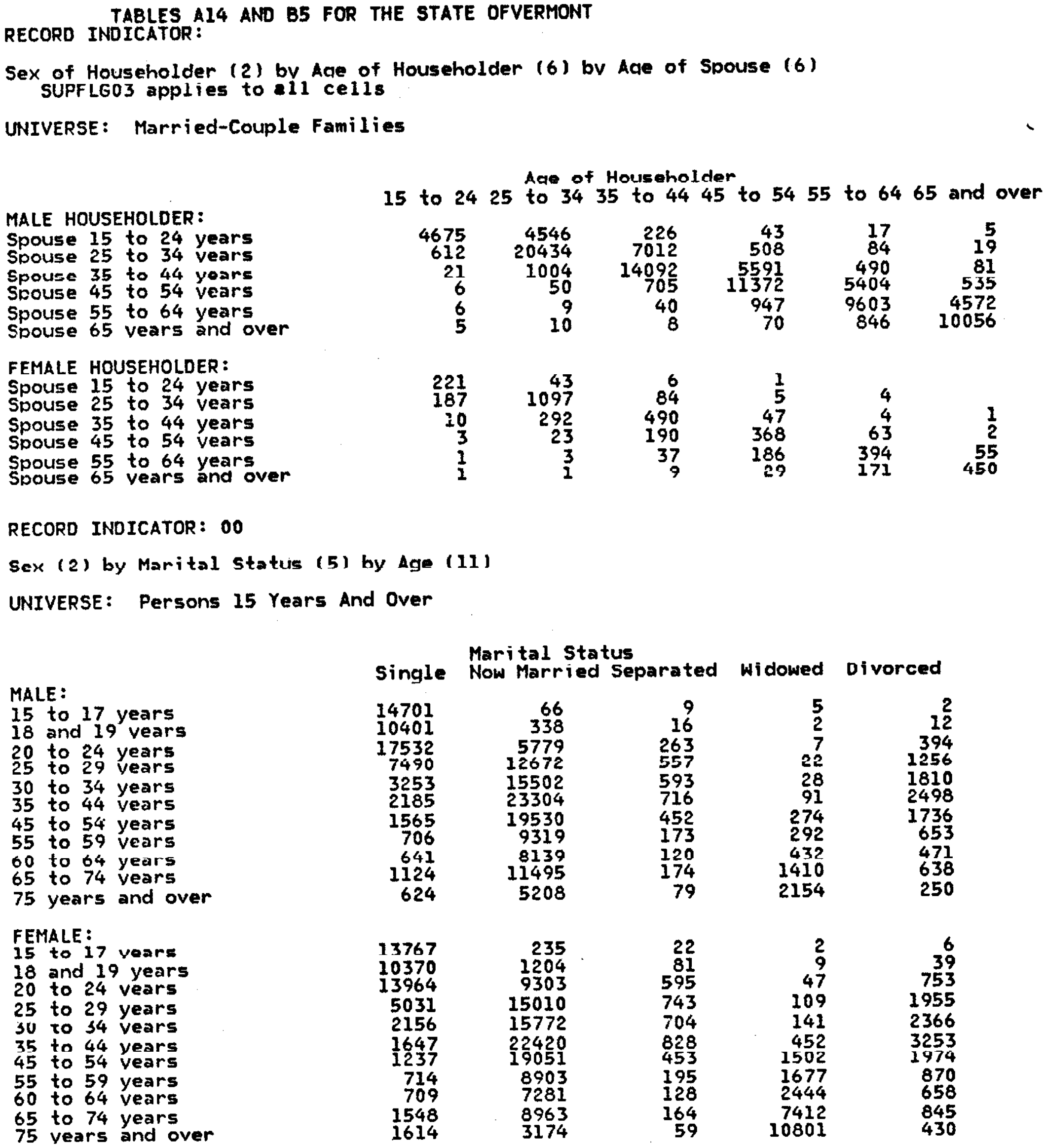
This example creates a report using both 'A' and 'B' records from the STF2A data file. Tables A14 and B5 are displayed for the state totals.

Commands:

```
FILEIN STF2 DEVICE DISK
*
* PRINT TABLE A14
*
REPORT HEAD @10.0 'TABLES A14 AND B5 FOR THE STATE OF' AREANAME
ITEM 'RECORD INDICATOR: ' RECOIND ITEM
IF RECOIND = '  '
ITEM H33.0 TABA14 H40.043 TABA14
ITEM X3.0 H40.083 TABA14 ITEM
ITEM U40.0 TABA14 ITEM ITEM
ITEM @45.0 'Age of Householder'
ITEM @33.0 '15 to 24 25 to 34 35 to 44 '
           '45 to 54 55 to 64 65 and over'
ITEM 'MALE HOUSEHOLDER:'
ITEM C30.0 TABA14 (1,1,C) TABA14 (1,*,C)
ITEM
ITEM 'FEMALE HOUSEHOLDER:'
ITEM C30.0 TABA14 (1,1,C) TABA14 (2,*,C)
ITEM ITEM ITEM ITEM;
* PRINT TABLE B5
IF RECOIND = '00'
ITEM H30.0 TABB5 H40.043 TABB5 ITEM
ITEM U40.0 TABB5 ITEM ITEM
ITEM @40.0 'Marital Status'
ITEM @30.0 '  Single  Now Married Separated  '
           'Widowed  Divorced'
ITEM 'MALE:'
ITEM C25.0 TABB5 (1,1,C) 5F11.0 TABB5 (1,*,C)
ITEM
ITEM 'FEMALE:'
ITEM C25.0 TABB5 (1,1,C) 5F11.0 TABB5 (2,*,C)
ITEM ITEM ITEM;
IF $INA > 5 STOP
```

Run 12

Printout:



```
                TABLES A14 AND B5 FOR THE STATE OFVERMONT
RECORD INDICATOR:

Sex of Householder (2) by Age of Householder (6) by Age of Spouse (6)
   SUPFLG03 applies to all cells

UNIVERSE:  Married-Couple Families

                                                 Age of Householder
                                 15 to 24 25 to 34 35 to 44 45 to 54 55 to 64 65 and over
MALE HOUSEHOLDER:
Spouse 15 to 24 years               4675     4546      226       43       17        5
Spouse 25 to 34 years                612    20434     7012      508       84       19
Spouse 35 to 44 years                 21     1004    14092     5591      490       81
Spouse 45 to 54 years                  6       50      705    11372     5404      535
Spouse 55 to 64 years                  6        9       40      947     9603     4572
Spouse 65 years and over               5       10        8       70      846    10056

FEMALE HOUSEHOLDER:
Spouse 15 to 24 years                221       43        6        1
Spouse 25 to 34 years                187     1097       84        5        4
Spouse 35 to 44 years                 10      292      490       47        4        1
Spouse 45 to 54 years                  3       23      190      368       63        2
Spouse 55 to 64 years                  1        3       37      186      394       55
Spouse 65 years and over               1        1        9       29      171      450

RECORD INDICATOR: 00

Sex (2) by Marital Status (5) by Age (11)

UNIVERSE:  Persons 15 Years And Over

                                      Marital Status
                              Single  Now Married Separated  Widowed  Divorced
MALE:
15 to 17 years                 14701        66         9         5         2
18 and 19 years                10401       338        16         2        12
20 to 24 years                 17532      5779       263         7       394
25 to 29 years                  7490     12672       557        22      1256
30 to 34 years                  3253     15502       593        28      1810
35 to 44 years                  2185     23304       716        91      2498
45 to 54 years                  1565     19530       452       274      1736
55 to 59 years                   706      9319       173       292       653
60 to 64 years                   641      8139       120       432       471
65 to 74 years                  1124     11495       174      1410       638
75 years and over                624      5208        79      2154       250

FEMALE:
15 to 17 years                 13767       235        22         2         6
18 and 19 years                10370      1204        81         9        39
20 to 24 years                 13964      9303       595        47       753
25 to 29 years                  5031     15010       743       109      1955
30 to 34 years                  2156     15772       704       141      2366
35 to 44 years                  1647     22420       828       452      3253
45 to 54 years                  1237     19051       453      1502      1974
55 to 59 years                   714      8903       195      1677       870
60 to 64 years                   709      7281       128      2444       658
65 to 74 years                  1548      8963       164      7412       845
75 years and over               1614      3174        59     10801       430
```

FILEIN STF2

This command identifies STF2 as the data file to be accessed.

REPORT HEAD @10.0 'TABLES A14 AND B5 FOR THE STATE OF' AREANAME

These 2 commands indicate that a report is to be printed and provide the first page heading line.

ITEM 'RECORD INDICATOR: ' RECORD ITEM

These ITEM commands label the record indicator field, print the RECOIND field and print a blank line following the record indicator.

IF RECOIND = ' '

This IF statement tests the record indicator field for a value of spaces. Spaces in this field indicate a 'A' record. The next 19 ITEM commands list data from the 'A' record and are executed only when the RECOIND field contains spaces.

ITEM H33.0 TABA14 H40.043 TABA14

This ITEM command lists heading labels from Table A14. The H33.0 lists the first 33 characters of the heading label. The H40.043 lists the first continuation label of the heading. Due to the spacing included in the labels (the data dictionary labels contain exactly what is printed in the data dictionary documentation listing) only parts of the first 2 40 character label records are printed. The H33.0 access the first 33 characters of the heading labels. The H40.043 retrieves characters 43-80 of the label which is really 37 characters of the first continuation label.

ITEM X3.0 H40.083 TABA14 ITEM

This command lists 40 more characters of the Table A14 heading label and prints a blank line following the last label line.

ITEM U40.0 TABA14 ITEM ITEM

This command line lists the universe description for Table A14, following it with 2 blank lines.

Run 12

```
ITEM @45.0 'Age of Householder'
ITEM @33.0 '15 to 24 25 to 34 35 to 44'
     '45 to 54 55 to 64 65 AND OVER'
```

Table A14 is to be printed as a 3 dimensional table (Sex of Householder by Age of Householder by Age of Spouse). The data cells in the table are 9 characters wide and are to be printed in the 9 character default format. The column labels for these data columns must fit within the 9 character width. Based on reviewing the data dictionary documentation it is determined that the labels available for the column headings are longer than 9 characters and therefore for printing Table A14 in this format are not useable. The 2 ITEM commands listed above provide column headings using literal information instead of using data dictionary labels.

```
ITEM 'MALE HOUSEHOLDER:'
```

This command is to label the first part of Table A14 as the 'Male Householder' portion.

```
ITEM C30.0 TABA14 (1, 1, C) TABA14 (1, *, C)
```

This command prints the Male Householder portion of Table A14. C30.0 TABA14 (1, 1, C) prints the 3rd level category labels in a 30 character wide column. These labels are for the Age of Spouse dimension. The TABA14 (1, *, C) requests data for Table A14. The data cells requested are for the first category of the first dimension, (1, *, C) or the Male Householder portion. Within the Male Householder portion of the table, the Age of Householder dimension or the second dimension, (1, *, C) is listed across the page, establishing 6 columns of data. The last dimension, Age of Spouse is listed in the columns (1, *, C). This prints the 36 cells of data for Male Householder in a 6 by 6 table.

```
ITEM
ITEM 'FEMALE HOUSEHOLDER:'
ITEM C30.0 TABA14 (1, 1, C) TABA14 (2, *, C)
```

These 3 commands provide a blank line following the male portion of the table and then label and print the female portion of the table in the same way as the male section

```
ITEM ITEM ITEM ITEM;
```

Four lines of blanks are printed at the end of TABA14. A semi-colon terminates the command list which began with the IF RECOIND = ' ' test. What follows is the selection and printing instruction for the 'B' record.

```
IF RECOIND = '00'
```

This IF command selects records based on the race/ethnicity record indicator field, selecting only '00' (for Total Population).

```
ITEM H30.0 TABB5 H40.043 TABB5 ITEM
```

These commands list the table heading from Table B5 followed by a blank line.

```
ITEM U40.0 TABB5 ITEM ITEM
```

These commands list the universe label from Table B5 followed by 2 blank lines.

```
ITEM @40.0 'Marital Status'
ITEM @30.0 'Single Now Married Separated'
           'Widowed Divorced'
```

These ITEM commands provide column headings for the table to be printed similar to the process used for TABA14 printed earlier. The existing data dictionary labels cannot be printed meaningfully in 9 character wide column headings so new labels are provided to meet the desired table format.

```
ITEM 'MALE:'
```

The first dimension of TABB5 is sex and the data for males are to be printed first. This command labels this portion of the table as 'MALE:'.

```
ITEM C25.0 TABB5 (1, 1, C) 5F11.0 TABB5 (1, *, C)
```

This command lists the labels and data from Table B5 for males. The 3rd dimension category labels of age are listed in a column at the left of the page. Next to these labels are the data for males (1, *, C) formatted with the 5 marital status categories going across the page (1 *, C) and the 11 age categories listed in columns under the marital status categories (1, *, C).

```
ITEM
ITEM 'FEMALE:'
```

This command provides the label 'FEMALE:'. The data for females from Table B5 is to be listed next.

```
ITEM C25.0 TABB5 (1, 1, C) 5F11.0 TABB5 (2, *, C)
```

This command lists the data for females from Table B5. The Age labels are listed first followed by the data for females (2, *, C) in a 2 dimentional table format. Marital status is listed across the page (2, *, C) and Age is listed in columns under the marital status categories (2, *, C).

Run 12

```
ITEM ITEM ITEM;
```

Three blank lines are printed below TABB5

```
IF $INA > 5 STOP
```

This command causes processing to be terminated after the 5th record. This has the effect of printing only the state summaries which appear first on the file.

CHAPTER IV

The CENSPAC Documentor

The CENSPAC Documentor is a subsystem of CENSPAC that assists users in preparing file documentation. The Documentor creates both printed data dictionaries and machine-readable data dictionary files for input to the CENSPAC system.

The Documentor creates a direct access, relative indexed file called the master data dictionary. The master data dictionary acts as a library and may contain multiple data dictionary files, such as STF1, STF2, etc., each of which is referenced by a specific dictionary name. The CENSPAC system uses the data dictionaries to obtain labeling information, data item descriptions, data item locations, and other file specifications needed for a run. After initiating a master file on a disk, the user can add or replace data dictionaries to the master file. These dictionaries can then be accessed by their data dictionary names.

The Documentor enters a data dictionary in the master file by reading data description statements, verifying the syntax of their format and content, and building entries in a temporary dictionary file. If, when all of the data description statements are read, no errors have been detected, the temporary data dictionary file is read and the information is used to replace or add a dictionary to the master data dictionary file. A printed report of the dictionary is automatically produced. If errors are detected, processing ceases and no dictionary is created.

The Documentor also performs various housekeeping functions on the master dictionary file. It can delete data dictionaries and compress the master file to eliminate deleted entries. (This compress function is not available in May or August 1980 versions of CENSPAC). The Documentor can also print dictionary documentation in several formats.

Systems Requirements

The Documentor was developed on an IBM OS/VS system and is written in standard ANSI COBOL. It requires a maximum approximately 120K characters of main storage but with program overlays can execute in 80K. The amount of disk file space for the intermediate work files and data dictionary files is dependent on the number of data files to be documented. The space requirements for particular dictionaries are shown in Appendix 5.

Data Dictionary File Description

The master data dictionary file can store up to 1,000 data dictionaries, provided enough disk space is allocated to contain all the entries. Within the master file, the Documentor logically stores separate entries for the individual data dictionaries, such as STF1.

When an individual data dictionary is re-processed to make changes, new space is taken up to accommodate the new version of the data dictionary. The space previously occupied by this data dictionary becomes unused and is also unavailable for further use temporarily. Because of this, the master data dictionary will have to be 'compressed' occasionally to make this unused space available.

Documentor Command Syntax

The Documentor control statements have format requirements different from the CENSPAC control statements. Commands must start in column one. For the operands that follow a command, the actual card column location is not important, but the sequence of fields is significant.

Some commands are abbreviated and are referenced only by their single-character notation. If consecutive commands are the same type, the command notation may be omitted from column one, but column one must remain blank. The operands that follow are then associated with the same type of command as the previous one.

Each operand has a default value which is assigned if the operand is skipped. In most cases, the default value is the value of the same operand in the previous command of the same type. If only some operands of a command are to default, commas must be used to indicate that an operand has been skipped. Except where commas are needed to identify omitted parameters, either commas or blanks may be used as field delimiters.

Following the operands, most control statements will contain a label that begins with a period (.). A maximum of 40 characters of labeling information can appear on one card image. Continuation labels, each a maximum of 40 characters, can be provided on subsequent cards with a colon (:) as the leading character. Sample Documentor statements and documentation are shown at the end of this chapter.

When an asterisk (*) appears as the first non-blank character in a control statement, the Documentor identifies the line as a comment. If the asterisk is not in column one, the Documentor enters the comment (first 40 characters only) in the data dictionary. If the asterisk is in column one, the Documentor treats it as a local comment and will not include it in the data dictionary.

Documentor Logic

The CENSPAC Documentor is designed to list dictionary documentation and to create new data dictionaries. A user cannot modify individual items or tables within an existing data dictionary without recreating the entire dictionary.

The commands required to list a data dictionary are:

```
DOCUMENT
END
```

Every documentor program must begin with the DOCUMENT command and end with the END command. If a user only wants to list the documentation of an existing data dictionary, then the only required commands are DOCUMENT and END. In the DOCUMENT command, certain keyworded operands must be specified or the documentor will assume a data dictionary is being created. See the DOCUMENT command for more information.

Machine-readable Documentation Distribution

The Census Bureau plans to document all 1980 Summary Tape Files (STFs), most of the 1970 Census summary files, and many other census public use data files, in data dictionary form. Data dictionary definition statements, the input statements to the Documentor, are available on magnetic tape for particular data dictionaries or for a number of data dictionaries. These data definition statements are prepared using both upper and lower case characters in the labels and text. For operating systems that can process only uppercase letters, modification of the operating system control language will be necessary.

A thorough understanding of the Documentor control statements is not necessary for those who want to use data dictionaries with CENSPAC or another software system. Only the DOCUMENT command needs to be understood.

Special Data Dictionaries

The CENSPAC system has 2 special data dictionaries stored in the master data dictionary. These are identified as 'SYSTEXT' and 'MACLIB'. Both of these data dictionaries are included with the CENSPAC system tape and should be included in any new master data dictionary file. SYSTEXT contains the text for both the CENSPAC error messages and the HELP command. If SYSTEXT is not available in a master data dictionary the HELP command will not function and no descriptive text will be displayed to assist in CENSPAC error resolution. If MACLIB is not available the CENSPAC library function will not operate. If corrections or additions are made to either of these data dictionaries, all the data definition statements for the entire data dictionary (SYSTEXT or MACLIB) including the changes, must be reprocessed by the Documentor program in order to re-build the complete corrected data dictionary. See Chapter 2, CENSPAC Command Library for more information on the MACLIB library function. In the May and August 1980 versions of CENSPAC, MACLIB is not available and SYSTEXT is named ERRTEXT.

Documentor Commands

The individual Documentor commands are presented on the following pages. Each command is discussed and a brief example is provided. A complete example derived from items included in the STF1 file is presented after this command description. This comprehensive example should provide a basis for learning enough about the Documentor to get started in preparing file documentation in CENSPAC data dictionary form. For users who will not be preparing entire data dictionaries, but who will be accessing the data dictionaries provided by the Census Bureau, a review of the DOCUMENT command function should be all that is necessary from this section of the manual.

CATEGORY

This command is required when documenting a table. It must follow the STRATIFIER command. The CATEGORY command identifies the individual categories within each stratifier. There must be one CATEGORY command for each category specified by the STRATIFIER command. The first CATEGORY command identifies the first category of the stratifier; the second CATEGORY command identifies the second category, and so on.

Syntax: C stratifier-number .label

C: This is required in the first CATEGORY command of a stratifier series. It is the single-letter abbreviation that identifies the CATEGORY command.

stratifier-number: This operand is required in the first CATEGORY command. It is a 1-digit number that links the category label with a particular stratifier. If the stratifier-number is omitted in subsequent CATEGORY commands, the default value is the last stratifier-number supplied.

.label: The period (.) separates the stratifier-number from the label. Up to 40 characters of label are allowed, and continuation lines may be supplied. A colon (:) must be the leading character of every continuation line. (The 40 characters following the period or colon are used for the label; all information beyond the 40 characters is ignored. No errors are detected for violation of the 40 character limit.)

Example

```
T TABSEX 5 63 N N
H .Sex by Marital Status
U .Universe - Persons
S 2 .Sex
S 5 .Marital Status
C 1 .Male
    .Female
C 2 .Single
    .Now Married, except Separated
    .Separated
    .Widowed
    .Divorced
```

The CATEGORY commands follow the TABLE and STRATIFIER commands (see the T and S commands for more information on these commands.) The 'C' is the single-letter abbreviation that references the CATEGORY command. The '1' links the first CATEGORY command with the first STRATIFIER command. The first CATEGORY command identifies the first category of the stratifier 'Sex' as Male. The second CATEGORY command identifies the second category of the stratifier 'Sex' as Female. Note that the 'C' and the '1' do not have to appear in the second CATEGORY command. The Documentor defaults to the definitions provided in the previous CATEGORY command.

The '2' in the third CATEGORY command links this CATEGORY command to the second stratifier, 'Marital Status'. Five CATEGORY commands identify the five categories of marital status. The 'C' and the '2' do not have to appear after they are identified in the third CATEGORY command. The Documentor defaults to the CATEGORY command in which this information was last supplied.

DATA

The DATA command is required when data items are entered in a data dictionary. The DATA command describes an individual data item in a data file. Commas must be used to delineate the operands 'length', 'begin', 'type' and 'category' if they are to default.

Syntax: D dataname length begin type category scale .label

D: This is required. It is the single-letter abbreviation that identifies the DATA command.

dataname: This operand is required. It is a user-supplied name for the data item being documented. The dataname may be two to eight characters in length, and may consist of alphabetic and/or numeric characters. The first character of all user-defined data names must be alphabetic.

length: This operand is optional. It is a 1-5 digit value identifying the length in characters of the data item. If this operand is omitted, the length of the preceding data item is the default value. If length has not been previously identified, a value of 1 is assumed.

begin: This operand is optional. It is a 1-5 digit value identifying the beginning location of the data item. If omitted, the beginning location is calculated by the documentor as equal to the beginning position of the previous data item plus the length of the previous data item. If no previous beginning position was established, a value of 1 is assumed.

type: This operand is optional. It identifies data type. The following types are allowed:

A: alphanumeric character
N: numeric character
B: numeric binary

If this operand is omitted, the type of the preceding data item is assumed. If there is no preceding data item, the default type is A.

category: This operand is optional. It identifies how a data item will be used. The values may be:

N: Numeric use. These data items may be added, subtracted, averaged, or used in other arithmetic operations.

C: Code use. Identifies data items that are not used numerically, such as geographic codes, record type codes, or alphabetic data.

If this operand is omitted, the category of the preceding data item is used. If there is no preceding data item, the default category is C.

scale: This operand is optional. It is a 1 digit number that identifies the scaling factor for insertion of a decimal point. A minus sign in front of the number identifies a decimal point, and the number identifies the decimal places. A -2 in this field would indicate a data field that contains 2 implied decimal places. If no minus sign appears, the data item is identified as a whole number raised to a power of ten equal to the value of the scaling factor provided. A +3 in this field would indicate a data field that is reported in units of 1000. If this operand is omitted, the default value is 0.

.label: This operand is optional. The period (.) separates the operands from the label. Up to 40 characters of label may be supplied. Continuation lines are allowed, and a colon (:) must be the first character in each continuation line. (The 40 characters following the period or colon are used for the label; all information beyond the 40 characters is ignored. No errors are detected for violation of the 40 character limit.)

Example

```
D FILEID 5 1 A C .Identifier for summary file
                 :i.e., STF1A, STF1B, etc.
```

This command describes the variable FILEID, which has a length of 5, begins in column 1, and is an alphanumeric code variable. It provides a label and a continuation line. The scale operand was not included; it defaults to zero.

DOCUMENT

The DOCUMENT command is required and must be the first command in a Document run. It signals the beginning of a set of commands to create or list a data dictionary.

To list a data dictionary, the user needs only the DOCUMENT and END commands. In the DOCUMENT command, the name of the data dictionary to be listed is entered as 'dictname'. The operand LIST must also be specified. If the operands LIST and status are omitted, the documentor will assume a data dictionary is being created (or an existing one is being updated or replaced) and it will erase any data dictionary with that dictname from the master file.

Syntax: DOCUMENT dictname status list-type LIST DT DC
device lrecl blksize segments

dictname: This operand is required. It is a 2-8 character name that identifies the data dictionary being created or listed.

status: This operand is optional. It is used when creating a master data dictionary file, or when deleting or reactivating data dictionaries in the master file. The following options are available:

NEW: This operand creates the master data dictionary file. Caution: NEW must be specified only for the first of the data dictionaries in the master data dictionary file. Specifying NEW will clear and initialize the entire master file. This will delete all previous data dictionary entries in the file. On systems where the Documentor space allocation procedure will not work as designed (see installation instructions in Chapter V, primarily non IBM OS/VS and non-UNIVAC EXEC-8), the NEW operand can be followed by a number enclosed in parentheses, i.e. (100), to indicate the number of dictionary file blocks that are to be allocated and initialized. If this operand is omitted, the Documentor program will allocate and initialize as much space as is available to it, or 800 records (whichever is less).

DEL: This operand deletes the data dictionary identified as dictname. The entries for dictname are not actually removed from the master data dictionary, but are marked for deletion. The deletion is not completed until a compress of the data dictionary file is activated.

ACT: This operand reactivates a data dictionary after it has been marked for deletion. This operand clears the deletion markers from dictname so that entries for dictname will no longer be marked for deletion.

COMP: This operand compresses the data dictionary file and makes space occupied by deleted data dictionaries available for use. All data dictionaries marked for deletion will be deleted.

list-type: This operand is optional. It specifies the kind of dictionary documentation that is to be printed. The following print option is available:

LONG: This option prints a detailed data dictionary listing that identifies every cell in all tables and prints the beginning character location of each cell.

If this operand is omitted a shorter listing will be prepared. This shorter listing provides for one iteration of all category labels within tables but does not repeat the category labels for tables with more than one dimension. The default listing prints an abbreviated table structure in which category labels are not repeated. For example, in a sex by age table, the separate age categories would be listed for the first category, male, but not for the second category, female. The LONG option will cause the printed data dictionary to list every cell on the tables.

LIST: This operand is optional. With it, a user can request a listing of the documentation of an existing data dictionary. If neither LIST nor status appear with the DOCUMENT command, the Documentor assumes a data dictionary is being created and any existing data dictionary with the dictname specified will be erased. The LIST operand cannot be used when creating a data dictionary because a report is listed automatically if no errors are detected.

DT, DC: These operands are print options for the formatted data dictionary listing. If DT is coded then the data type column will appear for data dictionary items. If DC is coded the data category column will be printed.

device: This operand is optional. It identifies the type of device on which the documented data file is recorded. Valid device indicators are:

TAPE:	standard labeled magnetic tape
TAPU:	unlabeled magnetic tape
DISK:	sequential disk
CARD:	card file

If this operand is omitted, the default value is TAPE. Note: on many systems, including IBM OS/VS and UNIVAC EXEC-8, (when using SDF data) the device information is not used but is determined at run time by the operating system.

lrecl: This operand is required. It is a 1-5 digit number with a range of 1 through 32,767 that identifies the record length of the file being documented. The default value for this operand is the record length calculated by the Documentor on the basis of data item location and length.

blksize: This operand is optional. It is a 1-5 digit number that specifies the blocksize in characters of the file being documented. The value must be within the range of 1 through 32,767. The default value is the lrecl value.

segments: This operand is optional. Usually it will apply only to census summary files. It is a 1 or 2 digit number that identifies the maximum number of segments in a census logical record. For the 1970 Census First Count summary file, for example, the value would be 2, indicating that two record segments comprise one census logical record.

Examples

```
DOCUMENT STF1 LIST LONG DT DC
END
```

This DOCUMENT command requests a listing of an existing data dictionary within the master data dictionary file. The listing being requested is for the STF1 data dictionary. The listing will contain all available data dictionary information about the STF1 file including a cell-by-cell listing of the tables (LONG) and presentation of the data type (DT) and data category (DC) attributes. The END statement is the only other Documentor program command in this run.

```
DOCUMENT CITYDATA LONG DT DC DISK 132 13200
```

(data definition statements follow)

This is an example of the first command that will create a data dictionary called CITYDATA. The CITYDATA file is on a disk, and has a logical record length of 132 characters and a block size of 13,200 characters. A full listing (LONG) of the CITYDATA data dictionary is requested. The data type and data category attributes of the data items will be listed.

```
DOCUMENT CITYDATA DEL
END
```

In this example, the data dictionary CITYDATA is to be deleted from the master data dictionary file. The END command indicates that there are no other Documentor statements for this request. The actual deletion from the data dictionary file does not take effect until the file is compressed.

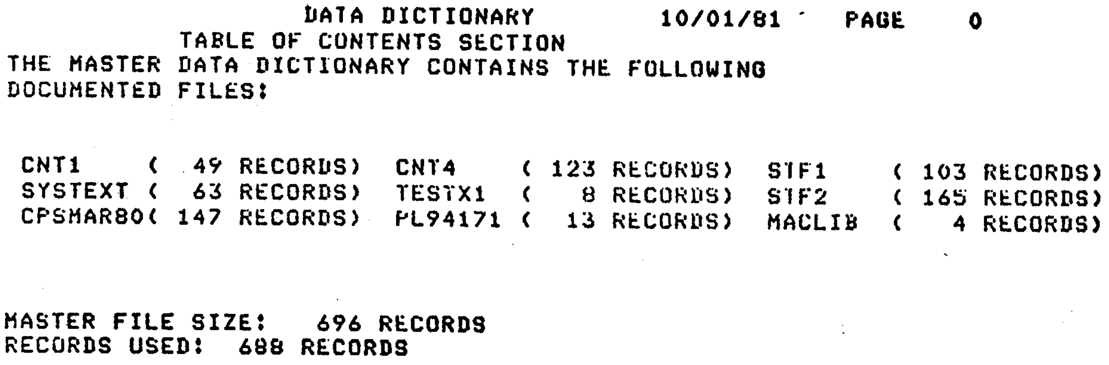
```
DOCUMENT COMP
END
```

These documentor statements will compress the data dictionary file. This procedure should be executed when the documentor program recognizes that there is an insufficient number of data dictionary blocks to create a data dictionary and prints the following message:

> "not enough records available for this request---execute compress function".

Additionally this problem can be anticipated by observing the values of the records used field which appears on page Ø of the data dictionary listing, (see sample page Ø which follows). When the number records used approaches the master file size (also on page Ø), a compress of the data dictionary file is necessary. The following example depicts a data dictionary file which is almost full. If, after a compress has been completed, the number of records used is almost as large as the master file size, then the master data dictionary file is full. The only way to expand the master data dictionary file is to delete the file and re-establish the file with more space. All desired dictionaries will have to be processed again to make them available in the new dictionary file.

Example



```
                    DATA DICTIONARY          10/01/81    PAGE     0
              TABLE OF CONTENTS SECTION
 THE MASTER DATA DICTIONARY CONTAINS THE FOLLOWING
 DOCUMENTED FILES:


  CNT1     (  49 RECORDS)  CNT4     ( 123 RECORDS)  STF1     ( 103 RECORDS)
  SYSTEXT  (  63 RECORDS)  TESTX1   (   8 RECORDS)  STF2     ( 165 RECORDS)
  CPSMAR80( 147 RECORDS)  PL94171  (  13 RECORDS)  MACLIB   (   4 RECORDS)


 MASTER FILE SIZE:   696 RECORDS
 RECORDS USED:  688 RECORDS
```

END

The END command is required. It terminates a set of data dictionary statements.

Syntax: END

The END command has no operands.

FOOTNOTE

This command is optional and is for documentation purposes only. The FOOTNOTE command supplies the text for footnotes. The FOOTNOTE command is placed at the end of the file description statements and is followed by a list of all footnote numbers and their labels. Only one FOOTNOTE command is needed to list all footnotes associated with the dictionary.

Syntax: FOOTNOTE

footnote-number: This operand is required. It links the FOOTNOTE text with the number specified in the NOTE command.

.label: This operand is required. The label provides the text for the footnotes referenced in NOTE commands. The period (.) separates the footnote-number from the label. Up to 40 characters of descriptive text are allowed. Continuation lines may be provided as needed. A colon (:) must be the leading character in every continuation line.

Example

```
1.  .Variable was calculated by the
    :Census Bureau

2.  .Includes all year-round housing units
```

To link the first note to a particular table, the following should be coded;

```
T   TAB3
N   1
S   1.HOUSEHOLD
```

FORMAT

This command is required when a data dictionary is being created. It begins the data item descriptions for a particular record format. If there is only one record format in file, only one FORMAT command is needed. If there is more than one record format in a file, more than one FORMAT statement will be needed. All data item descriptions for one record are entered after the FORMAT command. The appearance of another FORMAT command terminates one record format and initializes the beginning position of another. Printed documentation for the new record will begin on a new page.

Syntax: FORMAT formatcode

formatcode: This operand is required. It is a 1-2 character alphanumeric code that identifies the record format.

Example

FORMAT A1

The data definition statements to follow will describe a record layout labeled A1. The printed documentation will identify the record format at the top of each page. Any additional format descriptions will always be initiated on a new print page.

GROUP

This command is optional. It allows the user to insert group labels or caption lines in a table description.

Syntax: G .label

G: This is required. It is the single-letter abbreviation that references the GROUP command.

.label: The period (.) separates the command from the label. The label may be up to 40 characters in length, and continuation lines are allowed. A colon (:) must be the first character in any continuation line.

Example

```
G .Asian and Pacific Islanders
```

The above command might be used to label a number of related ethnic groups.

NOTE

The NOTE command links footnotes to specific tables.

Syntax: N footnote1 footnote2

footnote1: This operand is required and follows the TABLE command. It specifies the footnote numbers associated with the individual tables. Up to four footnote numbers may be entered. The text for the footnotes is supplied at the end of a documentor run in the FOOTNOTE command.

See FOOTNOTE command example.

STRATIFIER

This command is required when documenting a table. It must follow the TABLE command and precede the CATEGORY commands. The STRATIFIER commands identify the number of dimensions in a table and the number of categories in each dimension. The number of STRATIFIER commands that follow a TABLE command determine the number of stratifiers in the table. The operand 'categories' identifies the number of categories in each stratifier.

Syntax: S categories .label

S: This is required. It is the single-letter abbreviation that references the STRATIFIER command.

categories: This operand is required. It is a 1-4 digit number that identifies the number of categories within the stratifier. For example, the stratifier 'sex' contains two categories, male and female.

.label: This operand is required. The period (.) separates the 'categories' operand from the label. Up to 40 characters of label are allowed, and continuation lines may be supplied. A colon (:) must be the leading character in any continuation line.

Example

```
T TABSEX 5 63 N N
H .Sex by Marital Status
U .Universe - Persons
N 1
S 2 .Sex
S 5 .Marital Status
C 1 .Male
    .Female
C 2 .Single
    .Now Married, except Separated
    .Separated
    .Widowed
    .Divorced
```

The STRATIFIER commands follow the TABLE command (T). The 'S' is the single-letter abbreviation that references the STRATIFIER command. The first stratifier command identifies the first stratifier of the table. The '2' identifies the number of categories in the first stratifier, labeled 'sex'. The second STRATIFIER identifies the second stratifier of the table. The second stratifier has 5 categories and is labeled 'Marital Status'.

SUPPRESSION

This command is optional. It identifies the type of suppression that may occur in a summary table from the 1970 census files.

Syntax: X type

X: This is required. It is the single-letter abbreviation that references the SUPPRESSION command.

type: This operand is required. It identifies the type of suppression and is used by CENSPAC in processing the 1970 census files to indicate the presence or absence of suppression. On the 1980 summary tape files, suppression is controlled by separate flag fields in the individual data records. The valid suppression codes for 1970 are SX, SA, SO, S1, and S2. Only one of these codes may be used in a SUPPRESSION command. For more information on these suppression codes, see the 1970 Census User's Guide.

TABLE

This command is required when documenting a table. The TABLE command identifies a table and its characteristics. The TABLE command must be followed by STRATIFIER and CATEGORY commands.

Syntax: T tablename length begin type category scale

T: This is required. It is the single-letter abbreviation that references the TABLE command.

tablename: This operand is required. It identifies the table. The operand must be in the form TABxxxxx, where the first three characters must be TAB and the remaining 1-5 characters are alphanumeric.

length: This operand is optional. It is a 1-5 digit number that identifies the size of the cells within a table. All cells in the table must be the same size. If this operand is omitted, the length specified for the preceding table is used. If no length was previously specified, a value of 1 is assumed.

begin: This operand is optional. It is a 1-5 digit number identifying the beginning location in the data record of the first cell within a table. If this operand is omitted, the beginning location will be calculated by the documentor to be the beginning location of the last table or data item, plus the cell size multiplied by the number of cells in the previous table. If no previous beginning position was established, a value of 1 is assumed.

type: This operand is optional. It identifies data type. The following types are followed:

A: alphanumeric character
N: numeric character
B: numeric binary

If this operand is omitted, the type of the preceding data item is assumed. If there is no preceding data item, the default type is A.

category: This operand is optional. It identifies how the cells within the table will be used. All cells must be the same type. The values may be:

N: Numeric use. These data items may be added, subtracted, averaged, or used in other arithmetic operations.

C: Code use. Identifies data items that are not used numerically, such as geographic codes, record type codes, or alphabetic data.

If this operand is omitted, the category of the preceding data item is used. If there is no preceding data item, the default category is C.

scale: This operand is optional. It is a 1 digit number that identifies the scaling factor for insertion of a decimal point. A minus sign in front of the number identifies a decimal point, and the number identifies the decimal places. If no minus sign appears, the data item is identified as a whole number, raised to the power of ten equal to the value of the scaling factor. If this operand is omitted, the default value is 0.

Example

```
T TAB1 5 63 N N
```

A table is identified as TAB1. Each cell is length 5, and the first cell starts in column 63. Both type and category are numeric. Note: no scale was specified so that it defaults to zero. See the CATEGORY command for a full example of the table command set.

HEADING

This command is required for table definitions. It is used to provide a descriptive title for a table. The command is followed by 40 characters of label.

Syntax: H .label

H: This is required. It is the single-letter abbreviation that references the HEADING command.

.label: This operand is required. The period (.) separates the command from the label. Up to 40 characters of label may be supplied, and continuation lines are allowed. The first character in every continuation line must be a colon (:).

Example

```
H .Sex by Marital Status
```

TEXT

TEXT provides a free-form narrative description of a data file which will appear at the beginning of the printed documentation. TEXT is terminated by the next command.

Syntax: TEXT

The TEXT command has no operands.

Example

```
TEXT

     This text describes the sort sequence and geographic coverage of the
data file.

END
```

UNIVERSE

This command is optional. It provides the universe description for a table.

Syntax: U .label

U: This is required. It is the single-letter abbreviation that references the UNIVERSE command.

.label: This operand is required. The period (.) separates the UNIVERSE command from the label. Up to 40 characters of label may be supplied, and continuation lines are allowed. A colon (:) must be the leading character in any continuation line.

Example

```
U .Universe - All Persons
```

VLABEL and MVLABEL

This command is optional. The VLABEL command documents the value categories allowed for a data item and provides a description of each category. Several value label commands may be associated with a single data item.

A related command is MVLABEL, which documents missing value categories. The command abbreviation is M, and it has the same operands as the VLABEL command.

Syntax: V valuecode .label

V: This is required in the first VLABEL command of a series. It is the single-letter abbreviation that references the VLABEL command.

valuecode: This operand is required. It is a 1-8 character code or numeric value representing one of the values for a data item. If no valuecode is specified, valuecode defaults to blank.

.label: This operand is optional. The period (.) separates the VLABEL command from the label. Up to 40 characters of label may be supplied, and continuation lines may be used as needed. A colon (:) must be the leading character in any continuation lines. If no label is specified, the label will default to the value supplied in the valuecode operand.

Example

```
V 01 .United States
  02 .State
  03 .County
```

In the above example, the VLABEL command would immediately follow a DATA command. The VLABEL command identifies the acceptable values and the definitions of that data item.

Sample Data Dictionary Definition Statements

The statements provided here is part of the STF1 data dictionary definition statements. The printed data dictionary produced using these statements is provided at the end of this listing.

```
DOCUMENT STF1         1638 16380 2
FORMAT A
D FILEID 5 1 A C .Identifier for summary file,
                 :i.e., STF 1A, STF 1B, etc.
                 :STF1A-Summary Tape File 1A
                 :STF1B-Summary Tape File 1B
                 :STF1C-Summary Tape File 1C
D RECTYP 4 .Identifies multiple logical
           :record formats on STF files
           :It is blank if only one
           :format is present.  (Blank
           :on STF 1)
D SUMRYLVL 2,,A C .Identifies Geographic level
            :of current record
V 01 .United States
  02 .Region
  03 .Division
  04 .State or State equivalent
  05 .SCSA
  06 .SCSA/State
  07 .SMSA
  08 .SMSA/State
  09 .Urbanized Area
  10 .Urbanized Area/State
  11 .State/County or County equivalent
  12 .State/County/MCD (CCD)
  13 .State/County/MCD (CCD)/
     :Place
  14 .State/County/MCD (CCD)/
     :Place/Tract (BNA)
  15 .State/County/MCD (CCD)/
     :Place/Tract (BNA)/BG
V 16 .State/County/MCD (CCD)/
     :Place/Tract (BNA)/ED
V 17 .State/SMSA/County
  18 .State/SMSA/County/MCD (CCD)
  19 .State/SMSA/County/MCD (CCD)/
     :Place
  20 .State/SMSA/County/MCD (CCD)/
     :Place/Tract (BNA)
V 21 .State/SMSA/County/MCD (CCD)/
     :Place/Tract (BNA)/Block
V 22 .State/SMSA/County/MCD (CCD)/
     :Place/Tract (BNA)/ED
V 23 .State/SMSA/County/Place
  24 .State/SMSA/County/Place/
     :Tract (BNA)
  25 .State/SMSA/County/Place/
     :Tract (BNA)/Block
V 26 .State/SMSA/County/Place/
     :Tract (BNA)/ED
V 27 .State/Place
  28 .State/MCD Sequence Number
  29 .Indian Reservation (ANV)
  30 .Indian Reservation (ANV)/State
  31 .Indian Reservation (ANV)/State/
     :County
  32 .State/SMSA/County/Tract (BNA)
  33 .State/Congressional District
D URBARURL 2,,A C .Urban and Rural component
            :Only 01 and 08 are
            :indicated on STF 1.
V 00 .Not Urban and Rural component
  01 .Urban
  02 .  Inside urbanized areas
  03 .    Central Cities
```

```
   04 .    Urban fringe
   05 .  Outside urbanized areas
   06 .    Places of 10,000 or more
   07 .    Places of 2500 to 10,000
   08 .Rural
   09 .  Places of 1000 to 2500
   10 .  Other rural
   11 .  Farm
D SMSACOM 2 .Inside and Outside SMSA's
            :Component
            :Note:  Not applicable to STF 1,
            :field is blank.
V 00 .Not Inside and Outside SMSA
     :Component
   01 .Inside SMSA's
   02 .  Urban
   03 .    Central cities
   04 .    Not in Central cities
   05 .  Rural
   06 .Outside SMSA's
   07 .  Urban
   08 .  Rural
D RACESPAN 2 .Identifies Race/Spanish
             :Origin Group
             :Note:  Not applicable to STF 1,
             :field is blank.
D ANCESTRY 3 .Identifies specific ancestry
             :group
             :Note:  Not applicable to STF 1,
             :field is blank.
D BLOCKPT 1 .Blocked Portion Indicator
            :A blank indicates not
            :applicable or summary is
            :for the total geographic
            :area.
V 1 .This summary is for the
    :"blocked portion" of the
    :geographic area.
D FSTATUS 1 .Functional Status Code
V A .Active governmental unit
    :recognized for revenue
    :sharing, except Indian
    :Reservations and Alaska
    :Native villages
  B .Active governmental units,
    :not recognized for revenue
    :sharing
  I .Inactive governmental unit
  N .Nonfunctioning governmental
    :unit
  S .Statistical entity
  F .False entity
  R .Indian reservation or Alaska
    :Native village, recognized
    :for revenue sharing
  Q .Indian reservation or Alaska
    :Native village, not
    :recognized for revenue
    :sharing
D FLAG1 1 23 .Field is blank if less than 20 percent
             :of the persons or year-round housing
             :units were substituted.
V 1 .20 percent or more of the persons or
    :year-round housing units were
    :substituted.
D PARTCOU 1 24    .This code is applicable to New England
                  :summaries (level 11) only.  The field
                  :will be blank if the county is
                  :completely inside or outside any SMSA.
V 1 .New England County is partially inside
```

```
    :one or more SMSA's
D FILLER1 4 25    .This is a filler.
D REGION 1 31 .Region
V 1 .Northeast
V 2 .North Central
V 3 .South
V 4 .West
D DIVISION 1 32 .Division Code
                :This is the first digit
                :of the geographic State
                :code.
V 1 .New England
  2 .Middle Atlantic
  3 .East North Central
  4 .West North Central
  5 .South Atlantic
  6 .East South Central
  7 .West South Central
  8 .Mountain
  9 .Pacific
D STATEGEO 2 32 A C .Geographic State Code
V 63 .Alabama
  94 .Alaska
  86 .Arizona
  71 .Arkansas
  93 .California
  84 .Colorado
  16 .Connecticut
  51 .Delaware
  53 .Dist. of Columbia
  59 .Florida
  58 .Georgia
  95 .Hawaii
  82 .Idaho
  33 .Illinois
  32 .Indiana
  42 .Iowa
  47 .Kansas
  61 .Kentucky
  72 .Louisiana
  11 .Maine
  52 .Maryland
  14 .Massachusetts
  34 .Michigan
  41 .Minnesota
  64 .Mississippi
  43 .Missouri
  81 .Montana
  46 .Nebraska
  88 .Nevada
  12 .New Hampshire
  22 .New Jersey
  85 .New Mexico
  21 .New York
  56 .North Carolina
  44 .North Dakota
  31 .Ohio
  73 .Oklahoma
  92 .Oregon
  23 .Pennsylvania
  15 .Rhode Island
  57 .South Carolina
  45 .South Dakota
  62 .Tennessee
  74 .Texas
  87 .Utah
  13 .Vermont
  54 .Virginia
  91 .Washington
  55 .West Virginia
```

```
     35 .Wisconsin
     83 .Wyoming
   D STATE 2 34 .FIPS State Code
   V 01 .Alabama
     02 .Alaska
     04 .Arizona
     05 .Arkansas
     06 .California
     08 .Colorado
     09 .Connecticut
     10 .Delaware
     11 .Dist. of Columbia
     12 .Florida
     13 .Georgia
     15 .Hawaii
     16 .Idaho
     17 .Illinois
     18 .Indiana
     19 .Iowa
     20 .Kansas
     21 .Kentucky
     22 .Louisiana
     23 .Maine
     24 .Maryland
     25 .Massachusetts
     26 .Michigan
     27 .Minnesota
     28 .Mississippi
     29 .Missouri
     30 .Montana
     31 .Nebraska
     32 .Nevada
     33 .New Hampshire
     34 .New Jersey
     35 .New Mexico
     36 .New York
     37 .North Carolina
     38 .North Dakota
     39 .Ohio
     40 .Oklahoma
     41 .Oregon
     42 .Pennsylvania
     44 .Rhode Island
     45 .South Carolina
     46 .South Dakota
     47 .Tennessee
     48 .Texas
     49 .Utah
     50 .Vermont
     51 .Virginia
     53 .Washington
     54 .West Virginia
     55 .Wisconsin
     56 .Wyoming
   D SMSA 4,,A C .FIPS Standard Metropolitan
              :Statistical Area (SMSA)
   D COUNTY 3 .FIPS County Code
   D MCD 3 .Minor Civil Division/Census
           :County Division (MCD/CCD)
   D PLACE 4 .Census Geographic Place
             :Code
             :9999 in this field indicates
             :balance of higher level entity
             :outside of place. eg; Remainder
             :of MCD or of county.
   D TRACT 6 .Tract or Block Numbering
             :Area (BNA)
             :Tract is a four-digit basic
             :code with implied decimal and
             :2-digit suffix
```

```
                   :999999 in this field indicates untracted
                   :remainder of a hiaher level entitv,
                   :e.g., Remainder of MCD.
D TRACT4 4 50      .Tract or Block Numberina Area (BNA)
                   :This is the 4-digit basic code.
                   :BNA summaries will be numbered between
                   :9901 and 9989.
D TRACT2 2 54      .Tract or Block Numberina Area (BNA)
                   :This is the 2-digit suffix code.
D BLKGRP 1 56      .Block Group (BG)
                   :(First digit of block number)
  BLOCK 3 56       .Block
  PARTPLAC 1 59    .Place/Part Indicator
                   :This indicator will appear
                   :on records which may contain
                   :data for a part of a place.
                   :This indicator is applicable
                   :to summarv levels 13, 14, 15,
                   :16, 19, 20, 21, 22, 23, 24,
                   :25, and 26.
  V 0 .Neither Place, Nor Place Segment
      :Record
    1 .Split
    2 .Not Split
D PARTTRCT 1       .Tract/Part Indicator
                   :This indicator will appear
                   :on records which may contain
                   :data for a part of a tract or BNA.
                   :This indicator is applicable
                   :to summarv levels 14, 15, 16,
                   :20, 21, 22, 24, 25, and 26.
  V 0 .Neither Tract, Nor Tract Seament
      :Record
    1 .Split
    2 .Not Split
D PARTBLK 1        .Block/Part Indicator
                   :This indicator will appear
                   :on records which mav contain
                   :data for a part of a block.
                   :This indicator is applicable
                   :to summary levels 21 and 25.
  V 0 .Not Block, Block Seament
      :Record
    1 .Split
    2 .Not Split
D EDIND 1          .Enumeration District
                   :Indicator Prefix; Field
                   :is blank if ED is none
                   :of those listed below:
  V A .Adiacent Lands to Indian
      :Reservations (historic areas of
      :Oklahoma onlv, excludina urbanized
      :areas)
    M .Militarv Reservation
    N .Indian Reservation
    P .National or State Park or Forest Lands
    S .Other Special Place
    V .Crews of Vessels
D EDNUMBER 4       .Enumeration District Number
                   :(ED)
  EDSUFFIX 1       .Enumeration District Suffix
                   :If applicable, contains an
                   :alphabetic character A-Z.
                   :Otherwise, suffix will
                   :be blank.
```

```
T TAB1 9 253 N N
S 3 .Total urbanized area, and rural
H .URBAN AND RURAL (3)
  :
  : This table has no suppression
  :
U .Universe:  Persons
  :
  :NOTE:  Urban is derived by subtracting
  :          rural from total.
C 1 .Total
    .Inside urbanized areas
    .Rural
T TAB2
S 1 .Families
H .FAMILIES
  :
  : SUPFLG10 applies to all cells
  :
U .Universe:  Families
C 1 .Families
T TAB3
N 1
S 1 .Households
H .HOUSEHOLDS
  :
  : This table has no suppression
  :
U .Universe:  Households
C 1 .Households
T TAB4
N 2
S 3 .Urban and Rural
H .URBAN AND RURAL (3)
  :
  : This table has no suppression
  :
U .Universe:  Housing Units (Including.
  :             Vacant Seasonal And
  :             Migratory Units)
  :
  :NOTE:  Urban is derived by subtracting
  :          rural from total.
C 1 .Total
    .Inside urbanized areas
    .Rural
T TAB5
N 1
S 3 .Occupancy Status
H .OCCUPANCY STATUS (3)
  :
  : This table has no suppression
  :
U .Universe:  Year-Round Housing
  :             Units
C 1 .Total
    .Occupied
    .Vacant
T TAB6
S 2 .Sex
H .SEX (2)
  :
  : SUPFLG01 applies to all cells
  :
U .Universe:  Persons
C 1 .Male
    .Female
T TAB7
N 3 4
S 15 .Race
H .RACE (15)
  :
  : This table has no suppression
  :
U .Universe:  Persons
C 1 .White
    .Black
G   .American Indian, Eskimo, and Aleut:
C 1 .  American Indian
    .  Eskimo
    .  Aleut
G   .Asian and Pacific Islander:
C 1 .  Japanese
    .  Chinese
    .  Filipino
    .  Korean
    .  Asian Indian
    .  Vietnamese
    .  Hawaiian
    .  Guamanian
    .  Samoan
    .Other
T TAB8
S 5 .Spanish Origin
H .SPANISH ORIGIN (5)
  :
  : This table has no suppression
  :
U .Universe:  Persons
C 1 .Not of Spanish origin
    .Mexican
    .Puerto Rican
    .Cuban
    .Other Spanish
T TAB9
N 3 4
S 5 .Race
H .RACE (5)
  :
  : This table has no suppression
  :
U .Universe:  Persons Of Spanish Origin
C 1 .Total
    .White
    .Black
    .American Indian, Eskimo, Aleut, and
    : Asian and Pacific Islander
    .Other
T TAB10
S 2 .Sex
S 26 .Age
H .SEX (2) BY AGE (26)
  :
  : SUPFLG01 applies to all cells
  :
U .Universe:  Persons
C 1 .Total:
    .Female:
C 2 .Under 1 year
    .1 and 2 years
    .3 and 4 years
    .5 years
    .6 years
```

```
      .7 to 9 years
      .10 to 13 years
      .14 years
      .15 years
      .16 years
      .17 years
      .18 years
      .19 years
      .20 years
      .21 years
      .22 to 24 years
      .25 to 29 years
      .30 to 34 years
      .35 to 44 years
      .45 to 54 years
      .55 to 59 years
      .60 and 61 years
      .62 to 64 years
      .65 to 74 years
      .75 to 84 years
      .85 years and over
T TAB11,,,,, -1
S 3 .Median Age by Sex
    :This cell has one implied decimal.
H .MEDIAN AGE BY SEX (3)
  :
  : SUPFLG01 applies to all cells
  :
U .Universe:  Persons
  :
  :NOTE:  One implied decimal place.
C 1 .Total
    .Male
    .Female
T TAB12
N 4
S 5 .Race
S 4 .Age
H .RACE (5) BY AGE (4)
  :
  : SUPFLG01 applies to cells 1-4
  : SUPFLG02 applies to cells 5-8
  : SUPFLG03 applies to cells 9-12
  : SUPFLG04 applies to cells 13-16
  : SUPFLG05 applies to cells 17-20
  :
U .Universe:  Persons
C 1 .Total:
    .White:
    .Black:
    .American Indian, Eskimo, and Aleut:
    .Asian and Pacific Islander:
C 2 .Under 5 years
    .5 to 17 years
.   .18 to 64 years
    .65 years and over


FOOTNOTE
 1   .The count of households in the
     :complete-count tabulations by definition
     :equals the number of occupied housing
     :units.  For this documentation the term
     :"Household" is used unless the matrix is
     :stratified by a housing item such as
     :tenure, plumbing facilities, etc., in
     :which case, the term "Occupied housing
     :unit" is used.
 2   .Vacant seasonal/migratory units are
     :excluded from all other tabulation
     :matrices.
 3   .For data tabulated on a 100-percent
     :basis, includes all persons in the
     :category "Other."  For data tabulated
     :from the sample, excludes those persons
     :who have a write-in entry of an Asian
     :and Pacific Islander group in the
     :"Other" category.
```

4 ."Asian and Pacific Islander," in the
:100-percent tabulations, includes
:"Japanese," "Chinese," "Filipino,"
:"Korean," "Asian Indian," "Vietnamese,"
:"Hawaiian," "Guamanian," and "Samoan."
:"Asian and Pacific Islander," in sample
:tabulations, includes the groups
:listed above and those persons who have
:a write-in entry of an Asian or Pacific
:Islander group in the "Other" category.

5 .Relatives include householder, spouse,
:and the questionnaire categories:
:"Son/daughter," "Brother/sister,"
:"Father/mother," and "Other relative."
:Tabulations of "Other relatives"
:include all categories not shown
:separately in the matrix.

6 ."Nonrelatives" include the questionnaire
:categories: "Roomer, boarder,"
:"Partner, roommate," "Paid employee,"
:and "Other nonrelative." Tabulations
:of "Nonrelatives" include all
:categories not shown separately in
:the matrix.

7 .Tabulations of "Persons in household"
:based on 100-percent data by definition
:are the same as tabulations of "Persons
:in unit." The phrase "Persons in
:household" is used unless the matrix
:is stratified by a housing item such as
:tenure, plumbing facilities, etc., in
:which case, the phrase "Persons in
:unit" is used. Tabulations of
:"Persons in household" and "Persons
:in unit" based on sample data are not
:necessarily the same because of
:differences in the procedures used
:to inflate sample population and
:housing data.

8 .A "Child of householder" includes any
:son, daughter, stepchild, or adopted
:child of the householder. An "Own
:child of householder" is a never-
:married child under 18 years of age
:who is a son, daughter, stepchild,
:or adopted child of the householder.
:"Related children" include not only
:own children but also all other family
:members, regardless of marital status,
:who are under 18 years old, except the
:householder or spouse. Foster
:children are included in the
:"Nonrelative" category.

9 .Vacant housing units include the
:questionnaire categories: "For
:rent," "For sale only," "Rented
:or sold, not occupied," "Held
:for occasional use," and "Other
:vacant." Tabulations of "Other
:vacants" include all categories not
:shown separately in the matrix.

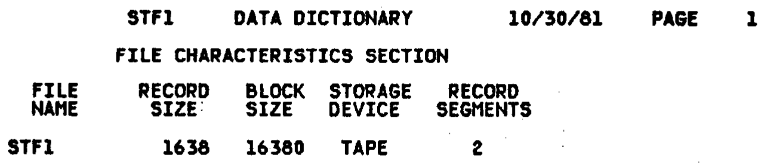
10 .This aggregate, along with the relevant
:count, will permit the computation of a
:mean. For example, the aggregate value
:for specified owner-occupied
:noncondominium units will yield the mean
:value when divided by the
:count of specified owner-occupied
:noncondominium units, and the aggregate
:rooms for occupied and vacant year-round
:units divided by the count of occupied
:and vacant year-round units yields
:mean rooms. (See footnote 14
:prior to computing mean value
:or price asked.)

```
11 .Value and price asked are tabulated
   :separately for noncondominium and
   :condominium units.
   :The noncondominium value distribution
   :is restricted to certain kinds of
   :"owner-occupied" units;
   :the noncondominium price
   :asked distribution is restricted to
   :certain kinds of "vacant-for-sale only"
   :units.  The following are excluded
   :from the tabulations on value and price
   :asked for noncondominium units:
   :      a.  Units at an address with two
   :          or more units
   :      b.  Units on 10 or more acres
   :      c.  Units with a commercial
   :          establishment or medical office
   :          on the property
   :      d.  Mobile homes or trailers
   :The condominium value distribution is
   :tabulated for all "owner-occupied"
   :condominium units; the condominium
   :price asked distribution is tabulated
   :for all "vacant-for-sale only"
   :condominium units.
12 .Contract rent is tabulated for all
   :"renter-occupied" units except
   :one-family houses on a property
   :of 10 or more acres.  Rent
   :asked is tabulated for all "vacant-for-
   :rent" units except one-family houses
   :on a property of 10 or more acres. Units
   :tabulated in the "No Cash Rent" category
   :are all occupied housing units reported
   :as "No Cash Rent" except one-family
   :houses on 10 or more acres.
13 .Lacking complete plumbing (facilities)
   :for exclusive use includes:  Complete
   :plumbing (facilities) but also used
   :by another household, some but not all
   :plumbing facilities, or no plumbing
   :facilities.
14 .Multiply the aggregate value and price
   :asked by $250 to obtain the true value
   :or price asked. The tabulation was
   :scaled by a factor of $250 for tally
   :purposes.
15 .Allocation of marital status for
   :persons under 15 years old are not
   :included in "Persons with one or more
   :items allocated."
16 ."Year-round housing units with one or
   :more housing items allocated"
   :includes allocations of "Units at
   :address," "Access," "Plumbing
   :facilities," "Rooms," "Tenure,"
   :"Condominium status," "Acreage of
   :property," "Commercial establishment
   :or medical office," "Value" or "Price
   :asked," "Contract rent" or "Rent asked,"
   :"Vacancy indicator," "Vacancy status,"
   :"Boarded up status," and "Duration of
   :vacancy."
TEXT
                    Census of Population and Housing, 1980-
                             Summary Tape File 1

   Geographic information in positions 1-204 of this file are in
a standard geographic record format which will be followed for all
1980 summary tape files.  Each field of this standard format is
identified in this dictionary, although it may not be used in STF 1.
   When processing this file, the geographic identification
portion of each record segment (positions 1-252 and positions
1639-1740 of the record) should be read as alphanumerics.  The
remaining portion of the file should be read as numeric.
   STF 1A, STF 1B, and STF 1C have the same record layout.
They differ only in their geographic coverage.
END
```

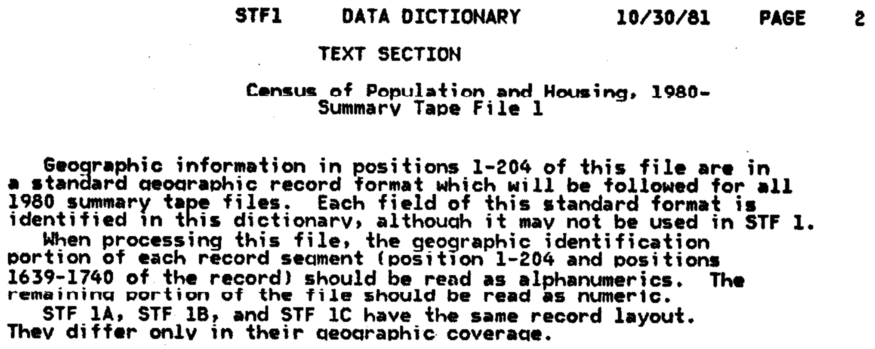
Sample Data Dictionary Listing

The data dictionary printout provided here is the result of processing the previous data dictionary definition statements through the Documentor program. This data dictionary listing is taken from parts of the STF1 data dictionary.



```
                  STF1     DATA DICTIONARY        10/30/81     PAGE    1
                 FILE CHARACTERISTICS SECTION
      FILE      RECORD    BLOCK  STORAGE   RECORD
      NAME       SIZE     SIZE   DEVICE   SEGMENTS

 STF1            1638    16380    TAPE        2
```



```
                  STF1     DATA DICTIONARY        10/30/81     PAGE    2
                         TEXT SECTION

                 Census of Population and Housing, 1980-
                         Summary Tape File 1

   Geographic information in positions 1-204 of this file are in
a standard geographic record format which will be followed for all
1980 summary tape files.  Each field of this standard format is
identified in this dictionary, although it may not be used in STF 1.
   When processing this file, the geographic identification
portion of each record segment (position 1-204 and positions
1639-1740 of the record) should be read as alphanumerics.  The
remaining portion of the file should be read as numeric.
   STF 1A, STF 1B, and STF 1C have the same record layout.
They differ only in their geographic coverage.
```

```
                  STF1     DATA DICTIONARY        10/30/81     PAGE     3
                                RECORD  A

                  SIZE/        RELATIVE
  NAME            SCALE  BEGIN  BEGIN

FILEID              5      1      1
                                  Identifier for summary file,
                                  i.e., STF 1A, STF 1B, etc.
                                  STF1A-Summary Tape File 1A
                                  STF1B-Summary Tape File 1B
                                  STF1C-Summary Tape File 1C


RECTYP              4      6      6
                                  Identifies multiple logical
                                  record formats on STF files
                                  It is blank if only one
                                  format is present.  (Blank
                                  on STF 1)


SUMRYLVL            2     10     10
                                  Identifies Geographic level
                                  of current record
                        01        United States
                        02        Region
                        03        Division
                        04        State or State equivalent
                        05        SCSA
                        06        SCSA/State
                        07        SMSA
                        08        SMSA/State
                        09        Urbanized Area
                        10        Urbanized Area/State
                        11        State/County or County equivalent
                        12        State/County/MCD (CCD)
                        13        State/County/MCD (CCD)/
                                  Place
                        14        State/County/MCD (CCD)/
                                  Place/Tract (BNA)
                        15        State/County/MCD (CCD)/
                                  Place/Tract (BNA)/BG
                        16        State/County/MCD (CCD)/
                                  Place/Tract (BNA)/ED
                        17        State/SMSA/County
                        18        State/SMSA/County/MCD (CCD)
                        19        State/SMSA/County/MCD (CCD)/
                                  Place
                        20        State/SMSA/County/MCD (CCD)/
                                  Place/Tract (BNA)
                        21        State/SMSA/County/MCD (CCD)/
                                  Place/Tract (BNA)/Block
                        22        State/SMSA/County/MCD (CCD)/
                                  Place/Tract (BNA)/ED
                        23        State/SMSA/County/Place
                        24        State/SMSA/County/Place/
                                  Tract (BNA)
                        25        State/SMSA/County/Place/
                                  Tract (BNA)/Block
                        26        State/SMSA/County/Place/
                                  Tract (BNA)/ED
                        27        State/Place
                        28        State/MCD Sequence Number
                        29        Indian Reservation (ANV)
                        30        Indian Reservation (ANV)/State
                        31        Indian Reservation (ANV)/State/
```

```
               STF1     DATA DICTIONARY        10/30/81    PAGE    4
                       RECORD  A

             SIZE/       RELATIVE
  NAME       SCALE  BEGIN  BEGIN
                               County
                     32        State/SMSA/County/Tract (BNA)
                     33        State/Congressional District


URBARURL       2       12     12
                               Urban and Rural component
                               Only 01 and 08 are
                               indicated on STF 1.
                     00        Not Urban and Rural component
                     01        Urban
                     02          Inside urbanized areas
                     03            Central Cities
                     04            Urban fringe
                     05          Outside urbanized areas
                     06            Places of 10,000 or more
                     07            Places of 2500 to 10,000
                     08        Rural
                     09          Places of 1000 to 2500
                     10          Other rural
                     11          Farm

SMSACOM        2       14     14
                               Inside and Outside SMSA's
                               Component
                               Note:  Not applicable to STF 1,
                               field is blank.
                     00        Not Inside and Outside SMSA
                               Component
                     01        Inside SMSA's
                     02          Urban
                     03            Central cities
                     04            Not in Central cities
                     05          Rural
                     06        Outside SMSA's
                     07          Urban
                     08          Rural

RACESPAN       2       16     16
                               Identifies Race/Spanish
                               Origin Group
                               Note:  Not applicable to STF 1,
                               field is blank.

ANCESTRY       3       18     18
                               Identifies specific ancestry
                               group
                               Note:  Not applicable to STF 1,
                               field is blank.

BLOCKPT        1       21     21
                               Blocked Portion Indicator
                               A blank indicates not
                               applicable or summary is
                               for the total geographic
                               area.
                     1         This summary is for the
                               "blocked portion" of the
                               geographic area.
```

RECORD A

NAME	SIZE/ SCALE	BEGIN	RELATIVE BEGIN
FSTATU	1	22	22 Functional Status Code
		A	Active governmental unit recognized for revenue sharing, except Indian Reservations and Alaska Native villages
		B	Active governmental units, not recognized for revenue sharing
		I	Inactive governmental unit
		N	Nonfunctioning governmental unit
		S	Statistical entity
		F	False entity
		R	Indian reservation or Alaska Native village, recognized for revenue sharing
		Q	Indian reservation or Alaska Native village, not recognized for revenue sharing
FLAG1	1	23	23 Field is blank if less than 20 percent of the persons or year-round housing units were substituted.
		1	20 percent or more of the persons or year-round housing units were substituted.
PARTCOU	1	24	24 This code is applicable to New England summaries (level 11) only. The field will be blank if the county is completely inside or outside any SMSA.
		1	New England County is partially inside one or more SMSA's
FILLER1	4	25	25 This is a filler.
REGION	1	31	31 Region
		1	Northeast
		2	North Central
		3	South
		4	West
DIVISION	1	32	32 Division Code This is the first digit of the geographic State code.
		1	New England
		2	Middle Atlantic
		3	East North Central
		4	West North Central

```
                  STF1     DATA DICTIONARY        10/30/81     PAGE     6
                               RECORD  A

                   SIZE/         RELATIVE
  NAME             SCALE   BEGIN  BEGIN
                             5           South Atlantic
                             6           East South Central
                             7           West South Central
                             8           Mountain
                             9           Pacific

STATEGEO             2        32     32
                                         Geographic State Code
                             63          Alabama
                             94          Alaska
                             86          Arizona
                             71          Arkansas
                             93          California
                             84          Colorado
                             16          Connecticut
                             51          Delaware
                             53          Dist. of Columbia
                             59          Florida
                             58          Georgia
                             95          Hawaii
                             82          Idaho
                             33          Illinois
                             32          Indiana
                             42          Iowa
                             47          Kansas
                             61          Kentucky
                             72          Louisiana
                             11          Maine
                             52          Maryland
                             14          Massachusetts
                             34          Michigan
                             41          Minnesota
                             64          Mississippi
                             43          Missouri
                             81          Montana
                             46          Nebraska
                             88          Nevada
                             12          New Hampshire
                             22          New Jersey
                             85          New Mexico
                             21          New York
                             56          North Carolina
                             44          North Dakota
                             31          Ohio
                             73          Oklahoma
                             92          Oregon
                             23          Pennsylvania
                             15          Rhode Island
                             57          South Carolina
                             45          South Dakota
                             62          Tennessee
                             74          Texas
                             87          Utah
                             13          Vermont
                             54          Virginia
                             91          Washington
                             55          West Virginia
                             35          Wisconsin
                             83          Wyoming
```

```
           STF1     DATA DICTIONARY       10/30/81    PAGE    7
                        RECORD  A

                 SIZE/         RELATIVE
  NAME           SCALE  BEGIN  BEGIN

STATE              2      34     34
                                  FIPS State Code
                        01        Alabama
                        02        Alaska
                        04        Arizona
                        05        Arkansas
                        06        California
                        08        Colorado
                        09        Connecticut
                        10        Delaware
                        11        Dist. of Columbia
                        12        Florida
                        13        Georgia
                        15        Hawaii
                        16        Idaho
                        17        Illinois
                        18        Indiana
                        19        Iowa
                        20        Kansas
                        21        Kentucky
                        22        Louisiana
                        23        Maine
                        24        Maryland
                        25        Massachusetts
                        26        Michigan
                        27        Minnesota
                        28        Mississippi
                        29        Missouri
                        30        Montana
                        31        Nebraska
                        32        Nevada
                        33        New Hampshire
                        34        New Jersey
                        35        New Mexico
                        36        New York
                        37        North Carolina
                        38        North Dakota
                        39        Ohio
                        40        Oklahoma
                        41        Oregon
                        42        Pennsylvania
                        44        Rhode Island
                        45        South Carolina
                        46        South Dakota
                        47        Tennessee
                        48        Texas
                        49        Utah
                        50        Vermont
                        51        Virginia
                        53        Washington
                        54        West Virginia
                        55        Wisconsin
                        56        Wyoming

SMSA               4      36     36
                                  FIPS Standard Metropolitan
                                  Statistical Area (SMSA)

COUNTY             3      40     40
                                  FIPS County Code
                 STF1     DATA DICTIONARY        10/30/81    PAGE    8
                              RECORD  A

                  SIZE/         RELATIVE
   NAME           SCALE   BEGIN  BEGIN

 MCD                3       43     43
                                    Minor Civil Division/Census
                                    County Division (MCD/CCD)

 PLACE              4       46     46
                                    Census Geographic Place
                                    Code
                                    9999 in this field indicates
                                    balance of higher level entity
                                    outside of place. eg; Remainder
                                    of MCD or of county.

 TRACT              6       50     50
                                    Tract or Block Numbering
                                    Area (BNA)
                                    Tract is a four-digit basic
                                    code with implied decimal and
                                    2-digit suffix
                                    999999 in this field indicates untracted
                                    remainder of a higher level entity,
                                    e.g., Remainder of MCD.

 TRACT4             4       50     50
                                    Tract or Block Numbering Area (BNA)
                                    This is the 4-digit basic code.
                                    BNA summaries will be numbered between
                                    9901 and 9989.

 TRACT2             2       54     54
                                    Tract or Block Numbering Area (BNA)
                                    This is the 2-digit suffix code.

 BLKGRP             1       56     56
                                    Block Group (BG)
                                    (First digit of block number)

 BLOCK              3       56     56
                                    Block

 PARTPLAC           1       59     59
                                    Place/Part Indicator
                                    This indicator will appear
                                    on records which may contain
                                    data for a part of a place.
                                    This indicator is applicable
                                    to summary levels 13, 14, 15,
                                    16, 19, 20, 21, 22, 23, 24,
                                    25, and 26.
                                  0         Neither Place, Nor Place Segment
                                            Record
                                  1         Split
                                  2         Not Split

 PARTTRCT           1       60     60
```

```
                              RECORD  A

   NAME       SIZE/           RELATIVE
              SCALE   BEGIN    BEGIN
                                    Tract/Part Indicator
                                    This indicator will appear
                                    on records which may contain
                                    data for a part of a tract or BNA.
                                    This indicator is applicable
                                    to summary levels 14, 15, 16,
                                    20, 21, 22, 24, 25, and 26.
                       0            Neither Tract, Nor Tract Segment
                                    Record
                       1            Split
                       2            Not Split


 PARTBLK        1        61      61
                                    Block/Part Indicator
                                    This indicator will appear
                                    on records which may contain
                                    data for a part of a block.
                                    This indicator is applicable
                                    to summary levels 21 and 25.
                       0            Not Block, Block Segment
                                    Record
                       1            Split
                       2            Not Split

 EDIND          1        62      62
                                    Enumeration District
                                    Indicator Prefix; Field
                                    is blank if ED is none
                                    of those listed below:
                       A            Adjacent Lands to Indian
                                    Reservations (historic areas of
                                    Oklahoma only, excluding urbanized
                                    areas)
                       M            Military Reservation
                       N            Indian Reservation
                       P            National or State Park or Forest Lands
                       S            Other Special Place
                       V            Crews of Vessels

 EDNUMBER       4        63      63
                                    Enumeration District Number
                                    (ED)

 EDSUFFIX       1        67      67
                                    Enumeration District Suffix
                                    If applicable, contains an
                                    alphabetic character A-Z.
                                    Otherwise, suffix will
                                    be blank.
```

```
                           RECORD  A

 NAME         SIZE/        RELATIVE  NUMBER OF
              SCALE  BEGIN  BEGIN      CELLS

TABLE 1         9     253    253         3
(TAB1)
            URBAN AND RURAL (3)

             This table has no suppression

              Universe:  Persons

              NOTE:  Urban is derived by subtracting
                        rural from total.

                Total
                Inside urbanized areas
                Rural

TABLE 2         9     280    280         1
(TAB2)
            FAMILIES

             SUPFLG10 applies to all cells

              Universe:  Families

                Families

TABLE 3         9     289    289         1
(TAB3)
            HOUSEHOLDS

             This table has no suppression

              Universe:  Households

              SEE FOOTNOTE  1

                Households

TABLE 4         9     298    298         3
(TAB4)
            URBAN AND RURAL (3)

             This table has no suppression

              Universe:  Housing Units (Including
                           Vacant Seasonal And
                           Migratory Units)

              NOTE:  Urban is derived by subtracting
                        rural from total.

              SEE FOOTNOTE  2

                Total
                Inside urbanized areas
                Rural
```

```
                        RECORD  A

              SIZE/         RELATIVE   NUMBER OF
NAME          SCALE  BEGIN  BEGIN        CELLS

TABLE 5         9     325    325           3
(TAB5)
             OCCUPANCY STATUS (3)

              This table has no suppression

               Universe:  Year-Round Housing
                            Units

               SEE FOOTNOTE  1

                  Total
                  Occupied
                  Vacant

TABLE 6         9     352    352           2
(TAB6)
             SEX (2)

              SUPFLG01 applies to all cells

               Universe:  Persons

                  Male
                  Female

TABLE 7         9     370    370          15
(TAB7)
             RACE (15)

              This table has no suppression

               Universe:  Persons

               SEE FOOTNOTE  3  4

                  White
                  Black
                  American Indian, Eskimo, and Aleut:
                     American Indian
                     Eskimo
                     Aleut
                  Asian and Pacific Islander:
                     Japanese
                     Chinese
                     Filipino
                     Korean
                     Asian Indian
                     Vietnamese
                     Hawaiian
                     Guamanian
                     Samoan
                  Other
```

```
                        RECORD  A

                SIZE/          RELATIVE  NUMBER OF
  NAME          SCALE   BEGIN   BEGIN      CELLS

TABLE 8           9      505     505          5
(TAB8)
               SPANISH ORIGIN (5)

                This table has no suppression

                 Universe:  Persons

                   Not of Spanish origin
                   Mexican
                   Puerto Rican
                   Cuban
                   Other Spanish

TABLE 9           9      550     550          5
(TAB9)
               RACE (5)

                This table has no suppression

                 Universe:  Persons Of Spanish Origin

                 SEE FOOTNOTE  3  4

                   Total
                   White
                   Black
                   American Indian, Eskimo, Aleut, and
                    Asian and Pacific Islander
                   Other

TABLE 10          9      595     595         52
(TAB10)
               SEX (2) BY AGE (26)

                SUPFLG01 applies to all cells

                 Universe:  Persons

                 THE STRATIFIERS ARE
                   Sex BY
                     Age

                   Total:
                     Under 1 year
                     1 and 2 years
                     3 and 4 years
                     5 years
                     6 years
                     7 to 9 years
                     10 to 13 years
                     14 years
                     15 years
                     16 years
                     17 years
                     18 years
                     19 years
                     20 years
                     21 years
                     22 to 24 years
                     25 to 29 years
```

```
                         RECORD  A

                SIZE/         RELATIVE   NUMBER OF
   NAME         SCALE  BEGIN   BEGIN       CELLS
                     30 to 34 years
                     35 to 44 years
                     45 to 54 years
                     55 to 59 years
                     60 and 61 years
                     62 to 64 years
                     65 to 74 years
                     75 to 84 years
                     85 years and over

                   Female:
                     REPEAT Age (26)

 TABLE 11        9/-1  1063   1063           3
 (TAB11)
               MEDIAN AGE BY SEX (3)

                SUPFLG01 applies to all cells

                 Universe:  Persons

                 NOTE:  One implied decimal place.

                   Total
                   Male
                   Female

 TABLE 12         9    1090   1090          20
 (TAB12)
               RACE (5) BY AGE (4)

                SUPFLG01 applies to cells 1-4
                SUPFLG02 applies to cells 5-8
                SUPFLG03 applies to cells 9-12
                SUPFLG04 applies to cells 13-16
                SUPFLG05 applies to cells 17-20

                 Universe:  Persons

                 SEE FOOTNOTE  4

                 THE STRATIFIERS ARE
                   Race BY
                     Age

                   Total:
                     Under 5 years
                     5 to 17 years
                     18 to 64 years
                     65 years and over

                   White:
                     REPEAT Age (4)
                   Black:
                     REPEAT Age (4)
                   American Indian, Eskimo, and Aleut:
                     REPEAT Age (4)
                   Asian and Pacific Islander:
                     REPEAT Age (4)
```

CENSPAC Installation

CENSPAC is made available on magnetic tape that includes all source code, data files for test runs, the correct output of those test runs and machine-readable data dictionaries. The physical characteristics of the release tape vary depending on the purchaser's specifications. Documentation on these details is included with the tape itself. The installation process should require one to three days of programmer time. Questions on the CENSPAC installation procedure should be referred to the CENSPAC development staff at (301) 763-5242.

Installation Steps

To install CENSPAC, you must:

1) Select appropriate CENSPAC source file.
2) Convert the source code programs (SOURCE and MODEL) to your operating system, if your operating system is other than IBM OS/VS compatible.
3) Modify the source code and data files, if necessary, as specified in the Census Bureau's CENSPAC User Notes.
4) Compile and save the object decks and executable programs in the appropriate libraries.
5) Initialize the data dictionary file.
6) Initialize CENSPAC MODEL file.
7) Execute test run.
8) Register the installation with the Census Bureau.

A flow chart that describes the installation procedure is provided on page 183.

The files included on the CENSPAC tape are:

File #	File Name	Contents
1	INSTJCL	Sample IBM OS JCL to install and run the CENSPAC system.
2	CHANGECP	This file contains the source code for a small COBOL program to convert the SOURCE and MODEL files from the system release tape operating system (IBM OS/VS) to the user's operating system (such as UNIVAC EXEC-8, DOS, etc.), and/or to split the source file into individual programs. This file is executable on IBM OS/VS

		compatible systems and is to be used only when a user is modifying CENSPAC for installation on IBM DOS, Burroughs B7700, or other systems except UNIVAC EXEC-8. Instructions in the program source code identify the changes necessary for DOS, CDC, DEC, and Burroughs 7700 installations.
3	CHANGECP	This file contains the source code for a small COBOL program to convert the SOURCE and MODEL files from the Bureau's operating system (IBM VS) to the user's operating system (such as UNIVAC EXEC-8, DOS, etc.) and/or or split the source file into individual programs. This file is executable on UNIVAC EXEC-8 systems. This version of the program will also generate a run stream to compile and save the source routines on UNIVAC EXEC-8 systems. In addition, it contains sample run statements to compile CHANGCP and run CENSPAC and DOCUMENTOR on UNIVAC EXEC-8 systems.
4	SOURCE1	The source code for CENSPAC programs and run-time subroutines. This source file (1 of 2 CENSPAC source files on the tape) contains CENSPAC and DOCUMENTOR broken into subroutines.
5	MODEL	The skeleton source code used by CENSPAC to generate the COBOL executor. This file is an input file used for the CENGEN program.
6	TESTDATA	Test data file for verifying that CENSPAC is correctly installed. This is a subset of the Richmond STF1 test file.
7	DICTION	The data dictionary definition statements for data dictionary files SYSTEXT, MACLIB, STF1, and STF2. This file is used as input to the DOCUMENTOR program to establish the data dictionary file prior to testing the CENSPAC installation.
8	DICTLIST	The correct output from the Documentor program when using the input file DICTION. This is a print image file of the SYSTEXT, MACLIB, STF1 and STF2 data dictionaries.
9	TESTRUN	Contains CENSPAC commands to execute a test run.
10	TESTPRT	The correct printed output from CENSPAC for the test run using the CENSPAC commands provided in file 9.

11 DICTION2 Data dictionary definition statements for 1970 First, Second, Fourth Population, and Fifth Count File C files, and the PL 94-171 reapportionment file (CNT1, CNT2, CNT4P, CNT5C and PL94171). This file is used as input to the DOCUMENTOR program to update the master data dictionary created with file DICTION (file #7).

12 DICTION3 Data dictionary definition statements for the remaining data dictionaries as identified in Appendix 5.

13 SOURCE2 The source code for CENSPAC programs and run-time subroutines. This source file (one of 2 on this tape) contains CENSPAC and DOCUMENTOR as individual programs.

Step 1: Select appropriate CENSPAC source file.

The release tape contains two files of CENSPAC source program statements (file #4 and file #13). Either file will result in the same CENSPAC system. The one you select to install (use only one) will depend upon your personal preference and on your operating system features.

The first source file, SOURCE1 file #4, is equivalent to the source file provided with previous releases of CENSPAC. It contains 27 program modules; the preliminary CENSPAC tapes contained 17 program modules. The additional modules are present because the Documentor program is now in 5 modules instead of one program and because CENSPAC includes 2 new program modules and 4 additional I/O modules.

For IBM OS/VS and UNIVAC EXEC-8 users, installation of this version of CENSPAC (version 3.0) using file #4, SOURCE1, should be largely similar to installation procedures followed for the August 1980 (version 2.0) CENSPAC system.

The second source file, file #13, SOURCE2, contains 15 program modules. This file includes CENSPAC and the Documentor program as individual programs with no subroutines, as well as CENGEN and the I/O modules as in file 4.

The CENSPAC source program files contain the following programs and subprograms:

Programs	Module Name	Module Function
CENSPAC	CENSPAC	main CENSPAC program
	PACIO	card, print, and FRAG file input and output handler for CENSPAC
	ITEMCMD	ITEM command processor
	OUTCMD	FILEIN-FILEOUT command processor
	INTRFAC	data dictionary interface module
	IFCMD	IF, COMPUTE, and TOTAL command processor
	GEN	EXECUTOR COBOL program generator module
	CENSLAB	LABEL and HELP command processor
	CENSDOC	FILEOUT documentation processor
CENGEN	CENGEN	CENGEN program processes 'FRAGS' from CENSPAC to create EXECUTOR COBOL program
DOCUMENTOR	DDFIO	card print, and data dictionary input and output handler for the Documentor program
	DDFPARS	Documentor input statement processing module
	DDFGEN	Documentor data dictionary creation module
	DDFLIST	Documentor data dictionary listing module
	DOCUMENT	Documentor main program
I/O MODULES	STF1AXX	one of 2 STF1 I/O modules
	STF1BXX	one of 2 STF1 I/O modules
	STF2AXX	one of 2 STF2 I/O modules
	STF2BXX	one of 2 STF2 I/O modules
	CNT1AXX	one of 2 1970 First Count I/O modules
	CNT1BXX	one of 2 1970 First Count I/O modules

CNT2AXX	one of 2 1970 Second Count I/O modules
CNT2BXX	one of 2 1970 Second Count I/O modules
CNT4PAX	one of 2 1970 Fourth Count I/O modules
CNT4PBX	one of 2 1970 Fourth Count I/O modules
CNT5CAX	one of 2 1970 Fifth Count I/O modules
CNT5CBX	one of 2 1970 Fifth Count I/O modules

It is expected that most users will use file #4, SOURCE1, to install CENSPAC. The installation instructions include program overlays for the subroutines in this file, which will reduce main storage requirements, and future User Note program corrections will be provided using this file as a base.

This second source file, file #13 , SOURCE2 is to be used for Burroughs B7700, and perhaps other systems except UNIVAC EXEC-8, IBM OS/VS, and IBM DOS compatible installations.

The one-large-program structure of this second source file avoids some subroutine linkage problems, since there are fewer subroutines in this program file. Except for references to program overlays and CENSPAC subroutines, the installation procedure described here is the same regardless of which file is used. Both installation procedures use I/O module subroutines. These I/O modules subroutines are used by the CENSPAC EXECUTOR in the last step of the CENSPAC procedure only. The other subroutines being discussed above (present in file #4, combined into the main program in file #13) affect the CENSPAC Documentor and the main CENSPAC program only.

CENSPAC is the first of three program steps in the CENSRUN procedure (JCL 'PROC' to execute CENSPAC system). It creates a work dataset on the FRAG file based upon input from the user's command file and the data dictionary file.

The second program module, CENGEN, is invoked as the second program step of CENSRUN. It is a stand-alone program which takes the FRAG file built by CENSPAC and the CENSPAC MODEL file to produce a COBOL program, the EXECUTOR, which is compiled and run as the third step of the CENSRUN procedure.

The third program module included in the source file, Documentor, processes user input to create and update the data dictionary file. The Documentor is not part of the CENSRUN procedure. The data dictionary file created by the Documentor is used by the CENSPAC program.

The SOURCE file also includes a collection of I/O modules designed to read from Census summary data files. These modules must be compiled and link-edited in the same manner as the above routines. On IBM OS/VS systems they should be individually linked into load module form with member names the same as the program ID names. The compile and link JCL provided in the INSTJCL file will do all this in one two step job. This library must then be concatenated with the other linkage editor libraries. On UNIVAC EXEC-8 systems these modules must be stored in relocatable form in a program file which has been 'PREP'ed and then 'LIB'ed when the EXECUTOR program is 'MAP'ed. On DOS systems, these I/O modules must be available in the system relocatable library. On Burroughs systems the object form is to be made available for binding with the EXECUTOR program. These I/O modules will be individually referenced during the third step of the CENSRUN procedure at run-time. The I/O routines all have names in the form:

aaaayXX or aaaaayX
where: aaaa or aaaaa is the preassigned file name for the Census Summary File that the routine processes, such as: STF1, STF2, CNT1, CNT2, CNT4P or CNT5C.

y is either an A or B to identify whether the routine is used with the 'A' (always used) or 'B' (only used in a MATCH) file. The only difference between the two files aaaaAXX and aaaaBXX is the assignments of the files at run time.

If special file processing capabilities are needed (such as the use of variable length records or non-sequential files) user-written I/O modules can be added to the library. These routines must:

1) open their data set on the first call, and

2) return one complete record to the CENSPAC EXECUTOR on each invocation.

The source for one of the supplied I/O modules, such as STF1AXX, will provide a model for the writing of such user routines. The names of these routines must be in the form:

aaaaayXX
where: a = the module name (5 characters or less).

y is A or B as for the supplied routines, such that the 'A' routine processes records for the 'A' file (DDNAME) AINP, and the 'B' routine processes records for the file BINP.

The name must be filled with 'X's as necessary to a length of 7 characters. The full 7 character name must also be used as the PROGRAM-ID.

These I/O modules can then be referenced in the IOMODULE parameter of the FILEIN command. For example, the CENSPAC command.

```
FILEIN DICTNAME STF1 IOMODULE CDEF
```

would use the STF1 data names and record layout information from the STF1 data dictionary, but actual file accesses would be handled by calling a subroutine identified as 'CDEFAXX', where 'CDEF' is the user name of the I/O module, the first FILEIN command assumes it an 'A' file designator and CENSPAC fills the name to 7 characters with X's. If this command were the second FILEIN command for a MATCH application, the file designator of 'A' would be 'B' and the system would expect to find an I/O module of 'CDEFBXX' in the I/O module library.

Step 2: Converting the CENSPAC source programs

If the user would like to compile program modules individually or convert them to another operating system (DOS, Burroughs 7700, UNIVAC EXEC-8, DEC VAX, DEC 10, DEC 20, and CDC 7000 the only other supported systems for CENSPAC), the CHANGECP program (file 2 or 3), may be used to convert the source code (files SOURCE1 or SOURCE2, and MODEL) and/or to split the large SOURCE1 file into individual programs modules. CHANGECP file 2, is for IBM compatible systems and with program changes as documented in the program serves as a base for other conversions. CHANGECP file 3, is for UNIVAC EXEC-8 systems.

For users at IBM OS/VS compatible installations, the installation JCL in file 1, INSTJCL, can be used to install CENSPAC; program conversion using CHANGECP is not needed. These users can skip to Step 3 of the instructions. The comments from CHANGECP are reproduced below to explain the use of CHANGECP.

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. CHANGCPU.
000300 AUTHOR.
000400 DATE-COMPILED.
000500*REMARKS.      THE CHANGCPU PROGRAM WILL READ COBOL SOURCE PROGRAM
000600*              FILES, MODIFY CERTAIN CARD IMAGES THAT HAVE
000700*              BEEN PREVIOUSLY IDENTIFIED FOR SYSTEM DEPENDENCIES,
000800*              AND WRITE A NEW COPY OF THE SOURCE DECK WHICH HAS
000900*              BEEN MODIFIED FOR THE TARGET COMPUTER SYSTEM.
001000*                  CHANGECP IS USED TO READ THE CENSPAC SOURCE
001100*              PROGRAM AND MODEL FILES.  DUE TO SYSTEM VARIATIONS
001200*              IN COMPILERS, THIS PROGRAM WAS DEVELOPED TO RECOGNIZE
001300*              THESE DIFFERENCES AND ADAPT THE CENSPAC SOURCE CODE
001400*              TO THE SYSTEM COMPILER ON WHICH CENSPAC IS BEING
001500*              INSTALLED.  THIS TASK IS ACCOMPLISHED THRU A SERIES
001600*              OF STEPS.
```

```
001700*
001800*              1. SELECT THE CORRECT CHANGECP PROGRAM FILE
001900*                 FROM THE CENSPAC SYSTEM TAPE;
002000*                     IBM-OS CHANGECP IS IN FILE 2
002100*                     EXEC-8 CHANGECP IS IN FILE 3
002200*              2. THE JOB CONTROL STATEMENTS NECESSARY
002300*                 TO COMPILE AND EXECUTE THE CHANGECP
002400*                 PROGRAM ARE LOCATED IN FILE 1 ON THE CENSPAC
002500*                 SYSTEM RELEASE TAPE.  THE EXECUTION IS SET
002600*                 UP TO CONVERT ALL PROGRAMS IN ONE RUN
002700*                 AND PLACE THE NEW SOURCE ON A DISK
002800*                 DATA SET.
002900*              3.  THE CHANGECP JCL CONTAINS THE INPUT
003000*                 CONTROL CARD TO CHANGECP.  THE FORMAT
003100*                 OF THIS CARD IS;
003200*                     COLUMNS  DESCRIPTION
003300*                     ·1-8     IBM-OS OR U-EXEC-8
003400*                      9       THIS DEFINES THE CODE USED TO
003500*                              DEFINE THE MODELS STATEMENT
003600*                              SYSTEM DEPENDENCIES
003700*                              1  IBM-OS
003800*                              C  U-EXEC-8
003900*                      11-20   THE PROGRAM NAME WHICH IS TO BE
004000*                             ISOLATED FROM ALL THE OTHER
004100*                             SOURCE CODE.  REFER TO THE
004200*                             PROG-NUM-NAME TABLE IN THE
004300*                             PROGRAM FOR THE LIST OF VALID
004400*                             PROGRAM NAMES.
004500*              4. CHANGECP CAN BE USED TO BREAK OUT
004600*                 INDIVIDUAL SOURCE PROGRAMS.  THIS IS
004700*                 ACCOMPLISHED BY PUTTING THE NAME
004800*                 OF THE PROGRAM BEGINNING IN COLUMN
004900*                 11 OF THE INPUT CARD TO CHANGECP.
005000*                 THE·PRGIN AND PRGOUT DD STATEMENTS
005100*                 SHOULD BE CHANGED TO REFER TO THE
005200*                 SOURCE CODE DATA SET BEING ISOLATED.
005300*              5. ON UNIVAC SYSTEMS CHANGECP WILL GENERATE
005400*                 A ADD FILE WITH THE PROPER FILE AND COMPILER
005500*                 PROVIDED THAT THE FILE NAME AND COMPLIER CALL
005600*                 WAS PROVIDED ON THE CONTROL CARD.
005700*                 USERS SHOULD NOTE THAT THE COMPILER I OPTION
005800*                 IS ALWAYS USED IN THIS ADD FILE.
005900*                 EXTENDED UNIVAC CONTROL CARD FIELDS
006000*                     COLUMNS  DESCRIPTION
006100*                       21-32  FILE NAME WHERE CENSPAC SOURCE
006200*                              ROUTINES ARE TO RESIDE
006300*                       33-80  COMPILER CALL (EG. @ACOB OR
006400*                            . WHATEVER LOCAL CONVENTION IS USED)
006500               6. ON SYSTEMS RUNNING IBM DOS CHANGECP
006600*                 WILL GENERATE A CATAL S. AND BKEND RECORD BEFORE

006700                  EACH SOURCE PROGRAM, AND A BKEND RECORD AFTER
006800                  EACH SOURCE PROGRAM. THE GENERATED FILE CAN
006900                  THEN BE USED AS A SYSIPT FILE AND LOADED
007000                  TO THE SOURCE PROGRAM LIBRARY WITH THE
007100                  MAINT UTITITY PROGRAM.
007200*
007300*          CHARACTER SET CONVERSION AND COMPUTATIONAL USAGE
007400*          ARE NOT HANDLED AT ALL BY THIS PROGRAM.  CHANGES
007500*          FOR THESE TWO ITEMS ARE BETTER HANDLED BY
007600*          A  TEXT EDITOR.  IN PARTICULAR, DOUBLE QUOTES
007700*          ARE USED ON BURROUGHS AND DEC SYSTEMS (AND OTHERS)
007800*          WHILE SINGLE QUOTES ARE USED FOR IBM AND UNIVAC.
007900*          THE CENSPAC PROGRAM AS PROVIDED USE SINGLE QUOTES.
008000*          THE CHANGE FROM SINGLE QUOTES TO DOUBLE QUOTES CAN
008100*          BE ACCOMPLISHED BY A TEXT EDITOR COMMAND SIMILAR TO
008200*          THE FOLLOWING:
008300*
008400*                CHANGE /'/"/ ALL
008500*
```

For users at UNIVAC EXEC-8 installations, the CHANGECP file 3 program is to be used to convert the source code. The CHANGECP program will generate a run stream that is to be used to compile and map all executable programs into a library in one run.

Users of other operating systems will have to convert the CHANGECP file #2 program to run on their system. Comments are provided in the source code to assist in the conversion. This converted CHANGECP program will be used to separate SOURCE1 file into program modules and convert SOURCE1 and MODEL for use on their system as specified in Step 3.

Step 3: Modifying source code and data files

Operating instructions for the CHANGECP are provided on the previous pages and in the CHANGECP program itself. Users on IBM OS or VS systems usually should not use the CHANGECP program. Users on UNIVAC EXEC-8 IBM DOS, Burroughs 7700, or other systems will have to use CHANGECP to convert the CENSPAC source programs from the IBM OS/VS version provided on the tape to a version that will be compatible with their system.

Enclosed in the user's CENSPAC material, and periodically mailed to users by the Census Bureau, are User Notes. These describe problems and fixes that were reported to the CENSPAC development staff. The user should make the changes described in these memos, storing the resulting files for later use in the installation procedure. All applicable User Notes are provided with CENSPAC manuals when distributed by the Census Bureau. User Notes 1 thru 6 were issued before the release of CENSPAC Version 3 and do not apply to Version 3.

Step 4: Compile and save executable files and data files

Files 5-12 of the system tape are data files which may be used from either tape or from disk storage volumes during the installations procedure.

The CENSPAC SOURCE file contains three executable program modules (CENSPAC, CENGEN, and DOCUMENT) and a library of I/O modules for use at run-time. SOURCE1 is organized so that all programs and modules can be compiled and link-edited in one pass when using an IBM OS/VS system with the 1974 ANSI COBOL compiler with the BATCH option (Version 4 or later) or the EXEC-8 operating system and CHANGECP. The output should be link-edited (or 'MAP'ed) and saved as an executable library.

For IBM OS/VS users, the JCL provided in INSTJCL, will compile and linkedit all programs at one time. The resulting load modules will not contain overlays and the largest program, CENSPAC, will be about 210K in size. Linkage editor control statements are provided in the following pages that can be used to re-linkedit the CENSPAC and Documentor programs with overlays. The resulting programs will require less than 140K storage.

For UNIVAC EXEC-8 users, the results of running the CHANGECP program will be a runstream that can be @ADDed to compile all the programs. The MAP instructions included on the following pages can be used to MAP the CENSPAC and Documentor programs with overlays.

Burroughs users are to use the second source file, SOURCE2. The programs CENSPAC, CENGEN, DOCUMENTOR and the 12 I/O modules are to be saved in object form for execution. The I/O modules will be 'bound' with the generated EXECUTOR program during CENSPAC runs.

DOS and other system users may compile all programs if the COBOL compiler being used has the BATCH option. If the BATCH option is not available, the program modules will have to be separated using CHANGECP or an editor and individually compiled.

Step 5: Initialize the Data Dictionary

The data dictionary files used with the August 1980 version (Version 2) of CENSPAC can be used with the January 1982 version of CENSPAC. If the Version 2 data dictionary is used, error message text, the HELP command and the library function will not be available (the information for these functions is stored in the 'SYSTEXT' and 'MACLIB' dictionaries). If the Version 2 data dictionary is not available then a new data dictionary file must be created.

To initialize the Version 3 data dictionary file, the CENSDOC procedure (included in file INSTJCL and provided on the following pages) should be run with input from the DICTION1 file. This will allocate and initialize the data dictionary file with data dictionaries for SYSTEXT, MACLIB, STF1 and STF2. The resulting data dictionary file is used as a direct access file. Enough space for the data dictionary file will have to be provided for to contain the data dictionaries that are to be made available for CENSPAC. 30 tracks of 3350 equivalent space is adequate for the DICTION1 data dictionaries. 30 tracks will provide space for 450 data dictionary records. About 80 tracks will be required to contain all data dictionaries included on the CENSPAC tape.

Appendix 5 lists all data dictionaries available on the CENSPAC tape as of January 1982. The amount of space required for these data dictionary files is listed in the table.

The data dictionary information in the DICTION file contains upper and lower case characters. For those systems not able to handle lower case letters, the data dictionary definition statements as provided on the CENSPAC tape (DICTION, DICTION2, and DICTION3) will have to be converted to all upper case. This can be accomplished by using an editor to convert each lower case letter to the corresponding upper case letter. For example, an editor command such as: CHANGE /a/ to /A/ IN ALL, will convert the a's. This can be repeated for the entire alphabet to complete the conversion. Another method to accomplish this conversion would be to write a program that accomplishes the equivalent of the COBOL INSPECT verb which can translate individual lower case letters to the corresponding upper case letter.

The CENSPAC Documentor program uses a file initialization technique for the data dictionary file that attempts to use all available space when initializing the data dictionary file. The amount of space actually made available to this file is usually limited by space allocation information through JCL or RUNSTREAM or other system control language. If this space allocation procedure does not operate correctly on your operating system (systems with known problems on this area include IBM CMS and Control Data Corporation - CDC), the block parameter of the NEW option of the DOCUMENT command must be used to have the Documentor program limit the space it will utilize.

The printed output from this run should be compared with the DICTLIST file. A character-by-character comparison is not necessary, but the following should be checked:

1) All data names and table names in the DICTLIST report should also appear in the test run.

2) The beginning locations and length of all data items and tables should be the same in both reports.

3) The number of pages should be equal, and the TEXT and FOOTNOTES should appear in the same locations and formats in both reports.

Once the above run has been verified for correctness, the DOCUMENTOR can be run again using the same master data dictionary that has just been created using the file DICTION2 as input. This will add data dictionary information for the 1970 First, Second, Fourth, and Fifth Counts, and for the PL 94-171 reapportionment files (CNT1, CNT2, CNT4, CNT5, and PL94171 respectively). Similarly, the file DICTION3 can be processed to add dictionaries for the remaining dictionaries listed in Appendix 5.

Note to DOS Users: A significant difference for the DOS CENSPAC system is the use of Indexed Sequential Access Method (ISAM) for the data dictionary file and CUF utility file. On other operating systems a relative file structure is used but due to the lack of the relative file technique in older COBOL compilers the ISAM technique was used instead of relative access. Both file structures support the random access needed by CENSPAC for the data dictionary file and CUF utility file.

Step 6: Initialize CENSPAC MODEL file

The MODEL initialization process reduces the MODEL file as provided on the system tape to a smaller file with some computational format changes in the records. This conversion is done by invoking CENGEN using the JCL and data given in the file INSTJCL. This JCL for OS/VS, DOS, EXEC-8 and B7700 users is also listed on the following pages. This initialization will reduce storage and CPU requirements for the CENGEN program. The output of this step must be retained for later use by CENSPAC runs.

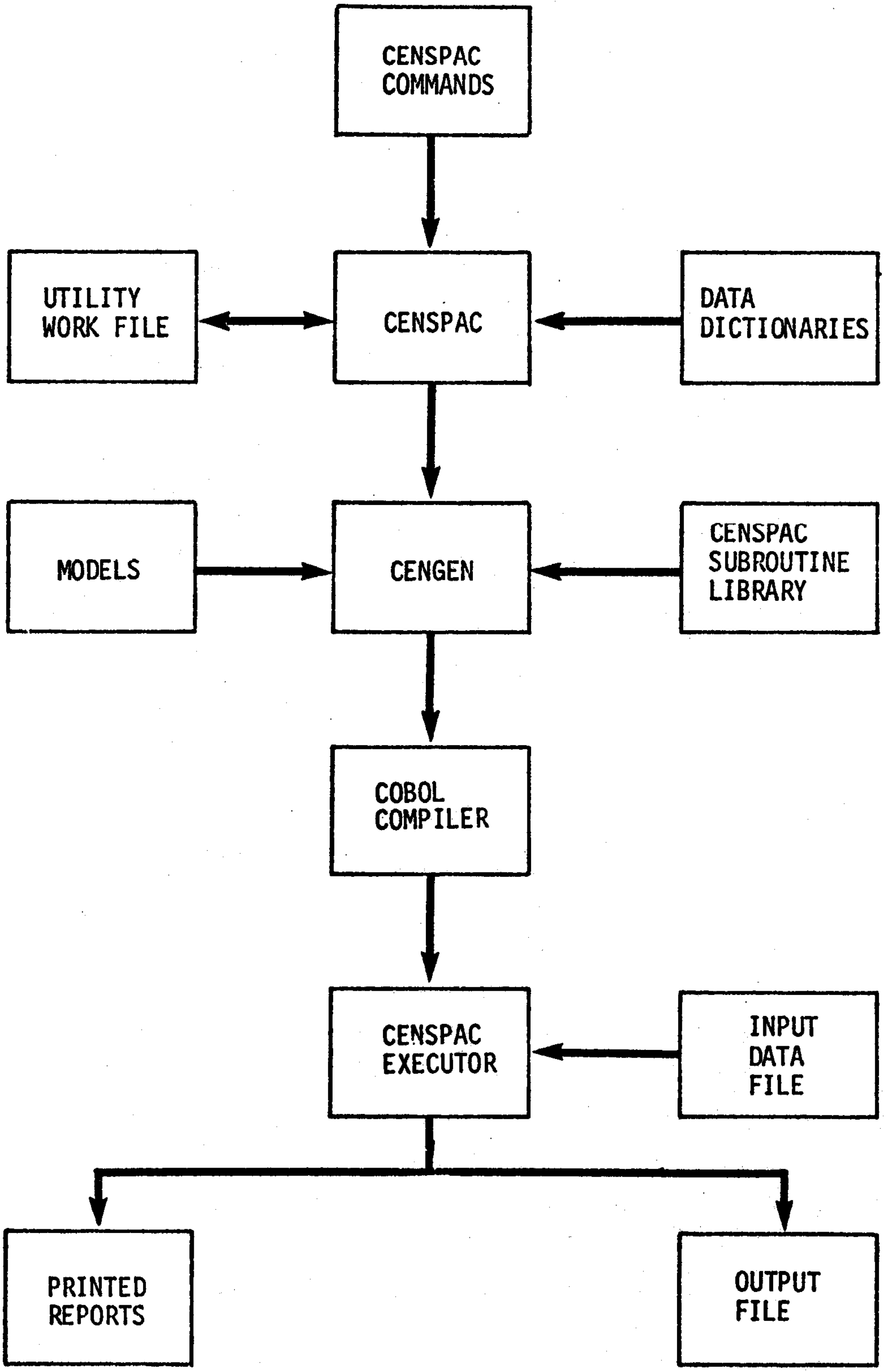
Step 7: Executing CENSPAC Test Runs

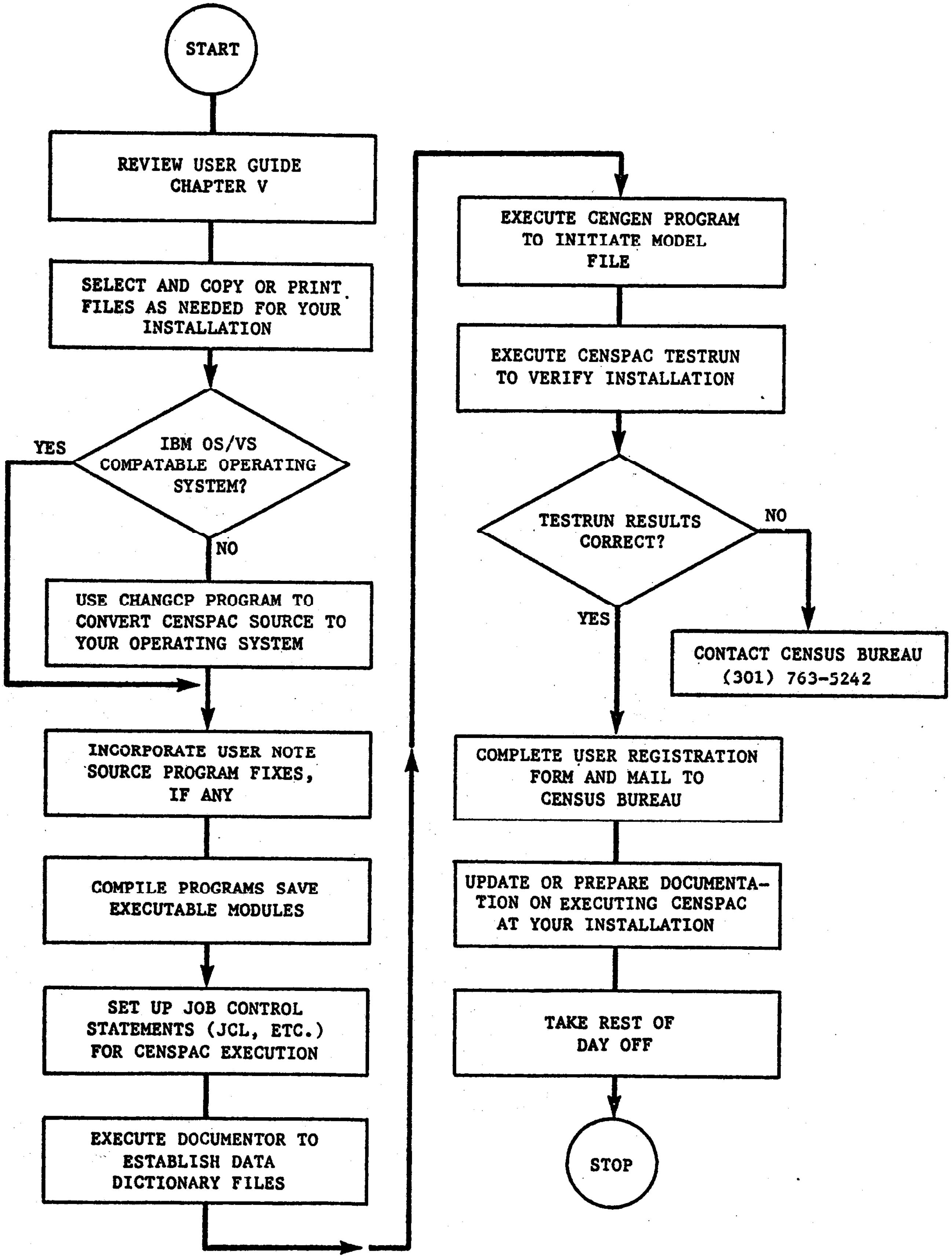
To test CENSPAC, the CENSRUN JCL procedure in file INSTJCL should be run with file #9, TESTRUN, as the input command stream and with file 6, TESTDATA, as test data file. The print output from this run should be compared with the TESTPRT file to verify the correctness of the installation of CENSPAC. For IBM Time Sharing Option (TSO), a CLIST for executing CENSPAC is also provided with the JCL on the following pages. This CLIST can be used to execute CENSPAC from an interactive terminal.

Step 8: Registration

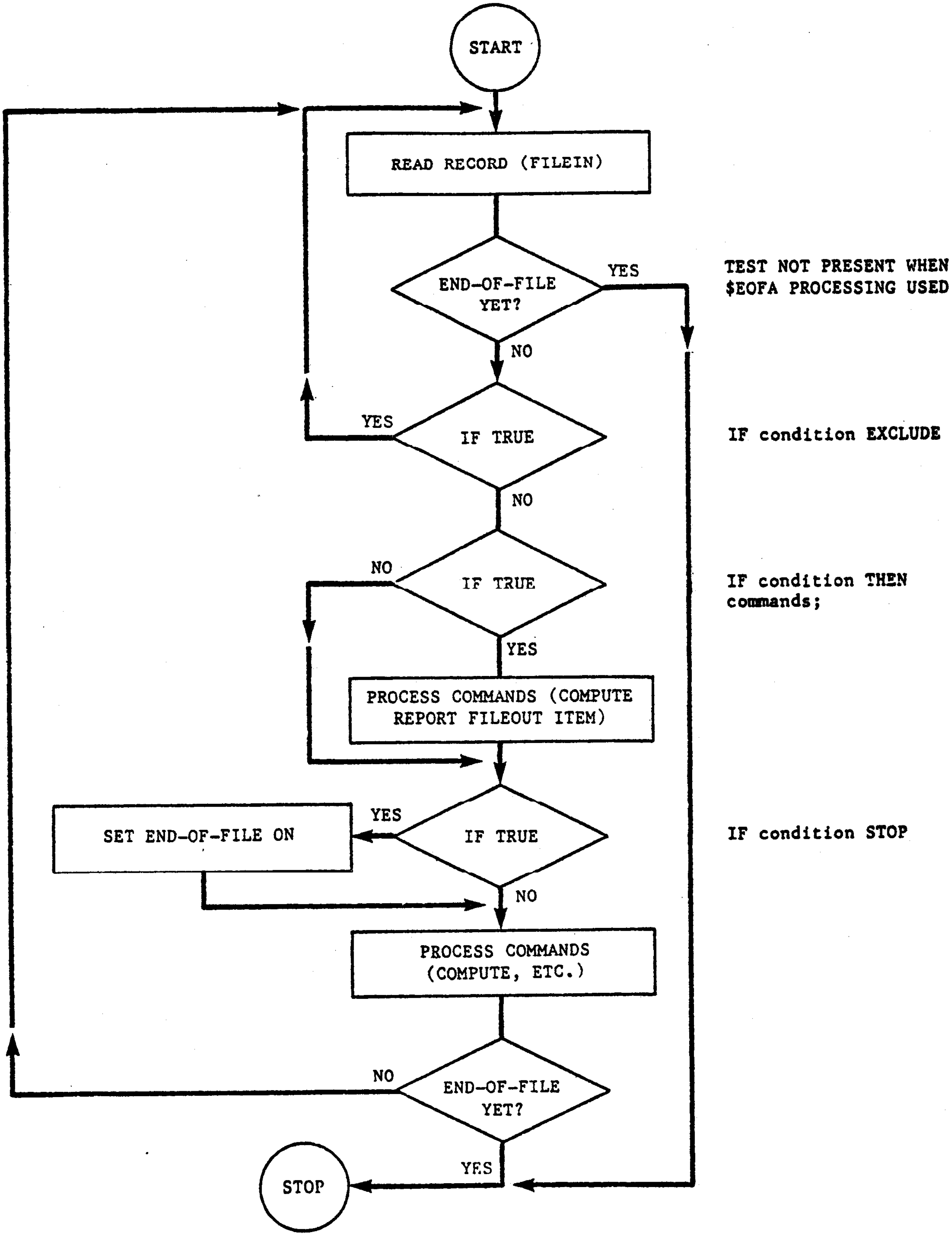
Complete the registration form provided in this manual on page 7. Return it to the CENSPAC development staff at the Census Bureau. This will take just a few minutes to complete, and it will assist the CENSPAC staff in its user support program. It will also insure that you are on the Bureau's mailing list for software news.

CENSUS SOFTWARE PACKAGE (CENSPAC) SYSTEM DIAGRAM

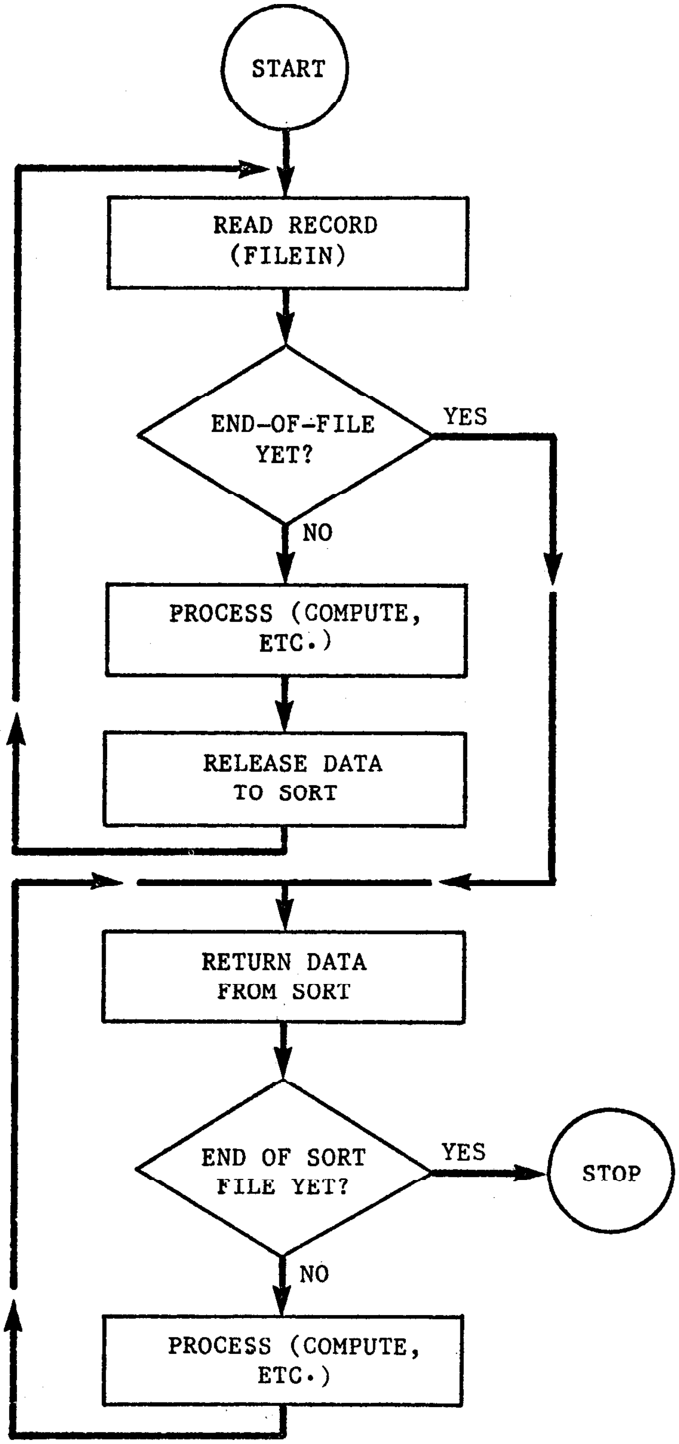



START
REVIEW USER GUIDE CHAPTER V
SELECT AND COPY OR PRINT FILES AS NEEDED FOR YOUR INSTALLATION
IBM OS/VS COMPATABLE OPERATING SYSTEM?
YES
NO
USE CHANGCP PROGRAM TO CONVERT CENSPAC SOURCE TO YOUR OPERATING SYSTEM
INCORPORATE USER NOTE SOURCE PROGRAM FIXES, IF ANY
COMPILE PROGRAMS SAVE EXECUTABLE MODULES
SET UP JOB CONTROL STATEMENTS (JCL, ETC.) FOR CENSPAC EXECUTION
EXECUTE DOCUMENTOR TO ESTABLISH DATA DICTIONARY FILES
EXECUTE CENGEN PROGRAM TO INITIATE MODEL FILE
EXECUTE CENSPAC TESTRUN TO VERIFY INSTALLATION
TESTRUN RESULTS CORRECT?
NO
YES
CONTACT CENSUS BUREAU (301) 763-5242
COMPLETE USER REGISTRATION FORM AND MAIL TO CENSUS BUREAU
UPDATE OR PREPARE DOCUMENTATION ON EXECUTING CENSPAC AT YOUR INSTALLATION
TAKE REST OF DAY OFF
STOP

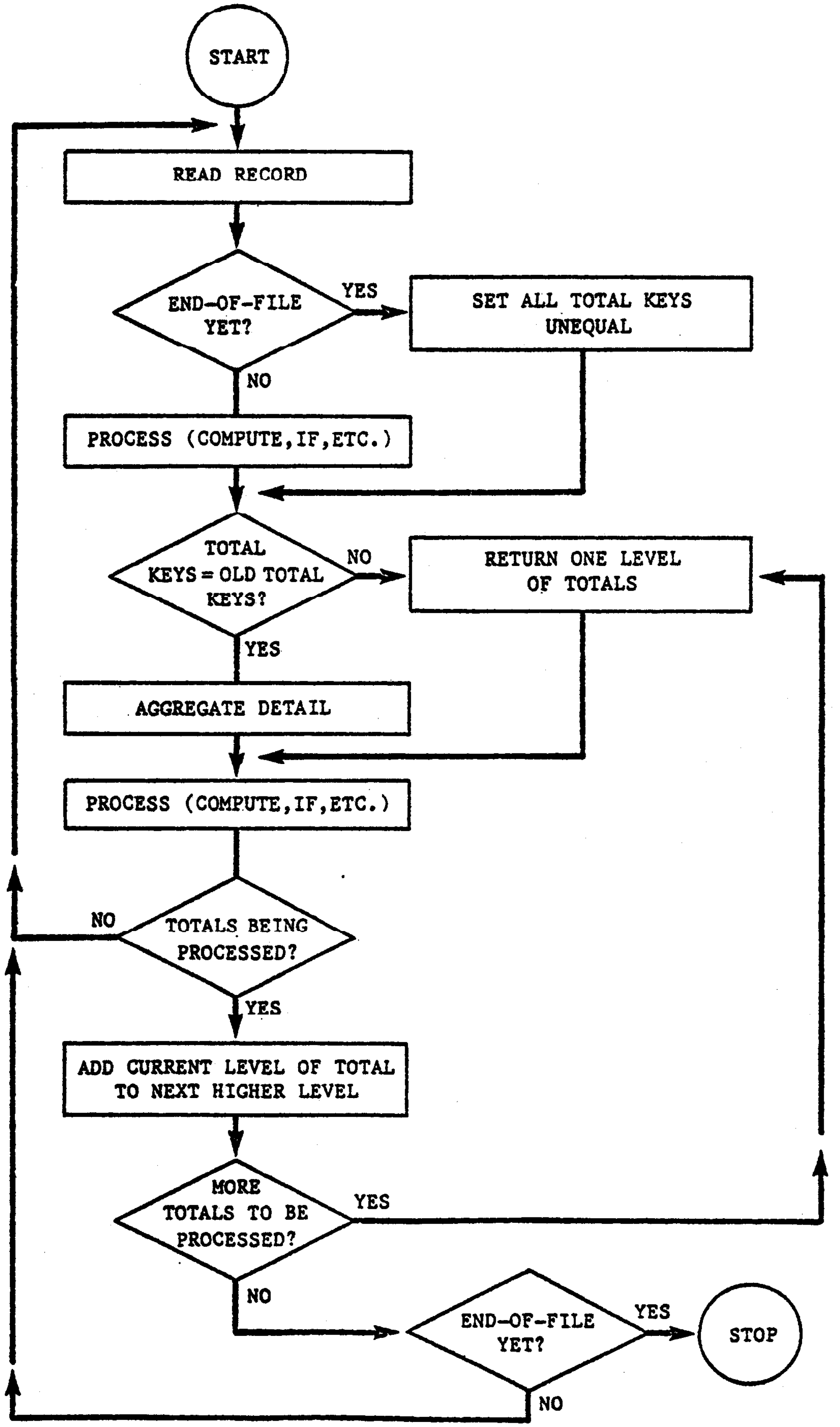
CENSPAC RECORD PROCESSING LOGIC



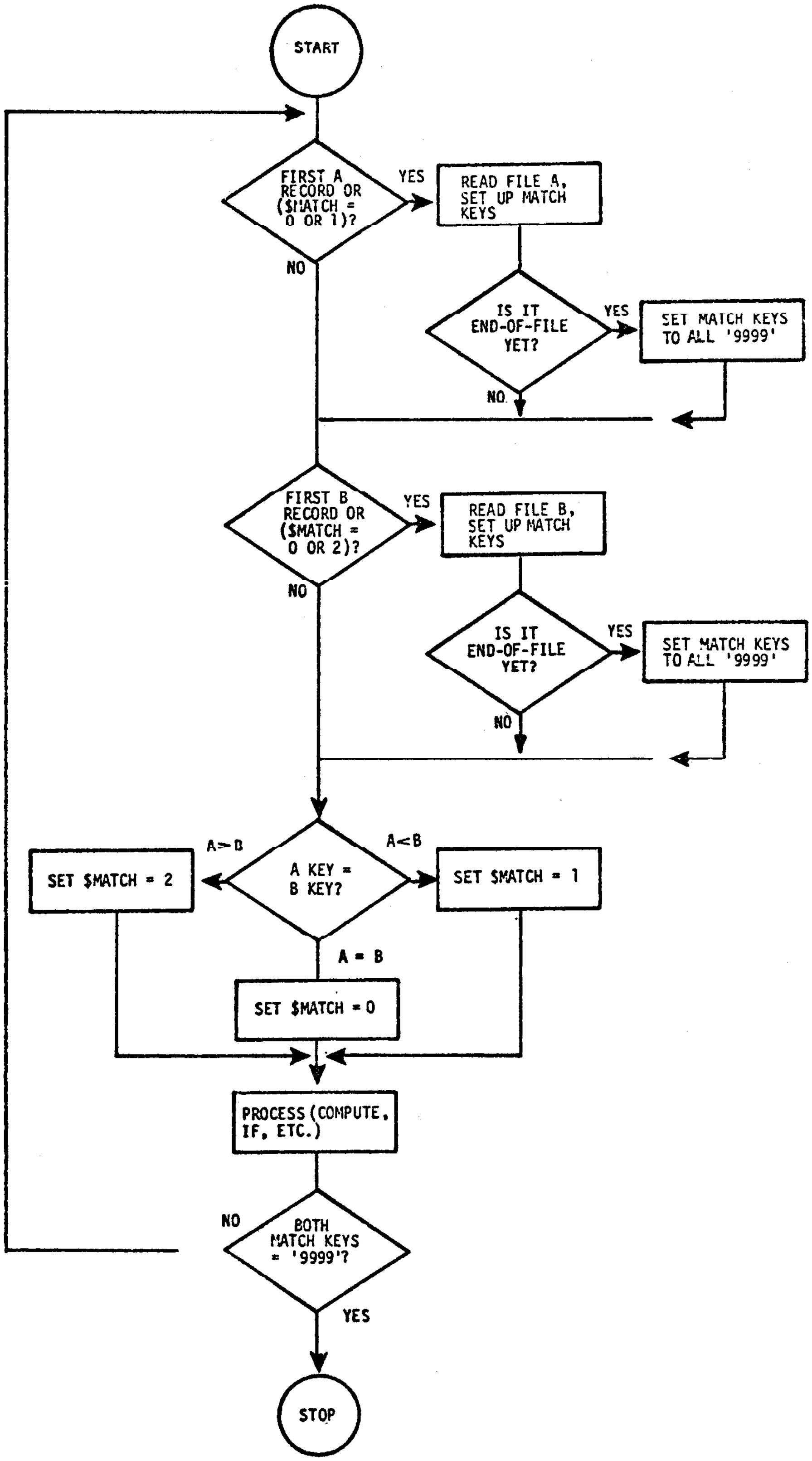
CENSPAC RECORD PROCESSING WITH SORT



CENSPAC RECORD PROCESSING WITH TOTAL


START
READ RECORD
END-OF-FILE YET?
YES
SET ALL TOTAL KEYS UNEQUAL
NO
PROCESS (COMPUTE,IF,ETC.)
TOTAL KEYS = OLD TOTAL KEYS?
NO
RETURN ONE LEVEL OF TOTALS
YES
AGGREGATE DETAIL
PROCESS (COMPUTE,IF,ETC.)
NO
TOTALS BEING PROCESSED?
YES
ADD CURRENT LEVEL OF TOTAL TO NEXT HIGHER LEVEL
MORE TOTALS TO BE PROCESSED?
YES
NO
END-OF-FILE YET?
YES
STOP
NO

CENSPAC RECORD PROCESSING WITH MATCH


START
FIRST A RECORD OR ($MATCH = 0 OR 1)?
YES
READ FILE A, SET UP MATCH KEYS
NO
IS IT END-OF-FILE YET?
YES
SET MATCH KEYS TO ALL '9999'
NO
FIRST B RECORD OR ($MATCH = 0 OR 2)?
YES
READ FILE B, SET UP MATCH KEYS
NO
IS IT END-OF-FILE YET?
YES
SET MATCH KEYS TO ALL '9999'
NO
A>B
A<B
SET $MATCH = 2
A KEY = B KEY?
SET $MATCH = 1
A = B
SET $MATCH = 0
PROCESS (COMPUTE, IF, ETC.)
NO
BOTH MATCH KEYS = '9999'?
YES
STOP

Logical Files Used in CENSPAC

Job Control Statements

PROCEDURE	executable-pgm name	DDNAME	DOS LOGICAL UNIT	Comments
CENSDOC	DOCUMENT	DDF	SYS012	Direct access file, unblocked records of 1024 characters. This is the data dictionary file.
	DOCUMENT	TOK	SYS010	Work file
	DOCUMENT	PAR	SYS004	input command file
	DOCUMENT	SYSOUT	SYSLST	printed output (system error messages, sort messages)
	DOCUMENT	ERR	SYS005	printed output (Documentor command listing and error messages
	DOCUMENT	FORM	SYS006	printed output (data dictionary report)
CENSRUN	CENSPAC	SYSPRINT	SYS005	printed output (command listing)
	CENSPAC	SYSIN	SYS004	input command file
	CENSPAC	DDF	SYS012	data dictionary (direct access, read only)
	CENSPAC	DOC	SYS008	data dictionary definition when documentation is requested with FILEOUT
	CENSPAC	FRAGTAPE	SYS006	work file, input to CENGEN
	CENSPAC	CUFILE	SYS013	direct access work file
	CENGEN	MODTAPE	SYS004	MODEL file
	CENGEN	SYSIN	SYS007	MODEL input file (used only when initializing the MODELS during installation.
	CENGEN	FRAGTAPE	SYS006	input from CENSPAC
	CENGEN	GENTAPE		EXECUTOR program source file input to COBOL compiler
	COBOL			COBOL compiler, compiles EXECUTOR program from previous step. Uses all standard work files associated with the standard system COBOL compiler.

CENSRUN (CONT)	LINKAGE EDITOR (MAP, BINDER, etc.)			Standard system programs to prepare compiled output for execution. All standard work files associated with this program are used. Main input is from the compile step, additional subroutines are also required when using data files requiring I/O modules (STFs, 1970 CNTs. Others as developed by local installations)
	GO	AINP	SYS006	input 'A' file
	GO	BINP	SYS007	input 'B' file
	GO	R001/PRINT$	SYS010	printed output, first report
	GO	R002	SYS011	printed output, second report, additional reports will require additional file assignments, R003, R004, etc.
	GO	F001	SYS008	output file, first FILEOUT command
	GO	F002	SYS009	output file, second FILEOUT command, additional output files will require additional file assignments F003, F004, etc.
	GO	WANP	SYS004	scratch for sorting input file 'A' (used only when SORT option of MATCH is used)
	GO	WBNP	SYS005	scratch for sorting input file 'B' (used only when SORT option of MATCH is used)

TBM OS/VS Job Control Language (JCL) for the CENSPAC Documentor

```
//DOCUMENT JOB (ACCOUNTING INFORMATION),REGION=150K
//*************************************************************
//*                    THIS PROGRAM JCL WILL EXECUTE THE DOCUMENTOR
//*                    SYSTEM.  THE INPUT TO DOCUMENTOR IS READ
//*                    FROM THE SYSIN DD STATEMENT.
//*                    SYSIN DD * CAN BE USED OR
//*                    THE SYSIN DD STATEMENT CAN IDENTIFY THE 'DICTION'
//*                    FILE DIRECTLY FROM TAPE.
//************************************************************
//STEP1     EXEC  PGM=IEFBR14
//*  THIS IEFBR14 STEP IS TO ALLOCATE A NEW DATA DICTIONARY MASTER FILE
//DICTION   DD  DSN=CENSPAC.DDF,DISP=(NEW,CATLG),
//             SPACE=(TRK,(20)),UNIT=3350,VOL=SER=SCRTS1
//CNNSDOC   PROC DDF='CENSPAC.DDF',OUTPUT=A,LOADLIB='CENSPAC.LOAD'
//S1        EXEC PGM=DOCUMENT
//STEPLIB   DD DSN=&LOADLIB,DISP=SHR
//DDF       DD DSN=&DDF,DISP=(OLD,KEEP)
//ERR       DD SYSOUT=&OUTPUT
//TOK       DD UNIT=SYSDA,SPACE=(TRK,(10,10)),
//             DCB=(LRECL=80,BLKSIZE=3120)
//FORM      DD SYSOUT=&OUTPUT
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//PAR       DD DDNAME=SYSIN
//SYSOUT    DD SYSOUT=&OUTPUT
//SYSPRINT DD SYSOUT=&OUTPUT
//          PEND
//STEP2     EXEC CNNSDOC
//SYSIN     DD  DSN=CNN.KXS.DICTIONX.FILEXXXX,DISP=(OLD,KEEP),
//        VOL=SER=XXXXXX,LABEL=(7,SL),UNIT=TAPE
/*
//
```

IBM OS/VS Job Control Language (JCL) to Initialize the CENSPAC MODEL file

```
//MODINIT  JOB (ACCOUNTING INFO)
//*.  THIS JCL INITIALIZES THE MODELS FILE
//*    INPUT IS THE MODELS FILE FROM THE RELEASE TAPE
//*     OUTPUT IS A DISK VERSION OF THE MODELS TO
//*     BE USED FOR CENSPAC RUNS
//*
//GEN   EXEC PGM=CENGEN,REGION=50K,COND=(8,LT)
//STEPLIB  DD DSN=CENSPAC.LOAD,DISP=SHR
//FRAGTAPE DD  *
CONTROL G000A
/*
//MODTAPE  DD  SPACE=(TRK,(2,2)),UNIT=3350,
//            DSN=MODEL.DATA,DISP=(NEW,KEEP),VOL=SER=XXXXXX
//*
//GENTAPE  DD  DSN=&GENOUT,DISP=(,PASS),UNIT=SYSDA,
//  SPACE=(TRK,(2,4))
//SYSIN    DD   DSN=MODEL.COBOL,DISP=(OLD,KEEP),
//         UNIT=TAPE,LABEL=(5,SL),VOL=SER=XXXXXX
//*          THE MODEL FILE CONTAINS COBOL STATEMENT SKELETONS
//*          AND IS MERGED WITH THE FRAGS TO PRODUCE THE
//*           CENSPAC EXECUTOR
// SYSPRINT DD SYSOUT=A
// SYSOUT   DD SYSOUT=A
```

Note: The 'FRAGTAPE' input file expects to read 20 character records. If the above JCL does not execute correctly due to this record size discrepancy, the IEBGENER utility program can be used to field select and prepare a file with the 'CONTROL' record in a file as a 20 character record.

IBM OS/VS Job Control Language (JCL) for CENSPAC

```
//CENSPAC JOB (ACCOUNTING INFO)
//*
//*
//**********************************************************
//*                 THIS IS THE JCL USED TO RUN THE CENSPAC
//*                 SYSTEM.  JCL CHANGES WILL HAVE TO BE MADE AS PER
//*                 EACH FACILITY ON WHICH CENSPAC IS BEING INSTALLED
//***********************************************************
//CENSRUN PROC DDMASTR='CENSPAC.DDF',
//         ADSN='STF1A',
//         AVOL=,
//         AUNIT=,
//         ALAB=,
//         BDSN='&DUMMY',
//         BVOL=,
//         BUNIT=SYSDA,
//         BLAB=,
//         BDISP=,
//         OCLASS=A,
//         LOADLIB='CENSPAC.LOAD',
//         COBPARM='(NOSOURCE,''BUF=10K'')'
//*
//PAC  EXEC PGM=CENSPAC,REGION=250K
//*    REGION SIZE CAN BE REDUCED BY USING OVERLAY AS
//*    SPECIFIED IN THE INSTALLATION GUIDE
//STEPLIB  DD  DSN=&LOADLIB,DISP=SHR
//SYSIN DD DDNAME=SYSIN
//DDF    DD  DSN=&DDMASTR,DISP=SHR
//*        FILE DDF IS THE MASTER DATA DICTIONARY FILE
//*             CREATED BY CENSPAC DOCUMENTOR
//SYSPRINT DD SYSOUT=&OCLASS
//FRAGTAPE DD  DSN=&FRAGX,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=(TRK,(2,4))
//*        FRAGTAPE IS A WORK FILE GENERATED BY CENSPAC
//*        AND PASSED TO CENGEN.  IT CONTAINS THE 'FRAGS'
//*        OR FRAGMENTS OF A COBOL PROGRAM.
//DOC     DD  DSN=&DOCOUT,DISP=(,PASS),UNIT=SYSDA,SPACE=(TRK,(1))
//*   DOC DD IS FOR OUTPUT FILE DOCUMENTATION
//CUFILE   DD  UNIT=SYSDA,SPACE=(TRK,(20))
//*    CENSPAC WORK FILE FOR DATA NAMES, LABELS,ETC.
//GEN   EXEC PGM=CENGEN,REGION=50K,COND=(8,LT)
//STEPLIB  DD DSN=&LOADLIB,DISP=SHR
//FRAGTAPE DD  DSN=&FRAGX,DISP=(OLD,DELETE)
//MODTAPE  DD  DSN=MODEL.DATA,DISP=SHR
//*     THIS IS THE OUTPUT FROM THE INITIALIZATION  JOB
//*       USING THE CENSPAC RELEASE TAPE MODELS AND CENGEN
//*
//*      GENTAPE  IS THE GENERATED COBOL PROGRAM
//*      PRODUCED BY CENGEN
//*
//GENTAPE  DD  DSN=&GENOUT,DISP=(,PASS),UNIT=SYSDA,
//  SPACE=(TRK,(2,4))
//SYSIN  DD  DUMMY
//SYSPRINT DD DUMMY,DCB=BLKSIZE=121
//SYSOUT   DD  DUMMY,DCB=BLKSIZE=121
//COB EXEC PGM=IKFCBL00,REGION=106K,COND=(8,LT),PARM=&COBPARM
//*        NOTE *********************************************
//*          'PCC.COBVSLIB' IS THE SYSTEM COBOL LIBRARY
//*           NEEDED FOR COBOL COMPILATION AND LINK EDIT
//*          IT MUST BE CHANGED TO REFLECT THE PROPER LIBRARY
//*          ON THE HOST COMPUTER SYSTEM  ***************
//*           CHANGE APPLIES TO COMPILE, LINK EDIT AND GO STEPS
//STEPLIB  DD DSN=PCC.COBVSLIB,DISP=SHR
//SYSIN    DD  DSN=&GENOUT,DISP=(OLD,DELETE)
//SYSPRINT DD DUMMY
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSLIN   DD DSN=&&LOADSET,UNIT=SYSDA,SPACE=(CYL,(1,1)),
//          DISP=(MOD,PASS),DCB=BLKSIZE=800
//LKED EXEC PGM=IEWL,COND=(8,LT)
//SYSPRINT DD DUMMY
//SYSLIB   DD DSN=PCC.COBVSLIB,DISP=SHR
//         DD DSN=&LOADLIB,DISP=SHR
//*        CENSPAC I/O MODULE LIBRARY
//*        THIS LIBRARY IS TO BE CONCATENATED
//*        WITH OTHER NEEDED LIBRARY(IES?) FOR
//*        COBOL LINK
//*
//SYSLIN   DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//SYSLMOD  DD  DSN=&&&GODATA(RUN),UNIT=SYSDA,SPACE=(CYL,(1,1,1)),
//          DISP=(,PASS)
//SYSUT1   DD UNIT=(SYSDA,SEP=(SYSLIN,SYSLMOD)),SPACE=(CYL,(1,1))
```

```
//GO EXEC PGM=*.LKED.SYSLMOD,COND=(8,LT)
//STEPLIB  DD DSN=PCC.COBVSLIB,DISP=SHR
//SYSLMOD  DD DSN=&&GODATA,DISP=(OLD,DELETE)
//SYSPRINT DD  SYSOUT=&OCLASS.
//SYSOUT   DD SYSOUT=&OCLASS.
//SORTWK01 DD  SPACE=(CYL,(1),,CONTIG),UNIT=3350
//SORTWK02 DD  SPACE=(CYL,(1),,CONTIG),UNIT=3350
//SORTWK03 DD  SPACE=(CYL,(1),,CONTIG),UNIT=3350
//*
//*     FILE 'AINP' IS THE 'A' INPUT FILE
//*     FILE 'BINP' IS THE 'B' INPUT FILE
//*     FILE 'R001' IS THE PRINT FILE FOR THE FIRST REPORT

//*     FILE 'R002' IS THE PRINT FILE FOR THE SECOND REPORT
//*     FILE 'F001' IS THE FILE FOR THE FIRST FILEOUT CMD
//*     FILE 'F002' IS THE FILE FOR THE SECOND FILEOUT CMD
//*     FILE 'WANP' IS A WORK FILE FOR SORTING THE 'A'
//*        FILE WHEN THE SORT OPTION IS USED ON THE MATCH
//*        COMMAND FOR THE 'A' FILE
//*     FILE  'WBNP' IS A SIMILIAR WORK FILE FOR FILE 'B' WITH
//*        MATCH AND SORT OPTION
//*
//AINP      DD  DSN=&ADSN,DISP=SHR,VOL=SER=&AVOL,UNIT=&AUNIT,
//         LABEL=&ALAB
//BINP      DD  DSN=&BDSN,DISP=&BDISP,VOL=SER=&BVOL,UNIT=&BUNIT,
//         LABEL=&BLAB,SPACE=(TRK,(1))
//R001      DD  SYSOUT=&OCLASS
//R002      DD  SYSOUT=&OCLASS
//*
//*     THE OUTPUT DATA FOR F001 AND F002 ARE ROUTED TO THE
//*     PRINTER FOR TEST PURPOSES ONLY.  IN A PRODUCTION
//*     ENVIRONMENT, THE JCL TO DEFINE THE OUTPUT DATA SET(S)
//*     SHOULD BE INCLUDED HERE.
//*
//F001      DD  SYSOUT=&OCLASS
//F002      DD  SYSOUT=&OCLASS
//WANP      DD  UNIT=SYSDA,SPACE=(CYL,(10))
//WBNP      DD  UNIT=SYSDA,SPACE=(CYL,(10))
// PEND
//CENSPAC EXEC CENSRUN,ADSN='TESTDATA.XXXXXXXX',
//          AVOL=XXXXXX,ALAB=6,AUNIT=TAPE
//SYSIN     DD DSN=TESTRUNX.CONTROLX,UNIT=TAPE,
//            DISP=(OLD,KEEP),VOL=SER=XXXXXX,LABEL=(9,SL)
//
```

IBM OS/VS Time Sharing Option (TSO) Command List (CLIST) for CENSPAC

```
PROC 0 SYSIN(*)  -
       AFILE(STF1A.DATA)  BFILE(STF1A.DATA)
FREEALL  /* FULLY QUALIFIED NAMES MUST BE SPECIFED TO -
            OVERRIDE INPUT FILE SPECIFICATIONS */
ALLOC F(FRAGTAPE) NEW SP(10,1) TRACKS DELETE /* TEMP FILE-
                                                 FOR CENGEN */
ALLOC F(DDF) DA(CENSPAC.DDF) SHR
ALLOC F(DOC) DA(DOCUMENT.DATA) NEW SP(10) BL(19000)
ALLOC F(SYSOUT) DA(*)
ALLOC F(SYSPRINT) DA(*)
 ALLOC F(CUFILE)  SP(20) BL(19000) NEW
FREE F(SYSIN)
ALLOC F(SYSIN) DA(&SYSIN.) SHR
CALL CENSPAC.LOAD(CENSPAC)
IF &LASTCC = 16 THEN GOTO LAST
FREE F(SYSIN DDF) /* FREE INACTIVE FILES */
ALLOC F(MODFILE) DA(MODGEN.DATA) SHR
ALLOC F(GENTAPE) DA(GENPGM.COBOL) NEW SP(40,1) TRACKS DELETE
WRITE PLEASE WAIT.
WRITE A CUSTOMIZED PROGRAM IS BEING GENERATED.
WRITE
CALL CENSPAC.LOAD(CENGEN)
IF &LASTCC = 16 THEN GOTO LAST
FREE F(MODFILE FRAGTAPE)
ALLOC DA(GENPGM.OBJ) NEW SP(19,1) TRACKS DELETE
ALLOC F(R001) DA(*)
ALLOC F(R002) DA(*)
ALLOC F(F001) DA(*)
ALLOC F(SORTWK01) SP(10) BL(19000) NEW
ALLOC F(SORTWK02) SP(10) BL(19000) NEW
ALLOC F(SORTWK03) SP(10) BL(19000) NEW
ALLOC F(AINP) DA('&AFILE') SHR
ALLOC F(BINP) DA('&BFILE') SHR
COBOL GENPGM BUF(10K) FLAGE
FREE F(GENTAPE)
IF &LASTCC = 16 THEN GOTO LAST
ALLOC DA(GENPGM.LOAD) DIR(3) NEW SP(19,19) TRACKS
LINK GENPGM LIB(CENSPAC.LOAD) LET COBLIB LOAD(GENPGM(GENPGM))
WRITE ATTENTION.  GENPGM COMMENCING EXECUTION
CALL GENPGM.LOAD(GENPGM)
LAST:SET &LASTCC = 0
FREEALL
END
```

Linkage Editor Statements (IBM OS/VS) for the Documentor Using Overlays

```
//LKED EXEC PGM=HEWL, PARM='OVLY'
//MODLIB DD DSN=CENSPAC.OBJ,DISP=OLD
//SYSLIN DD *
INSERT DOCUMENT DDFIO DDFPARS
OVERLAY A
INSERT DDFGEN
OVERLAY A
INSERT DDFLIST
/*
//SYSLIB DD DSN=SYS1.COBLIB,DISP=SHR
//SYSLMOD DD DSN=CENSPAC.LOAD(DOCUMENT),DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(3,1))
```

Linkage Editor Statements (IBM OS/VS) for CENSPAC Using Overlays

```
//LKED EXEC PGM=HEWL,PARM='OVLY'
//MODLIB DD DSN=CENSPAC.OBJ,DISP=OLD
//SYSLIN DD
INSERT CENSPAC
INSERT PACIO
INSERT INTRFACE
INSERT GEN
OVERLAY A
INSERT ITEMSCMD
OVERLAY A
INSERT OUTCMD
INSERT CENSDOC
OVERLAY A
INSERT IFCMD
OVERLAY A
INSERT CENSLAB
/*
//SYSLIB DD DSN=SYS1.COBLIB,DISP=SHR
//SYSLMOD DD DSN=CENSPAC.LOAD(CENSPAC),DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(3,1))
```

UNIVAC EXEC-8 RUNSTREAM FOR CENSPAC

UNIVAC EXEC-8 RUNSTREAM TO EXECUTE CENSPAC

```
@ASG,T FRAGTAPE.,F2///2000
@ASG,T CUFILE.,F2///20
@ASG,A DATADICTFILE.
@USE DDF.,DATADICTFILE.
@ASG,T DOC.,F2///100
@XQT CENSPAC.CENSPAC
@ADD COMMANDS.          CENSPAC INPUT COMMANDS HERE
@TEST TNE/40/S5
@JUMP END
@ASG,A MODTAPE.GENERATED MODELS FILE FROM INSTALLATION RUN
@ASG,T GENTAPE
@XQT CENSPAC.CENGEN
@ASG,TJV AINP.,U9,099999         INPUT CENSUS DATA FILE
@ASG,A BINP.  USED FOR FILE B IN MATCH
@ACOB,FIE ,EXEC
@ADD GENTAPE.
@MAP,I ,EXEC
 IN EXEC
 LIB CENSPAC.
@ASG,T F001. FIRST FILEOUT FILE
@ASG,T R002. SECOND REPORT OUTPUT (FIRST GOES TO PRINT$)
@XQT EXEC
@END:
```

UNIVAC EXEC-8 RUNSTREAM FOR CENSPAC DOCUMENTOR

```
@RUN JOB ACCOUNTING INFORMATION
@ASG,A DATADICTFILE.
@ASG,T PAR.,F2///250
@ASG,UP FORM.,F2///500
@ASG,UP ERR.,F2///500
@DATA,I PAR.        ⎧Sample Documentor statements requesting a listing of the
DOCUMENT STF1 LIST  ⎨STF1 data dictionary.
END                 ⎩
@USE   DDF.,DATADICTFILE
@ASG,T TOK.,F2///4000
@XQT CENSPAC.DOCUMENTOR
@END
```

UNIVAC EXEC-8 MODEL INITIALIZATION RUNSTREAM

```
@RUN JOB ACCOUNTING INFORMATION
@ASG,UP MODTAPE.,F2///30 . GENERATED MODELS FILE
@ASG,T GENTAPE.
@ASG,A MODELS.MODELS FILE FROM RELEASE TAPE (CONVERTED TO UNIVAC 1100)
@ASG,T FRAGTAPE.
@DATA,Q FRAGTAPE
CONTROL GØØØA
@END
@XQT CENSPAC.CENGEN
@FIN
```

UNIVAC EXEC-8 CHANGCP RUNSTREAM TO CONVERT AND COMPILE CENSPAC

```
@RUN,M JOB ACCOUNTING INFORMATION
@LOG ANY LOG INFORMATION NECESSARY
@ASG,TJV SDFILE.,U9,INPUT TAPE # . THIS IS FOR EBCDIC TAPE
@ASG,UP OUTPUT.,F2///1000 . OUTPUT FILE FOR CPU CHANGE
@ASG,UP CENSPAC.,F2///1000 . PROGRAM FILE FOR OUTPUT OF COMPILER
@USE SDFILEO.,OUTPUT.
@MOVE,C SDFILE.,9  . MOVE TAPE TO SOURCE PROGRAM FILE
@XQT FILENAME.ELTNAME  . EXECUTE CPU CHANGE PROGRAM
U-EXEC-8C           CENSPAC     @ACOB,SF
@ . CONTROL CARD CONTAINS COMPUTER SYSTEM IN 1 TO 8
@ . PROGRAM FILE NAME IN 21-32
@ . COMPILER CALL INFORMATION IN 33-80 EG. @ACOB OR WHATEVER
@ADD.L OUTPUT.  . ADDING OUTPUT FILE WILL COMPILE ALL CENSPAC PROGRAMS
@ . ADDING OUTPUT WILL COMPILE ALL CENSPAC ROUTINES
@ . AND INPUT THEM INTO THE PROGRAM FILE SPECIFIED
@PREP CENSPAC.  . PREPPING FILE PREPARES ENTRY POINT TABLE
@MAP CENSPAC.CENSPAC
IN CENSPAC.CENSPAC
LIB CENSPAC.  . THIS LIB STATEMENT WILL INCLUDE ALL CENSPAC SUBROUTINES
LIB YOUR COBOL LIBRARY
@EOF
@FIN
```

UNIVAC EXEC-8 MAP Instructions for CENSPAC

```
@MAP CENSPAC.CENSPAC
IBANK,M PACIOINTRGEN
IN CENSPAC.CENSPAC
IN CENSPAC.PACIO
IN CENSPAC,INTRFACE
IN CENSPAC,GEN
IBANK,D ITEMS
IN CENSPAC.ITEMS
IBANK,D IFCMD,ITEMS
IN CENSPAC.IFCMD
IBANK,D OUT,ITEMS
IN CENSPAC.OUTCMD
IBANK,D LABEL,ITEMS
IN CENSPAC.CENSLAB
IBANK,D CENSDOC,ITEMS
IN CENSPAC,CENSDOC
DBANK,M FORM PACIOINTRGEN
LIB CENSPAC.
LIB SYS$*RLIB$
@EOF
```

UNIVAC EXEC-8 MAP for Instruction DOCUMENTOR

```
@MAP CENSPAC,DOCUMENTOR
IBANK,M MAINIO
IN CENSPAC.DOCUMENT
IN CENSPAC.DDFIO
IBANK,D PARS
IN CENSPAC.DDFGEN
IBANK,D GEN,PARS
IN CENSPAC.DDFGEN
IBANK,D LIST,PARS
IN CENSPAC.DDFLIST
DBANK,M FORM MAINIO
LIB CENSPAC.
LIB SYS$*RLIB$
@EOF
```

Burroughs 7700 Work Flow Language (WFL) to Execute CENSPAC

```
RUN NEWPAC; PRINTLIMIT=500;    %
FILE "CARD-INPUT"(KIND=PACK,FILETYPE=7,TITLE=TURKEY); %
FILE "DATA-DICTIONARY-FILE"(FILETYPE=7,TITLE=F13,KIND=PACK); %
FILE "CUF-INIT-FILE"(PROTECTION=SAVE,TITLE=CUF/FILE, %
KIND=PACK,MAXRECSIZE=14,BLOCKSIZE=420.AREASIZE=30,UNITS=WORDS); %
FILE "FRAG-FILE"(PROTECTION=SAVE,TITLE=CENSPAC/FRAGFILE, %
KIND=PACK,MAXRECSIZE=14.BLOCKSIZE=420,AREASIZE=30,UNITS=WORDS); %
FILE "SECONDARY-CUF"(FILETYPE=7,KIND=PACK,TITLE=CUF/FILE);  %
 FILE "PRNTR1"(KIND=REMOTE);  %
 REMOVE NEW/GENTAPE ;  %
 REMOVE NEW/ERROR;  %
 RUN CENGEN ;  %
 TYPE NEW/GENTAPE COBOL ;  %
COMPILE NEW/GENTAPE;
 BIND GENPROG AS XOT;
 RUN XQT; PRINTLIMIT=500;   %
 FILE FOO1FILE(KIND=PACK,TITLE=FOO1OUT.PROTECTION=SAVE); %
 FILE CORTFILE(KIND=PACK); %
 FILE ROO1FILE(KIND=REMOTE); %
 FILE FILEIN(KIND=PACK.FILETYPE=7.TITLE=F6);
```

IBM DOS Job Control Language (JCL)

CENSPAC CHANGCP Compile and Link-edit JCL

```
// JOB  CHANGECP   'ACCOUNTING INFO'    JOB CARD
*  SETP 1  CATALS AND BOOKEND ARE PLACED ON SYSIPT TAPE
*
// PAUSE MOUNT CENSPAC TAPE ON 191 SCRATCH WORK TAPE ON 190
// ASSGN SYS004,X'00C'                  THIS JOB STEP PUTS A BKEND CARD ON THE
// ASSGN SYS005,X'190'                  TAPE OUTPUT - SYSIPT FORMAT
// UPSI 00100
// EXEC CDTP
// UCT TC,FF,A=(80,80),B=(80,80)
// END
 CATALS C.CHANGECP
 BKEND  C.CHANGECP
/*
*  STEP 2  CHANGECP PROGRAM IS LOADED TO SYSIPT TAPE
*
// ASSGN SYS004,X'191'                  CENSPAC TAPE- THIS STEP COPYS THE
// TLBL UIN                             CHANGECP PROGRAM TO
// MTC FSF,SYS004,3
// ASSGN SYS005,X'190'                  TAPE OUT - SYSIPT FORMAT
// EXEC TPTP
// UTT TR,FF,A=(80,2000),B=(80,80)
// END
*  STEP 3  BKEND CARD  PLACED ON SYSIPT
*
// ASSGN SYS004,X'00C'                  THIS STEP PUTS BKEND ON
// ASSGN SYS005,X'190'                  SYSIPT TAPE
// EXEC  CDTP
// UCT TC,FF,A=(80,80),B=(80,80)
// END
 BKEND
/*
*  STEP 4  CHANGECP IS CATALOGED TO SOURCE STMT LIBRARY
*
// MTC REW,SYS004                       THIS STEP CATALOGS CHANGECP
// ASSGN SYSIPT,X'190'                  IN THE SOURCE STATEMENT LIBRARY
// EXEC MAINT
// RESET ALL
// MTC REW,X'190'
// MTC REW,X'191'
*  STEP 5  CHANGECP IS COMPILED AND LINK EDITED
*
// OPTION CATAL                         COMPILE AND LINK
 PHASE CHANGECP,*
// EXEC FCOBOL
 CBL LIB
BASIS CHANGECP
/*
// LBLTYP TAPE
// EXEC LNKEDT
*  STEP 6  CHANGECP IS EXECUTED TO INSERT CATALS & BKEND CARDS
*
// ASSGN SYS004,X'191'                  CENSPAC TAPE
// TLBL SYS004
// MTC FSF,SYS004,9
// ASSGN SYS005,X'190'                  SYSIPT OUT
// EXEC CHANGECP
IBM-DOS
/*
// MTC REW,SYS005
*  STEP 7  CENSPAC SOURCE IS CATALOGED INTO SOURCE STMT LIBRARY
*
// ASSGN SYSIPT,X'190'
// EXEC MAINT
*  STEP8  CHANGECP IS EXECUTED TO LOAD MODELS TO DISK FILE
*
// ASSGN SYS004,X'191'                  CENSPAC TAPE
// MTC REW,X'191'
// TLBL SYS004
// MTC FSF,SYS004,12
// ASSGN SYS006,X'233'
// DLBL SYS006,'DU.MODELS'
// EXTENT SYS006,DUSD01,,,1000,15
// EXEC CHANGECP
IBM-DOS I
/*
/&
```

Installation Initialization JCL

```
// JOB CENSINIT    'ACCOUNTING INFO'   CENSPAC INIT
// ASSGN SYS004,X'00C'
//ASSGN SYS005,X'231'
// DLBL SYS005,'DU.CENSPAC.FRAG'
// EXTENT SYS005,D99997,,,20,20
// EXEC CDDK
// UCD TR,FF,A=(80,80),B=(20,1000),E=(2314)
// END
CONTROL G000A
000000100000C
/*
// ASSGN SYS005,X'231'
// DLBL SYS005,'DU.GEN.COBOL'
// EXTENT SYS005,D99997,,,100,100
// ASSGN SYS006,'231'
// DLBL SYS006,'DU.CENSPAC.FRAG'
// EXTENT SYS006,D99997,,,20,20
// ASSGN SYS004,'233'
// DLBL SYS004,'DU.MODGEN'                    GENERATED MODELS FILE
// EXTENT SYS004,DUSD01,,,1200,12
// ASSGN SYS007,X'233'                        INPUT MODELS FILE
// DLBL SYS007,'DU.MODELS'
// EXTENT SYS007,DUSD01,,,1000,20
// ASSGN SYS011,X'233'
// DLBL CUFILE,'DU.CUFILE',,ISC
// EXTENT SYS011,DUSD01,4,1,360,10
// EXTENT SYS011,DUSD01,1,2,370,20
// EXEC CENGEN
/*
/&
```

CENSPAC JCL

```
// JOB CENSPAC     'ACCOUNT INFO'      CENSPAC JCL
// OPTION LINK
// ASSGN SYS012,X'233'
// DLBL DDFILE,'DU CENSPAC.DOF',,ISE
// EXTENT SYS012,DUSD01,4,1,20,20
// EXTENT SYS012,DUSD01,1,2,40,300
// ASSGN SYS004,'00C'
// ASSGN SYS005,'00E'
// ASSGN SYS006,'231'
// DLBL SYS006,'DU.CENSPAC.FRAG',0     WORK FILE
// EXTENT SYS006,D99997,,,20,20        UTILITY PACK -WORK PACK
// ASSGN SYS013,X'233'                 CENSPAC UTILITY FILE
// DLBL CUFILE,'DU.CUFILE,,ISE         ISAM WORK AREAS
// EXTENT SYS013,DUSD01,4,1,360,10
// EXTENT SYS013,DUSD01,1,2,370,20
// ASSGN SYS008,X'191'                 OUTPUT DOC FILE
// TLBL SYS008,'DU.CENSPAC.DOC'        DICTIONARY INPUT STATEMENTS
// EXEC CENSPAC
------------- CENSPAC COMMANDS  -------------------
/*
// RESET ALL
// ASSGN SYS005,X'231'
// DLBL SYS005,'DU.GEN.COBOL'
// EXTENT SYS005,D99997,,,100,100
// ASSGN SYS006,'231'
// DLBL SYS006,'DU.CENSPAC.FRAG'
// EXTENT SYS006,D99997,,,20,20
// ASSGN SYS004,'233'
// DLBL SYS004,'DU.MODGEN'             GENERATED MODELS FILE
// EXTENT SYS004,DUSD01,,,1200,12
// ASSGN SYS007,X'233'                 INPUT MODELS FILE
// DLBL SYS007,'DU.MODELS'
// EXTENT SYS007,DUSD01,,,1000,20
// ASSGN SYS011,X'233'
// DLBL CUFILE,'DU.CUFILE',,ISC
// EXTENT SYS011,DUSD01.4.1.360.10
// EXTENT SYS011,DUSD01,1,2,370,20
// EXEC CENGEN
/*
/+
// RESET SYS
// DLBL IJSYSIN,'DU.GEN.COBOL'
// EXTENT SYSIPT,D99997,,,100,100
   ASSGN SYSIPT,X'231'                 THIS IS A PERMANENT ASSGN
// EXEC FCOBOL
   CLOSE SYSIPT,X'00C'                 THIS CLOSE CLOSES THE ASSGN OF
// RESET SYS                           SYSIPT
// LBLTYP TAPE
// EXEC LNKEDT
// ASSGN SYS001,X'231'              USED IF SORT COMMAND WAS USED
// DLBL SYS001
// EXTENT SYS001,D99997,,,150,57
// ASSGN SYS002,X'231'
// DLBL SYS002
// EXTENT SYS002,D99997,,,210,57
// ASSGN SYS003,X'231'
// DLBL SYS003
// EXTENT SYS003,D99997,,,270,57
// ASSGN SYS004,X'231'
// DLBL SYS004                      USED IF SORT OPTION OF MATCH COMMAND
// EXTENT SYS004,D99997,,,340,57    IS USED
// ASSGN SYS005,X'231'
// DLBL SYS005
// EXTENT SYS005,D99997,,,400,57
// ASSGN SYS006                     INPUT FILE A ALWAYS USED
// TLBL                             EXTENTS AND LABELS AS NECESSARY
// ASSGN SYS007                     INPUT FILEB USED WHEN SECOND
// TLBL                             FILEIN COMMAND USED IF ON TAPE
// ASSGN SYS008                     OUTPUT FILE 1 USED WHEN FILEOUT
// DLBL  SYS008                     COMMAND IS USED
// EXTENT SYS008,                      FILL IN EXTENT INFORMATION
// ASSGN  SYS009,X'   '          USED IF SECOND FILEOUT COMMAND IS USED
// DLBL SYS009                   DLBL INFORMATION AS NECESSARY
// EXTENT SYS009                 EXTENT INFO
// ASSGN SYS010,X'00E'           USED IF REPORT COMMAND IS USED
// ASSGN SYS011,X'00E'           USED IF SECOND REPORT COMMAND IS USED
// EXEC
/*
/*
```

CENSPAC DOCUMENTOR Program JCL

```
// JOB 1DOCUMEN     'ACCOUNTING INFO'
* DOCUMENT PREPARES A INDEXED-SEQUENTIAL ACCESS METHOD FILE
*   IS USED AS A DATA DICTIONARY BY THE CENSPAC PROGRAM
// PAUSE MOUNT TAPE ON 193
// TLBL SYS004
// ASSGN SYS001,X'231'
// ASSGN SYS002,X'231'
// ASSGN SYS003,X'231'
// ASSGN SYS004,X'193'
// ASSGN SYS005,X'00E'
// ASSGN SYS006,X'00E'
// ASSGN SYS010.X'231'
// ASSGN SYS012,X'233'
// ASSGN SYS013,X'233'
// DLBL SORTWK1
// EXTENT SYS001,D99997,,,220,100
// DLBL SORTWK2
// EXTENT SYS002,D99997,,,20,100
// DLBL SORTWK3
// EXTENT SYS003,D99997,,,120,100
// DLBL SYS010'TOKENS WORK FILE',000
// EXTENT SYS010,D99997,,,320,100
// DLBL DDFILE2,'DU.CENSPAC.DDF',,ISC
// EXTENT SYS013,DUSD01,4,1,20,20
// EXTENT SYS013,DUSD01,1,2,40,300
// DLBL DDFILE,'DU.CENSPAC.DDF',,ISE
// EXTENT SYS012,DUSD01,4,1,20,20
// EXTENT SYS012,DUSD01,1,2.40.300
// EXEC DOCUMENT
/*
```

Appendix 1

CENSPAC ERROR MESSAGES

```
0001     GEN PHASE DATA TABLE OVERFLOW:
                 REDUCE NUMBER OF DATA OR
                 COMMAND REFERENCES OR
                 INCREASE THE SIZE OF THE
                 DATA TABLE.
0002     INVALID OPERAND:
             A.  EXPECTING VARIABLE NAME, BUT
                 FOUND RESERVED WORD; OR
             B.  AN EXISTING VARIABLE IS REQUIRED
                 FOR TOTAL.
0003     INVALID  REFERENCE:
          AT LEAST ONE VARIABLE MUST BE
          SPECIFIED, BUT NONE FOUND.
0004     INVALID COMPARISON:
          THE 'IF' COMMAND CONTAINS A
          CONDITION WHICH COMPARES TWO LITERALS.
0005     INVALID COMPARISON OR USAGE:
          COMPARISON BETWEEN ALPHANUMERIC AND
          NUMERIC ITEMS OR ALHANUMERIC ITEM
          REFERENCED IN THE COMPUTE OR TOTAL
          OPERATION.
0006     INVALID  REFERENCE:
          THE "TOTAL" COMMAND HAS A REFERENCE
          TO A VARIABLE WHOSE DATA TYPE IS "CODE"
          CATEGORY.
0007     INVALID VARIABLE USAGE:
          INVALID USE OF OBSERVATION
          VARIABLE IN THE FREQUENCY
          DISTRIBUTION.
0008     INVALID OPERAND:
          NUMERIC DATA ITEM REQUIRED.
0009     INVALID NUMERIC OPERAND.
          NUMERIC LITERAL > 18 CHARACTERS LONG
          OR CONTAINS MORE THAN 6 DECIMAL PLACES
0010     INVALID OPERAND, EXPECTING RESERVED
          WORD.
0011     UNIDENTIFIED COMMAND:
          THE CENSPAC SYSTEM DOES NOT
          RECOGNIZE THE PRINTED SYMBOL
          AS A COMMAND.
0012     NO CENSPAC COMMANDS FOUND:
          END OF CARD FILE OCCURRED BUT
          NO CENSPAC COMMANDS WERE INPUT
          OR PROCESSED.
0013     FILEIN ERROR:
             A.  MORE THAN 2 FILEIN
                 COMMANDS FOUND, OR
             B.  FIRST COMMAND NOT A FILEIN, OR
             C.  A SECOND FILEIN COMMAND WITH
                 NO CORRESPONDING MATCH COMMAND,
                 OR
             D.  NO DICTIONARY SPECIFIED.
0014     INVALID COMMAND SEQUENCE.
0015     DUPLICATE DEFINITION OR REFERENCE:
          DUPLICATE INPUT DATA DEFINITION OR
          AMBIGUOUS REFERENCE IN THE TOTAL
          COMMAND.
0016     MATCH COMMAND SEQUENCE ERROR:
          CORRESPONDING FILEIN, MATCH, OR
          ENDM COMMAND NOT FOUND.
0017     ALL REFERENCES TO #17 SHOULD NOW BE #41
```

```
0018      REFERENCE TO $LCTR OR $LCTF
           VARIABLE BEFORE CORRESPONDING
           REPORT OR FILEOUT COMMAND NOT
           ALLOWED
0019      FILEIN RECORD LENGTH ERROR:
           THE ITEMS DEFINITION EXCEEDS
           THE LRECL; THE DEFAULT LRECL
           IS EIGHTY (80) CHARACTERS.
0020      ITEM OR FORMAT COMMAND SEQUENCE ERROR:
           NO CORRESPONDING FILEIN, FILEOUT,
           REPORT OR ITEM PROCEDURE FOUND.
0021      CHARACTER LITERAL EXCEEDS ALLOWABLE
           LENGTH
              A.   1 CARD IMAGE (UP TO 78
                   CHARACTERS) FOR LITERAL IN
                   ITEM COMMAND
              B.   40 CHARACTERS FOR LABEL COMMANDS
              C.   52 CHARACTERS FOR COMPUTE
                   COMMAND
0022      INVALID VARIABLE NAME:
              A.   NAME GREATER THAN 8 OR LESS
                   THAN 2 CHARACTERS IN LENGTH
              B.   'TAB' USED AS VARIABLE NAME
              C.   FIRST CHARACTER NOT ALPHABETIC;
                   THE FIRST CHARACTER OF THE
                   NAME MUST BE ALPHABETIC OR THE
                   DOLLAR ($) CHARACTER.
0023      INVALID TABLE REFERENCE:
           TABLE ITEMS ARE NOT ALLOWED IN THE
           SORT, TOTAL OR MATCH COMMANDS.
0024      TOO MANY OPERANDS FOR COMMAND
0025      SORT OPTION NOT ALLOWED WITH SEGMENTED
           FILE
0026      ARITHMETIC EXPRESSION IS TOO LARGE:
           RERUN CENSPAC WITH SIMPLIFIED
           EXPRESSIONS.
0027      CENSPAC COMPUTE PROGRAM ERROR:
           CONTACT DATA USER SERVICES
           DIVISION, U.S. BUREAU OF CENSUS.
0028      PREVIOUS IF COMMAND STILL ACTIVE WHEN
           NEW IF COMMAND WAS ENCOUNTERED,
           OR INCLUDE, EXCLUDE, STOP OR ';'
           WAS ENCOUNTERED WITHOUT IF COMMAND
           BEING ACTIVE.
0029      EQUAL '=' SIGN DOES NOT SEPARATE RESULT
           ITEM FROM ARITHMETIC EXPRESSION.
0030      INVALID SYNTAX IN ARITHMETIC EXPRESSION:
           SYMBOL IS INVALID, UNDEFINED OR NOT
           IN THE REQUIRED SEQUENCE.
0031      RIGHT PARENTHESIS OUT OF SEQUENCE.
0032      INVALID SYNTAX IN COMPUTE OR IF COMMAND,
           SCAN TERMINATED.
0033      UNBALANCED PARENTHESIS:
           PARENTHESES MUST BE PAIRED.
0034      TABLE DIMENSION ERROR:
           MORE THAN SIX DIMENSIONS REFERENCED.
0035      TABLE SUBSCRIPT ERROR:
           SYMBOL NOT AN ASTERISK AND NOT AN
           UNSIGNED NUMERIC LITERAL.
0036      MISSING SUBSCRIPT:
           TABLE REFERENCE IN THIS CONTEXT
           REQUIRES A SUBSCRIPT -  NONE FOUND.
0037      ARRAY OPERATION ERROR:
```

```
               A.  'C' OR '*' NOT FOUND IN
                   LABEL SUBSCRIPT.
               B.  ARRAY OPERATOR ('C' OR '*')
                   SPECIFIED IN THE FILEIN COMMAND.
0038       ARRAY/REPETITION FACTOR MISMATCH:
            WARNING, THE REPETITION FACTOR
            IS INSUFFICIENT TO DEFINE ALL
            CELLS IN THE RESPECTIVE TABLE
            DEFINITION FOR A FILEIN ITEM.
            THE NUMBER OF CELLS IS USED AS
            THE DEFAULT REPETITION FACTOR.
0039       ARRAY SUBSCRIPT RANGE ERROR:
               A.  THE ARRAY SUBSCRIPT LITERAL IS
                   OUTSIDE THE RANGE OF THE
                   DIMENSION.
               B.  MULTIPLE DIMENSIONS INDICATED
                   FOR STRATIFIER SUBSCRIPT - ONE
                   REQUIRED.
0040       ARRAY REFERENCES DO NOT CONTAIN AN
            EQUAL NUMBER OF DIMENSIONS OR
            SUBSCRIPT AND DICTIONARY DIMENSIONS
            ARE INCOMPATIBLE.
0041       INVALID OR UNDEFINED VARIABLE NAME
0042       INPUT DEFINITION ERROR:
            NO FORMAT SPECIFICATION
            ASSOCIATED WITH THE INPUT VARIABLE.
0043       DIMENSION ERROR:
            DATA ITEM CONTAINS SUBSCRIPT
            REFERENCE TO A NON-SUBSCRIPTED
            VARIABLE.
0044       FORMAT SPECIFICATION ERROR:
            WIDTH SPECIFICATION LESS THAN ONE.
0045       INVALID RESERVED WORD USAGE:
            RESERVED WORD $PAGE OR $DATE
            INVALID IN ITEM COMMAND.
0046       WARNING, ANOTHER FORMAT STATEMENT IS
            CURRENTLY ACTIVE AT THE TIME A LITERAL
            OR ANOTHER FORMAT IS ENCOUNTERED.
0047       FORMAT SPECIFICATION ERROR:
               A.  NO CORRESPONDING DATA ITEMS
                   SPECIFIED; OR
               B.  REPETITION FACTOR GREATER
                   THAN 999 OR REPETITION
                   FACTOR = 0
0048       WIDTH INCONSISTENT WITH USAGE:
               A.  THE NUMBER OF DECIMALS EXCEED
                   THE WIDTH OF THE FIELD OR;
               B.  THE WIDTH SPECIFICATION EXCEEDS
                   18 AND NUMERICAL USAGE IS
                   SPECIFIED OR IMPLIED.
0049       WIDTH AND 'T' SPECIFICATION ERROR:
            THE WIDTH SPECIFICATION IS LESS
            THAN TWO AND 'T' SPECIFICATION
            IS $, ⊢, AND/OR A DECIMAL MASK
            IS REQUIRED.
0050       FORMAT SYNTAX IN THE ITEM COMMAND.
0051       COMMAND OPERAND NOT YET OPERATIONAL
0052       COMMAND CONTAINS DUPLICATE SPECIFICATION
            OR FILEOUT/REPORT COMMANDS.
0053       INVALID INPUT FILE SPECIFICATION:
               A.  LITERAL INVALID AS FILEIN ITEM.
               B.  BLKSIZE TO LARGE OR BLKSIZE
                   NOT A MULTIPLE OF THE LRECL.
               C.  UNIDENTIFIED PARAMETER FOUND
0054         $SUP VARIABLE REFERENCED FOR TABLE
              THAT HAS 'SX' SUPPRESSION.
0055         ASTERISK REFERENCE NOT VALID IN THE 'IF'
              ARRAY, AND FILEIN COMMANDS.
0056         OUTPUT OF ALPHANUMERIC DATA UNDER A
              NUMERIC FORMAT SPECIFICATION IS
              INVALID.
0057         COMPLETE TABLE WILL NOT PRINT DUE TO
              FORMAT CONTROL; THE NUMBER OF CELLS
              FOR OUTPUT IS LIMITED TO THE REPETITION
              FACTOR IN THE FORMAT.
0058         COLUMNAR AND LINE ARRANGEMENT OF TABLE
              LABELS FOUND IN THE SAME ITEM COMMAND.
              USE AN ADDITIONAL ITEM COMMAND TO
              CORRECT THIS PROBLEM.
0059         ARRAY ERROR;  ONLY NEW VARIABLES
              PREFIXED BY 'TAB' AND DEFINED WITH
              VALID ARRAY PARAMETERS ARE PERMITTED.
0060         INVALID SYNTAX IN THE IF COMMAND
0061         CENSPAC TOTAL CAPACITY EXCEEDED:
              THE TOTAL AREA IS LIMITED TO
              4095 CELLS.
0062         ZERO NUMERIC DATA ITEMS TOTALLED
              BY TOTAL COMMAND
0063         MULTIPLE SORT REFERENCES:
              MORE THAN ONE SORT COMMAND FOUND
              AFTER THE FILEIN SECTION, ONLY
              ONE IS ALLOWED.
0064         DUPLICATE MATCH COMMAND REFERENCES:
              MULTIPLE MATCH REFERENCES TO A FILE.
0065         THE MAXIMUM NUMBER OF TOTAL LEVELS
              (5) EXCEEDED.
0066         INVALID RELATIONAL OPERATOR IN THE
              'IF' COMMAND.
0067         RECORD LENGTH DEFINED BY ITEMS, OR
              HEAD EXCEEDS MAXIMUM PAGE WIDTH; OR
              FOR FILEOUT, THE RECORD LENGTH DEFINED
              BY THE FIRST ITEM COMMAND IS EXCEEDED.
0068         HEAD COMMAND IS NOT IMMEDIATELY
              SUBORDINATE TO A REPORT COMMAND.
0069         $ INVALID AS FILEIN OR FILEOUT.
0070         THRU USAGE ERROR
              THRU NOT PRECEEDED OR NOT FOLLOWED
              BY DATA-NAME
              BEGIN DATA NAME NOT A DATA DICTIONARY
              ITEM
0071         END OF RECORD FORMAT OR END OF
              DICTIONARY FOUND BEFORE END
              OF THRU RANGE
0072         MACRO REQUEST MADE, HOWEVER SYSTEXT
              NOT AVAILABLE OR REQUEST NOT FOUND
0073         BINARY DATA ITEMS MUST BE
              2, 4, 6, OR 8 CHARACTERS
              IN WIDTH
0074         LIBRARY CALL FROM WITHIN
              LIBRARY CALL FOUND, NOT
              ALLOWED
0075         NO OUTPUT COMMANDS IN RUN
              RUN TERMINATED. RE-RUN WITH APPROPRIATE
              FILEOUT AND/OR REPORT COMMANDS
0076         TAB POSITION SPECIFIED IS LESS
              THAN CURRENT RECORD POSITION
```

```
0077      MORE THAN 40 LINES IN LIBRARY ELEMENT,
           MAXIMUM ALLOWED IS 40 LINES.
0078      $DATE OR $PAGE USED MORE
           THAN 1 TIME IN A REPORT
0080      A RUN TIME LABEL REQUEST WAS MADE UPON A
           VARIABLE WHICH ALREADY HAS ITS OWN
           LABELS.
0081      A RUN TIME LABEL REQUEST WAS MADE, BUT
           THE SUBJECT AND OBJECT VARIABLES WERE
           OF A DIFFERENT CHARACTER. ONE WAS A
           TABLE ITEM--- THE OTHER WAS A DATA
           ITEM. THEY MUST BOTH BE OF THE SAME
           TYPE.
0083      A RUN TIME LABEL REQUEST WAS MADE, BUT
           THE LABEL TYPE IS INVALID OR MISSING.
           THE VALID LABEL TYPES ARE:
                  H, U, S, C, G FOR TABLE ITEMS;
                  C AND D FOR DATA ITEMS.
0084      A RUN TIME LABEL REQUEST WAS MADE BUT
           AN ILLEGAL SPECIFICATION WAS MADE.
           CHECK THE FOLLOWING.
              1.  THE SUBFIELD (IN PARENTHESIS)
                  CAN ONLY BE SPECIFIED FOR
                  THE LABEL TYPES S OR C
              2.  THE SUBFIELD (IN PARENTHESIS)
                  IS NOT A NUMERIC ITEM.
              3.  THE SUBFIELD (IN PARENTHESIS)
                  IS GREATER THAN THE ARRAY
                  MAXIMUM FOR THE TABLE IN
                  QUESTION.
0085      A RUN TIME LABEL REQUEST WAS MADE BUT
           THE DATA CATEGORY VALUE SPECIFIED
           IS INCOMPATIBLE WITH:
              1.  THE NUMERIC DATA ITEM
                  SPECIFIED OR
              2.  THE SIZE OF THE CATEGORY
                  VALUE IS TOO LARGE
0086      A RUN TIME LABEL REQUEST WAS MADE BUT
           THE LABEL TEXT IS MISSING.
0087      A RUN TIME LABEL REQUEST WAS MADE WITH
           CATEGORY LABELS WHICH ARE NOT IN THE
           PROPER SEQUENCE.  RESEQUENCE THE
           CATEGORY LABELS AND RESUBMIT.
0088      A RUN TIME LABEL REQUEST WAS ATTEMPTED
           ON THE SAME LABEL TYPE TWICE.
0089      A RUN TIME LABEL REQUEST WAS MADE---
           ONLY ONE GROUP LABEL REFERENCE CAN BE
           MADE WITHOUT THE INTRODUCTION OF AN
           INTERVENING CATEGORY LABEL.
0090      A RUNTIME LABEL REQUEST WAS MADE USING
           A CATEGORY LABEL DEFINITION WHICH IS
           OUT OF THE PREDEFINED DOMAIN OF THE
           TABLE IN USE.
0095      RECORD-POINTER IN WORK FILE
           INDEX > FILE-SIZE OR LESS
           THAN 1 (SYSTEM ERROR
           RERUN JOB OR CONTACT
           CENSUS BUREAU)
0096      INVALID KEY ON CUF-WRITE
           POSSIBLE I/O ERROR. RE-RUN JOB.
           IF ERROR PERSISTS CONTACT CENSPAC
           STAFF FOR HELP.
0097      INVALID KEY ON CUF-READ


           POSSIBLE I/O ERROR. RE-RUN JOB.
           IF ERROR PERSISTS CONTACT CENSPAC
           STAFF FOR HELP.
0099      RECORD-PTR (CUF KEY) OUT OF RANGE
           BETWEEN LAST RECORDS WRITTEN ON
           WRITE. CENSPAC SYSTEM ERROR. CONTACT
           CENSPAC STAFF FOR HELP.
0100      WARNING----MISSING RIGHT PARENTHESIS--
           THE RIGHT PARENTHESIS IS ASSUMED
           PRESENT
0101      FILEIN WARNING:
           DATA DICTIONARY SPECIFIED IN
           FILEIN COMMAND NOT AVAILABLE.
0103      UNIDENTIFIED NUMERIC FOUND IN
           COLUMNS 73-80 ON FILEIN COMMAND
           COLUMNS 73-80 WILL NOT BE
           SCANNED IN REMAINING COMMANDS
0104      SYSGEN OPTION SPECIFIED
           CENSPAC WORK FILE AND MODEL
           FILE TO BE INITIALIZED ONLY
           IN THIS RUN
0105      EDIT COMMAND-
           EDIT RUN ONLY
```

Appendix 2

USER-SUPPLIED SOURCE CODE

By including COBOL source statements in a CENSPAC run, a user can extend the capabilities of CENSPAC without modifying the CENSPAC system. CALL statements can be included in the COBOL statements to call separately compiled subroutines that perform the data processing functions needed for an application. On operating systems where CALL can be made from COBOL to other programming languages, the CALLS might link CENSPAC to scientific subroutines or special output functions for graphic display. In order to do this, a familiarity with COBOL and certain CENSPAC conventions is required.

CENSPAC does not edit the COBOL statements provided in SOURCE mode. Errors detected by the compiler will have to be corrected using the COBOL source listings and COBOL error messages.

The SOURCE facility may be used one or more times in a CENSPAC program, and there is no limit to the number of statements allowed with each use. The COBOL statements will be incorporated into the CENSPAC program immediately after the command that precedes the SOURCE command. Paragraph names and COBOL procedure statements are allowed. COBOL source statements that begin in column one of a card image are treated as paragraph names. Statements beginning after column one are treated as procedure division statements. COBOL 'SECTIONS' should not be used because they will interfere with the control structure of the program. Only columns 1-54 of the SOURCE commands are incorporated into the CENSPAC program. Columns 55-80 are ignored.

Data Name Rules

In order to properly use data names in the SOURCE COBOL statements, the user must follow the internal naming conventions of CENSPAC. There are several work areas established in a CENSPAC program, some are used by particular CENSPAC commands, and a common work area is shared by all commands. Further, a CENSPAC program is generated in three parts: (1) input, (2) processing (data manipulation), and (3) output.

As data records are read, the data items are available first in the input work area. The data are then moved to a common work area where they are changed to computational form, if necessary. All data manipulation takes place in the common work area, and the processing commands (COMPUTE, IF, SORT, TOTAL) use the common work area as the source for data.

If a TOTAL command is present, data are aggregated to a TOTAL work area from the common area. The TOTAL work area has multiple levels, one for each level of TOTAL to be generated. As TOTAL breaks occur, the data are moved back to the common work area for use by the next command.

If a SORT command is present, the data flow from the common work area to the SORT work area, are released to the system sort, and then returned from the SORT and made available in the common work area again.

In most cases SOURCE statements will reference data in the common work area. The data in all work areas and the CENSPAC reserved words are available to the user. However, the easiest manner of accessing data items is from the common work area. Also all data items referenced in SOURCE mode must also be referenced in a CENSPAC command outside of SOURCE mode. The naming conventions used in generating the data names in the work areas are as follows:

1. All data names are a maximum of 11 characters in length, including the up-to-8 character data dictionary name and a 2-3 character prefix.

2. Data names in the input work area are prefixed with the letters 'A-' or 'B-'. 'A-' is used for those items from the first, or only file if no MATCH is to be done, 'B-' is used to prefix data names from the second file in a MATCH run. Input work area items should be referenced only when original input format is needed before computational conversion, such as might occur with 1970 suppressed data.

3. The data items in the common work area are prefixed with 'A00' or 'B00' for first and second file data items, respectively. Data items that are locally defined such as from a COMPUTE command and are not associated with an input record are prefixed with 'A00'.

4. System variables are available; the leading dollar sign ($) is replaced with Q. These items are defined independently of the common work area.

5. The '#' variables are also available in a separate area. The '#' character is removed and the data name is prefixed with 'A00'. For example, the data name '#ABC' would be identified as 'A00ABC' in the work area.

Other Rules

Numeric literals may be used as needed. The maximum length, including leading sign and decimal point, is 9 characters.

Character literals may be used as needed. The literal must be enclosed in single quotation marks. The maximum length, including the 2 quotation marks, is 9 characters.

Data use must be consistent with COBOL rules. Only items defined as numeric and numeric literals may be used in operations involving numeric functions (adding, multiplying, etc.) Alphanumeric items and character literals cannot be used in a numeric context. Special characters -, +, *, /, (,), <,> , and =, may be used to construct COBOL logical and arithmetic statements Data items needed only in the SOURCE statements must be referenced in the CENSPAC run so that they are available in the input and common work areas. This can be done in an ITEM command or in any CENSPAC command in the program.

Several paragraph and section names are generated in a CENSPAC executor program, but only a few are important to the SOURCE user. The important section names are 'PROCSFILA' and 'OUTFIL'. 'PROCSFILA' is always generated and is the main processing section for the program. 'OUTFIL' is generated only when using the SORT command and it controls output processing of the sorted file. After processing a record, control passes to the name of the current section ('PROCSFILA' or 'OUTFIL') in order to get the next record. The ending paragraph for these sections are named 'END-FILA' (for 'PROCSFILA') and 'END-FILM' (for 'OUTFIL'). Control should be allowed to 'fall through' to the next CENSPAC generated statement or must pass to the appropriate ending paragraph name to terminate processing of the current record and to continue with the next record if end-of-file has not been reached.

In a MATCH, additional section names are generated and the function of 'PROCSFILA' changes. It is the section that handles CENSPAC File A logic. 'PROCSFLIB' handles the File B logic. 'PROCSFILN' is generated for the matched logic processing. Each of these sections ends with a corresponding 'END-FILn' paragraph where 'n' is A, B, or N depending on which section is involved.

Additional paragraph names are generated to control each IF and COMPUTE command. These paragraph names can appear in any of the sections mentioned previously, and they are sequentially numbered. The names for the IF pagagraphs are in the form 'IFCMDnnn;, where 'nnn' is a 3 digit sequence number beginning with 001 for the first IF command.

For the COMPUTE commands the paragraph names are 'Cnnn-X', where 'nnn' is a 3 digit paragraph sequence number for the COMPUTE commands, starting with 001. For both the IF and COMPUTE commands, the paragraph name is generated at the end of the logic needed to perform the command.

Other paragraph and section names are generated for SORT, TOTAL, REPORT, FILEOUT, and FILEIN commands. However, CENSPAC SOURCE statements usually would not interface directly with these functions.

Before using the SOURCE features with paragraph names, a user should review a sample CENSPAC executor program.

Example

```
FILEIN STF1
IF SUMRYLVL NOT = '14' EXCLUDE
COMPUTE RADIUS = 0
COMPUTE XX = 38.0130
COMPUTE YY = 77.0000
SOURCE
     CALL 'GEOSUB' USING A00LATITUDE
     A00LONGITUD A00RADIUS A00XX
     A00YY
SOURCE
IF RADIUS> 5 EXCLUDE
FILEOUT BLFACTOR 10
ITEM LATITUDE LONGITUD XX YY RADIUS
     STATE COUNTY TRACT ED TAB001 TAB005
```

In this example, a user-written subroutine named 'GEOSUB' is called to calculate the distance in miles from a given latitude and longitude. An output file containing the geographic coordinates, selected geocodes, and Tables 1 and 5 is created for those records that are within a 5 mile radius.

Appendix 3

CENSPAC and DOCUMENTOR Keywords

AND
ARRAY
ASCII
BLFACTOR
BLKSIZE
CARD
CENIO
COMPUTE
DATANAME
DEVICE
DOC
DOCUMENT
DICTNAME
DISK
EBCDIC
EDIT
END
ENDM
EQUAL
ERRTEXT
EXCLUDE
EXPRESS
FIELD
FILEIN
FILENAME
FILEOUT
FOOT*
FOOTNOTE*
FORMAT*
FROM
FUNCTION
GEOLEVEL*
GREATER
HEAD
HELP
IBMSL
IBMUL
IF
INCLUDE
IOMODULE
ITEM
LABEL
LESS
LINK**
LITERAL
MATCH
NOT
NOTE*
OPTION
OR
PAGESIZE
PWIDTH
RECSIZE
RENAME
REPORT
SDF
SEQ
SEQUENCE
SORT
SOURCE
STOP
SUM
SYNTAX
TAPE
TAPU
TEXT*
THAN
THEN
THRU
TRACE
TOTAL

All file names in Appendix 4
All CENSPAC predefined system variables ($) in Chapter II

*Documentor program only
**not functional, word reserved

Appendix 4

CENSPAC RESERVED FILE NAMES

STF1	1980 Summary Tape File One
STF2	1980 Summary Tape File Two
STF3	1980 Summary Tape File Three
STF4	1980 Summary Tape File Four
STF5	1980 Summary Tape File Five
CNT1	1970 First Count File
CNT2	1970 Second Count File
CNT3	1970 Third Count File
CNT4P	1970 Fourth Count Population File
CNT4H	1970 Fourth Count Housing File
CNT5C	1970 Fifth Count File (BG/ED)
CNT6P	1970 Sixth Count Population File
CNT6H	1970 Sixth Count Housing File

Appendix 5

CENSPAC DATA DICTIONARIES*

Name	File Description	Space Requirements**
STF1	1980 Summary Tape File One	103
STF2	1980 Summary Tape File Two	187
CNT1	1970 First Count	49
CNT2	1970 Second Count	103
CNT4P	1970 Fourth Count Population	123
CNT5C	1970 Fifth Count	52
PL94171	1980 Special Population File	13
CPSMAR80	1980 March CPS	147
MARF	Master Area Reference File	31
LINK1	1972 Link of Census Establishment and IRS File 1	6
LINK2	1972 Link of Census Establishment and IRS File 2	6
LINK3	1972 Link of Census Establishment and IRS File 3	6
LINK4	1972 Link of Census Establishment and IRS File 4	6
GBFDIME	1980 Geographic Base FILE/DIME	10
RC77AT1	Census of Retail Trade, 1977: Geographic Area Files	12
RC77AT2		13
RC77AT3	Files T1 Through T4 are Kind-of-Business	15
RC77AT4		12
RC77AT5		13
RC77AT6	Files T5 Through T8 are Individual Geographic Areas	20
RC77AT7		10
RC77AT8		10
CBP78ST	County Business Patterns, 1978: Table 1B	18
CBP78CTY	County Business Patterns, 1978: Table 2	13
CBP77SMS	County Business Patterns, 1977: SMSA Summary	7
CBPUS77	County Business Patterns, 1977: U.S. Summary	13
AHSTR76	Annual Housing Survey, National Travel-to-Work File	11
GRSACT78	General Revenue Sharing, 1978; Population Estimates	27

*These data dictionaries were made available with the January 1982 CENSPAC system tape. Additional data dictionaries are being made available. A current list is provided with the CENSPAC system tape and is also available through Systems and Programming Branch, Data User Services Division, Washington, D.C. 20233.

**Space requirements indicate the number of 1024 character records required to contain the data dictionary information for a particular data dictionary.

☆U.S. GOVERNMENT PRINTING OFFICE: 1982-360-997/2059

CENSUS SOFTWARE PACKAGE (CENSPAC)

CENSPAC USER REGISTRATION FORM

Notice of CENSPAC corrections and other information will be provided to persons who return this form. Please send the form to Chief, Data User Services Division, Bureau of the Census, Washington, D.C. 20233.

1. CENSPAC Liasion or contact name______________________________

   Organization Name______________________________

   Address______________________________

   ____

   Phone number____________________

2a. Computer System used:

   Manufacturer Model______________________________

   Operating system______________________________

2b. If CENSPAC was converted to operate on a system other than IBM OS/VS or UNIVAC EXEC-8, will you make the converted version available to others?____________________
   If yes, who is the contact person for information regarding this conversion.______________________________

3. Were CENSPAC and DOCUMENTOR installed?
   CENSPAC______ DOCUMENTOR______

4. Please describe any problems encountered in the installation of CENSPAC?

5. Was the CENSPAC staff at the Census Bureau contacted for installation assistance?______________

CENSUS SOFTWARE PACKAGE (CENSPAC)

6. Please indicate source of your CENSPAC program tape

Census Bureau_____ State Data Center_____ Other_____

7. Describe problems encountered in using CENSPAC.

8. Do you have any comments on the content, format, and structure of the preliminary CENSPAC Users Manual?

9. Recommendations for changes, improvements, or other comments.

August 1980

CENSUS SOFTWARE PACKAGE (CENSPAC)

CENSPAC USER REGISTRATION FORM

Notice of CENSPAC corrections and other information will be provided to persons who return this form. Please send the form to Chief, Data User Services Division, Bureau of the Census, Washington, D.C. 20233.

1. CENSPAC Liaison or contact name ________________________________

Organization Name ________________________________

Address ________________________________

____

Phone number ____________________

2a. Computer System used:

Manufacturer Model ________________________________

Operating system ________________________________

2b. If CENSPAC was converted to operate on a system other than IBM OS/VS or UNIVAC EXEC-8, will you make the converted version available to others? ________________ If yes, who is the contact person for information regarding this conversion.________________________________

3. Were CENSPAC and DOCUMENTOR installed?
CENSPAC ______ DOCUMENTOR______

4. Please describe any problems encountered in the installation of CENSPAC?

5. Was the CENSPAC staff at the Census Bureau contacted for installation assistance? ________________

6. Please indicate source of your CENSPAC program tape

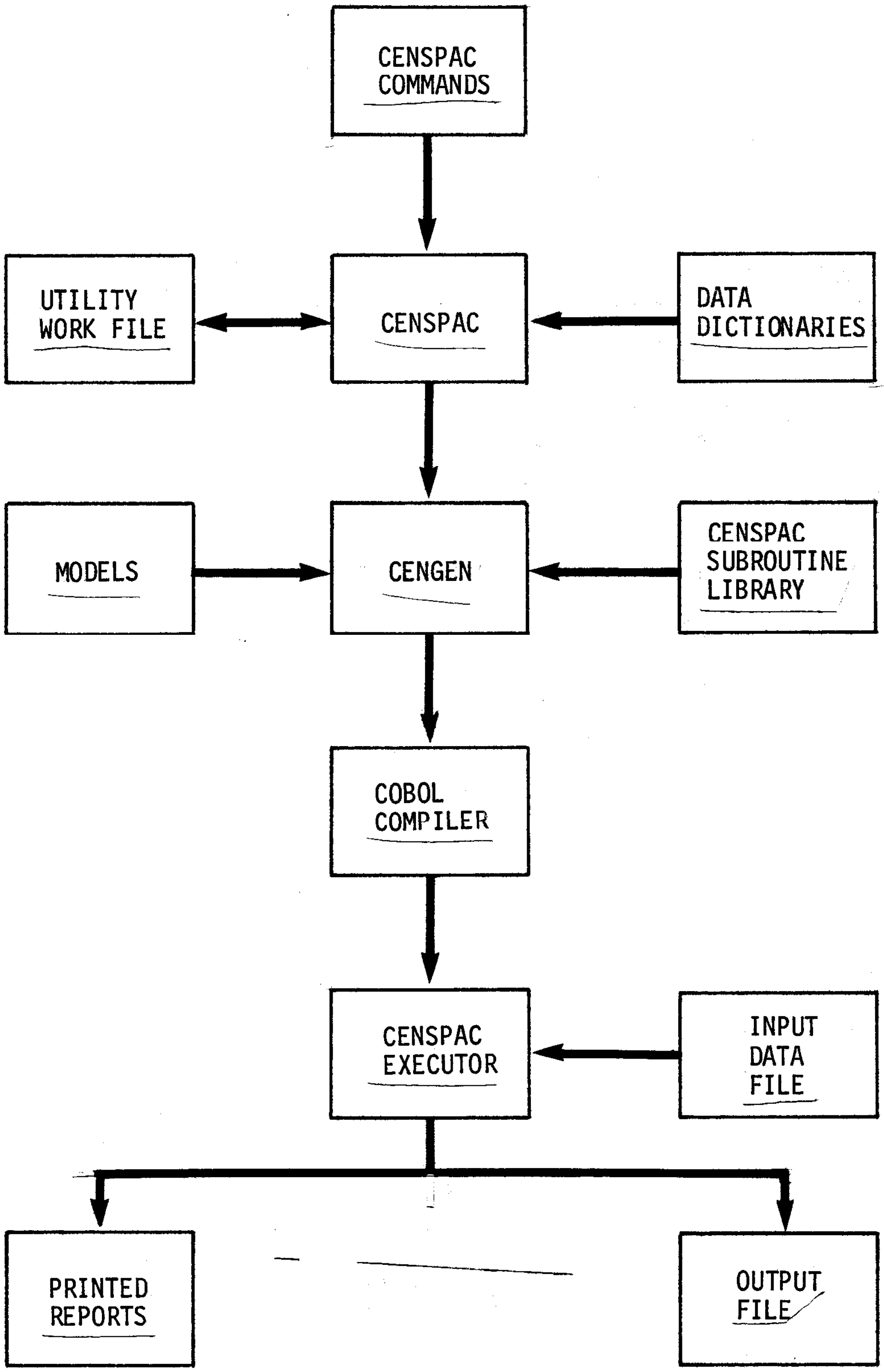
Census Bureau__________ State Data Center __________ Other ______________

CENSPAC IBM VERSION

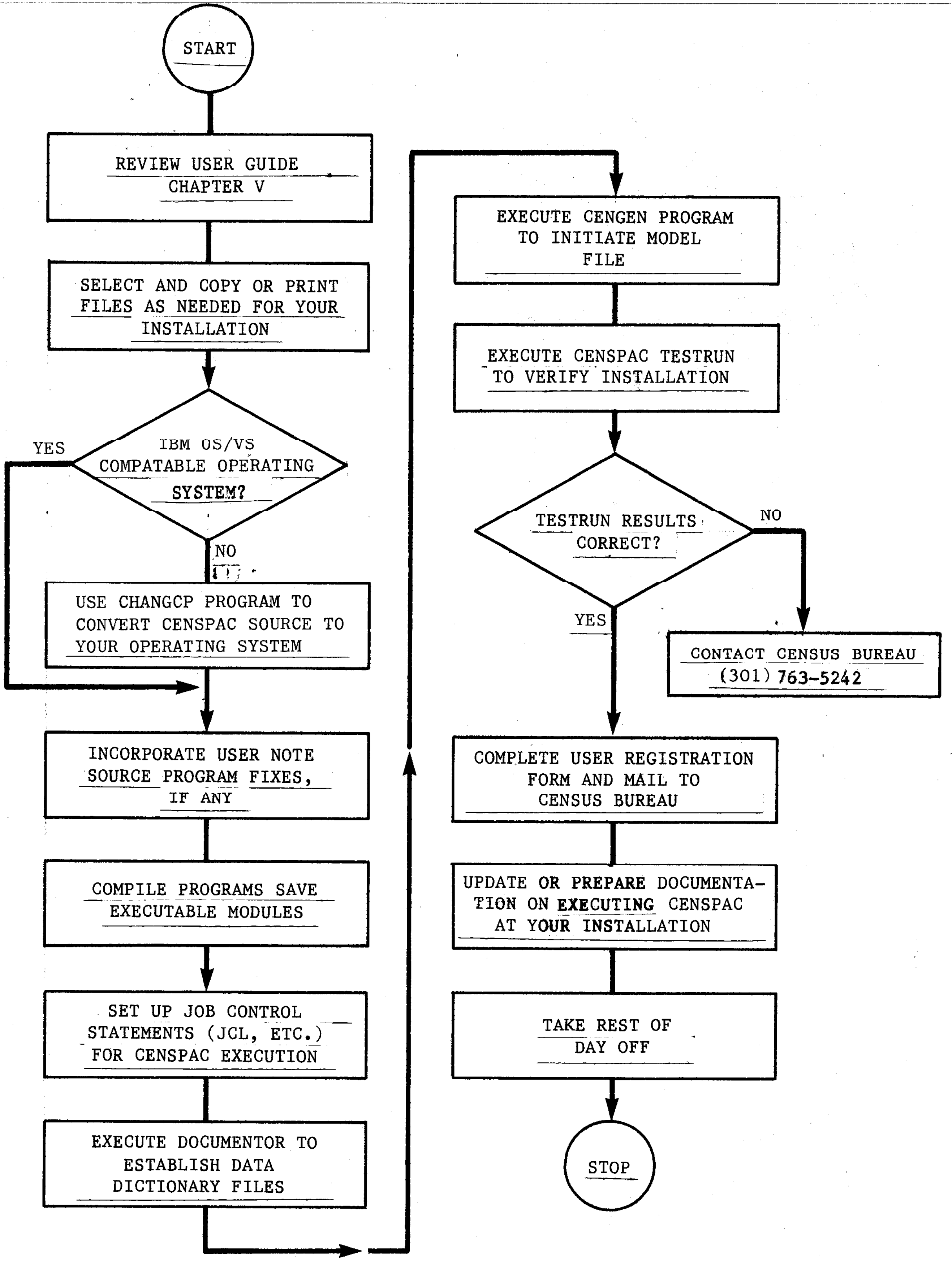
CENSPAC has been developed and tested using an IBM VS 1974 ANSI COBOL compiler. It is being released prior to the full range of tests on the UNIVAC system, in order that users with IBM systems can begin to work with it now. Although the UNIVAC version is included on the sales tape, changes on this version may be required and orders for customers with UNIVAC systems will be filled as soon as this testing is completed.

DATA USER SERVICES DIVISION
May 5, 1980

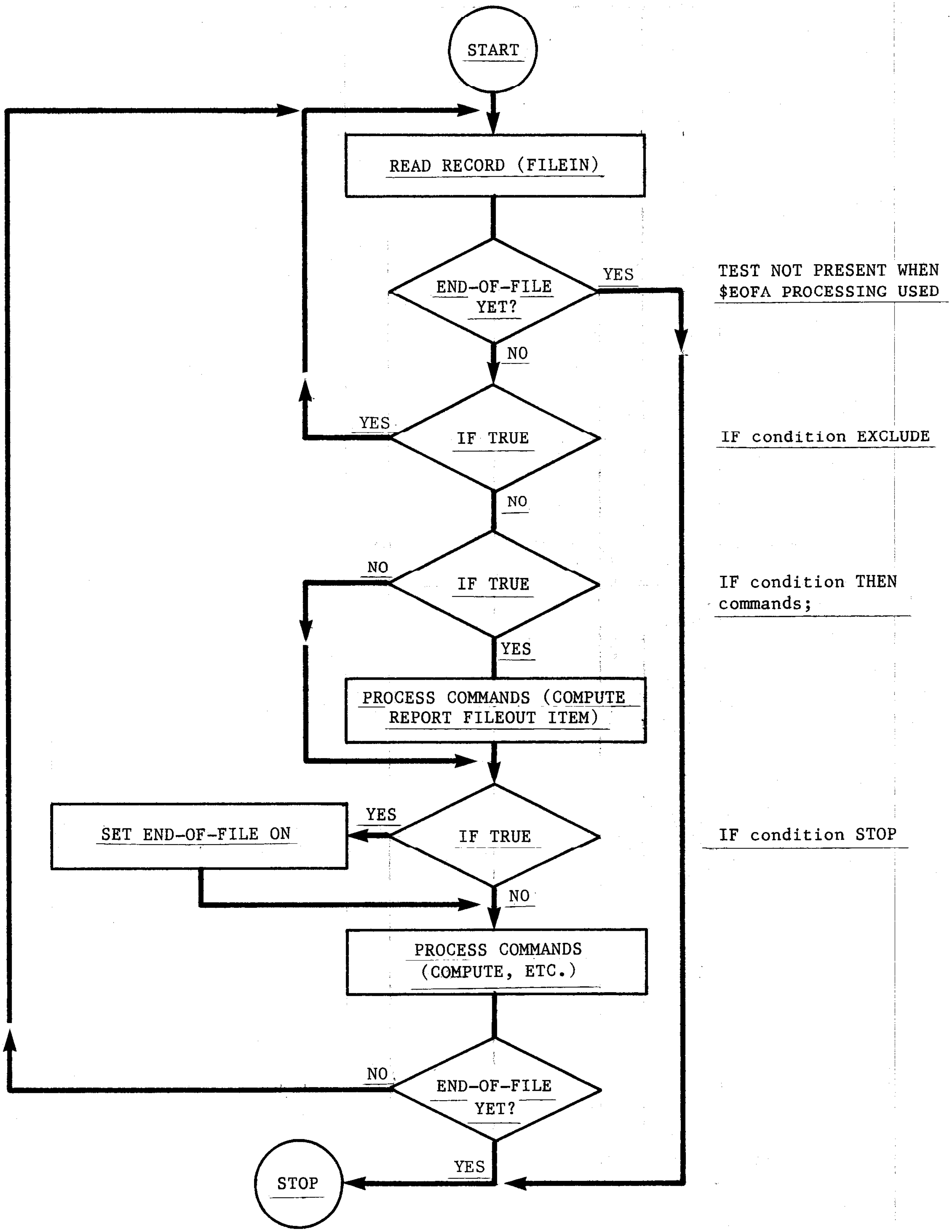
CENSUS SOFTWARE PACKAGE (CENSPAC) SYSTEM DIAGRAM



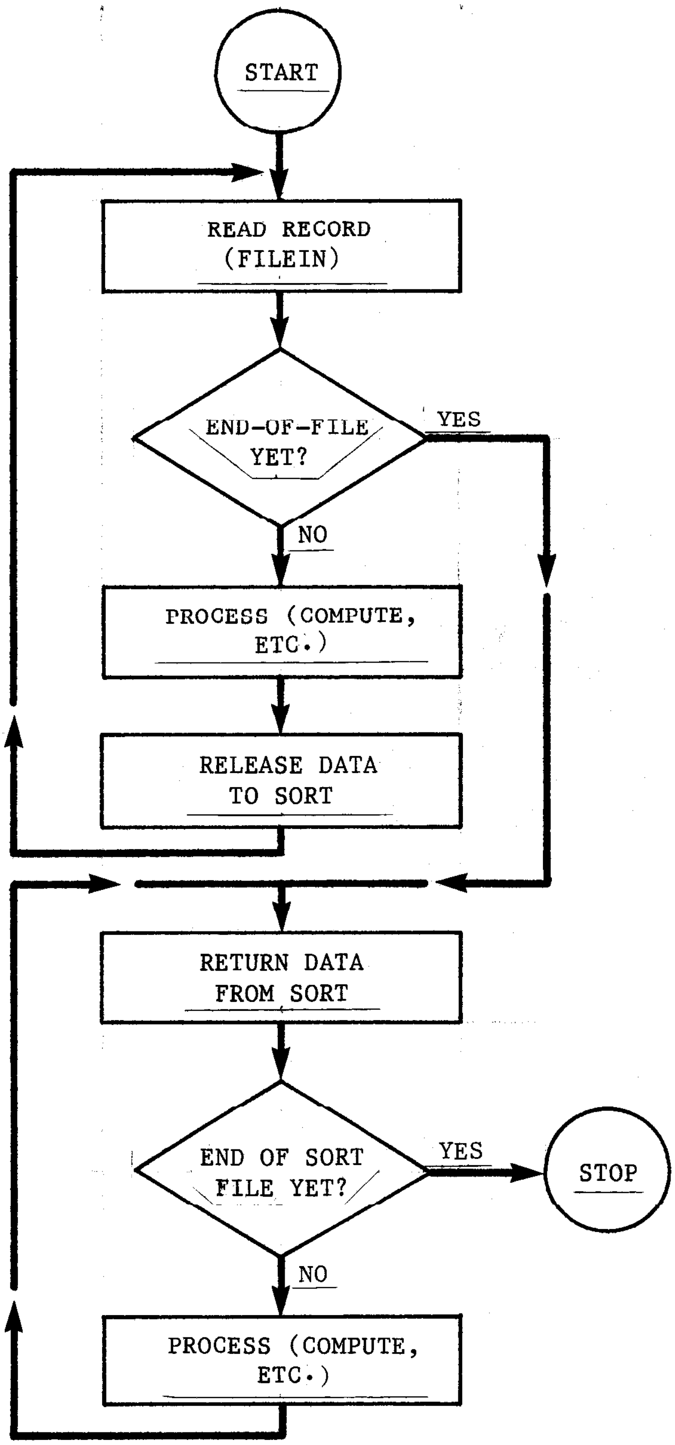
CENSPAC INSTALLATION PROCEDURE


START
REVIEW USER GUIDE CHAPTER V
SELECT AND COPY OR PRINT FILES AS NEEDED FOR YOUR INSTALLATION
IBM OS/VS COMPATABLE OPERATING SYSTEM?
YES
NO
USE CHANGCP PROGRAM TO CONVERT CENSPAC SOURCE TO YOUR OPERATING SYSTEM
INCORPORATE USER NOTE SOURCE PROGRAM FIXES, IF ANY
COMPILE PROGRAMS SAVE EXECUTABLE MODULES
SET UP JOB CONTROL STATEMENTS (JCL, ETC.) FOR CENSPAC EXECUTION
EXECUTE DOCUMENTOR TO ESTABLISH DATA DICTIONARY FILES
EXECUTE CENGEN PROGRAM TO INITIATE MODEL FILE
EXECUTE CENSPAC TESTRUN TO VERIFY INSTALLATION
TESTRUN RESULTS CORRECT?
NO
CONTACT CENSUS BUREAU (301) 763-5242
YES
COMPLETE USER REGISTRATION FORM AND MAIL TO CENSUS BUREAU
UPDATE OR PREPARE DOCUMENTATION ON EXECUTING CENSPAC AT YOUR INSTALLATION
TAKE REST OF DAY OFF
STOP

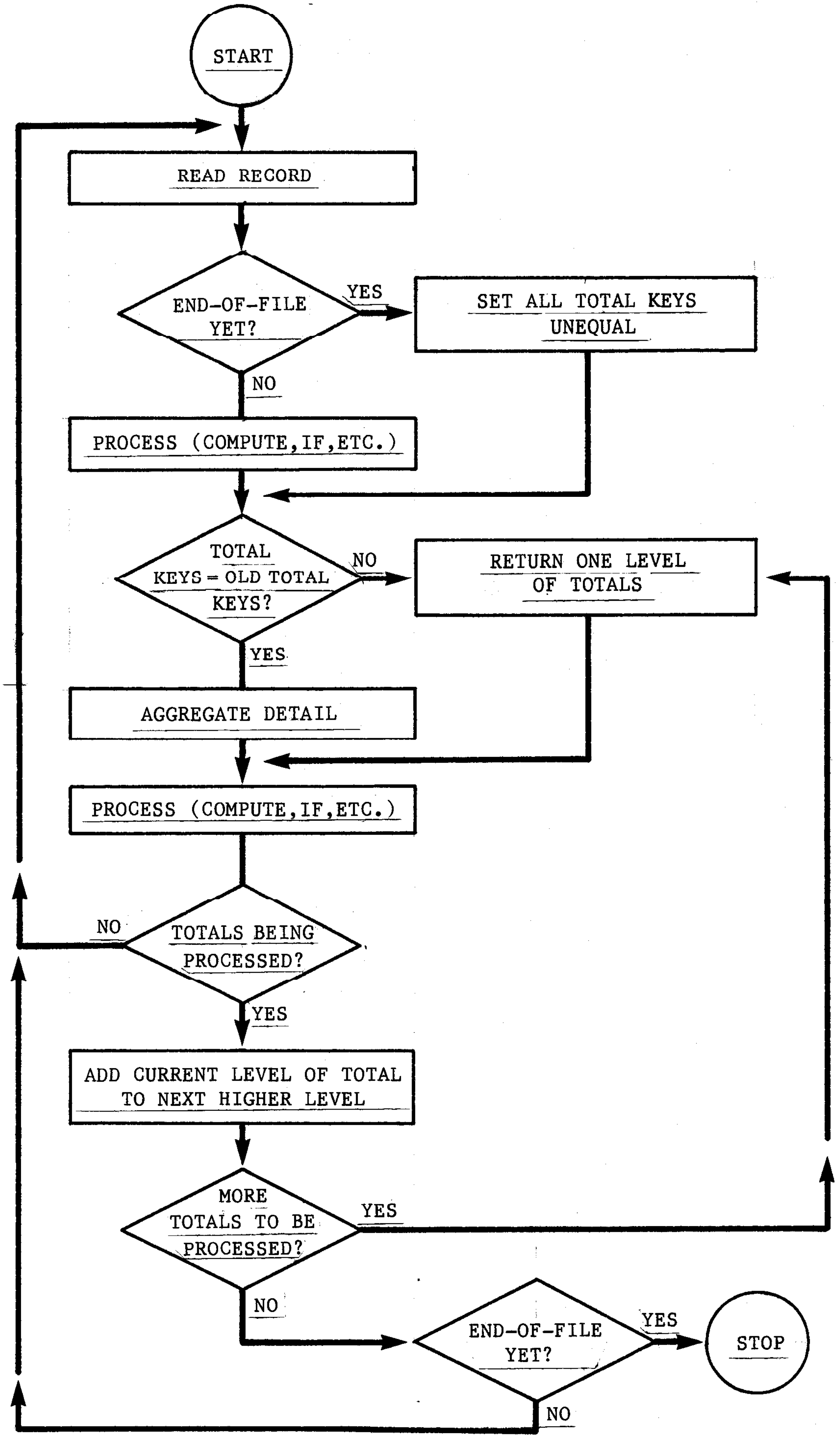
CENSPAC RECORD PROCESSING LOGIC



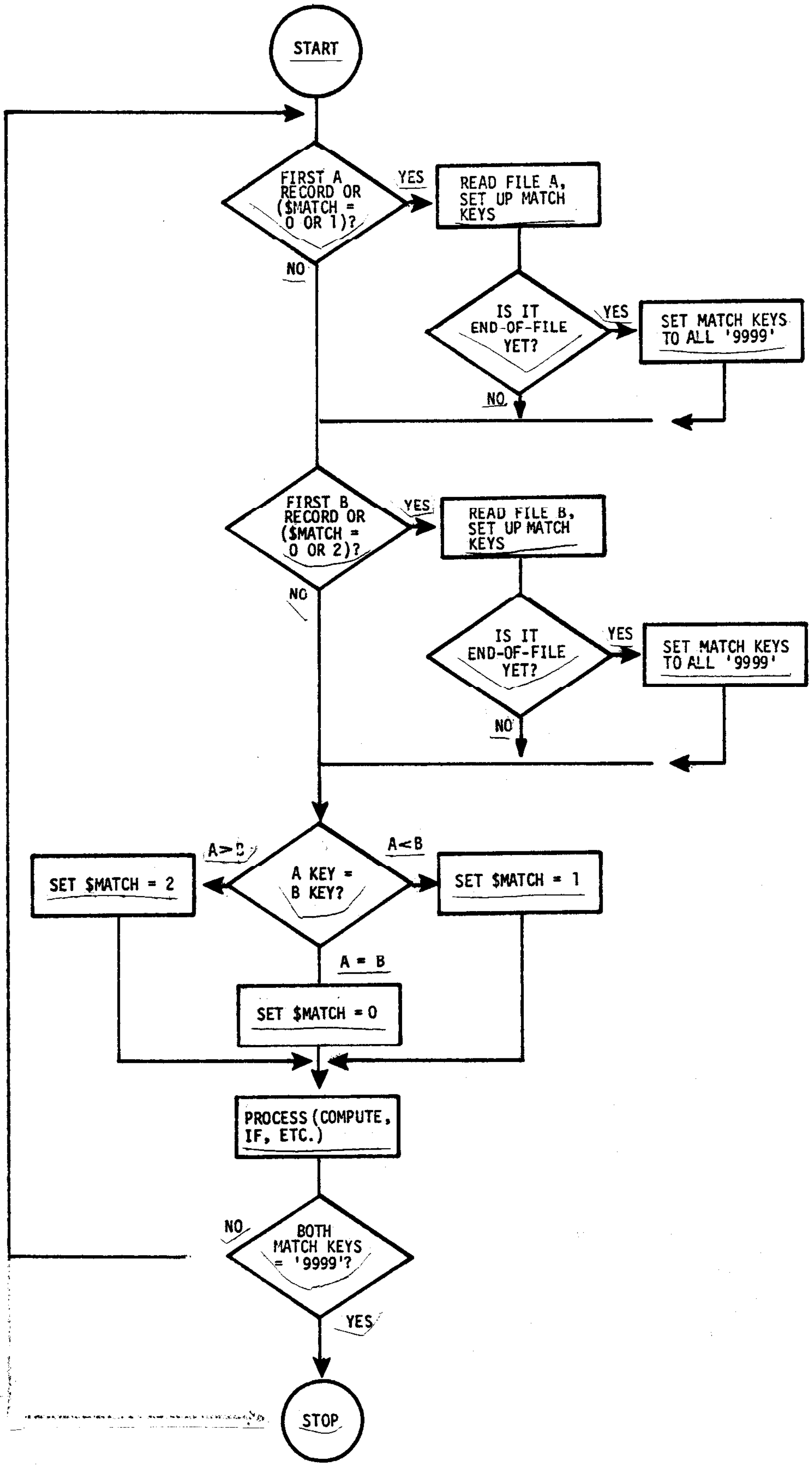
CENSPAC RECORD PROCESSING WITH SORT



CENSPAC RECORD PROCESSING WITH TOTAL


START
READ RECORD
END-OF-FILE YET?
YES
SET ALL TOTAL KEYS UNEQUAL
NO
PROCESS (COMPUTE,IF,ETC.)
TOTAL KEYS = OLD TOTAL KEYS?
NO
RETURN ONE LEVEL OF TOTALS
YES
AGGREGATE DETAIL
PROCESS (COMPUTE,IF,ETC.)
NO
TOTALS BEING PROCESSED?
YES
ADD CURRENT LEVEL OF TOTAL TO NEXT HIGHER LEVEL
MORE TOTALS TO BE PROCESSED?
YES
NO
END-OF-FILE YET?
YES
STOP
NO

CENSPAC RECORD PROCESSING WITH MATCH


START
FIRST A RECORD OR ($MATCH = 0 OR 1)?
YES
NO
READ FILE A, SET UP MATCH KEYS
IS IT END-OF-FILE YET?
YES
NO
SET MATCH KEYS TO ALL '9999'
FIRST B RECORD OR ($MATCH = 0 OR 2)?
YES
NO
READ FILE B, SET UP MATCH KEYS
IS IT END-OF-FILE YET?
YES
NO
SET MATCH KEYS TO ALL '9999'
A KEY = B KEY?
A>B
A<B
A = B
SET $MATCH = 2
SET $MATCH = 1
SET $MATCH = 0
PROCESS (COMPUTE, IF, ETC.)
BOTH MATCH KEYS = '9999'?
NO
YES
STOP