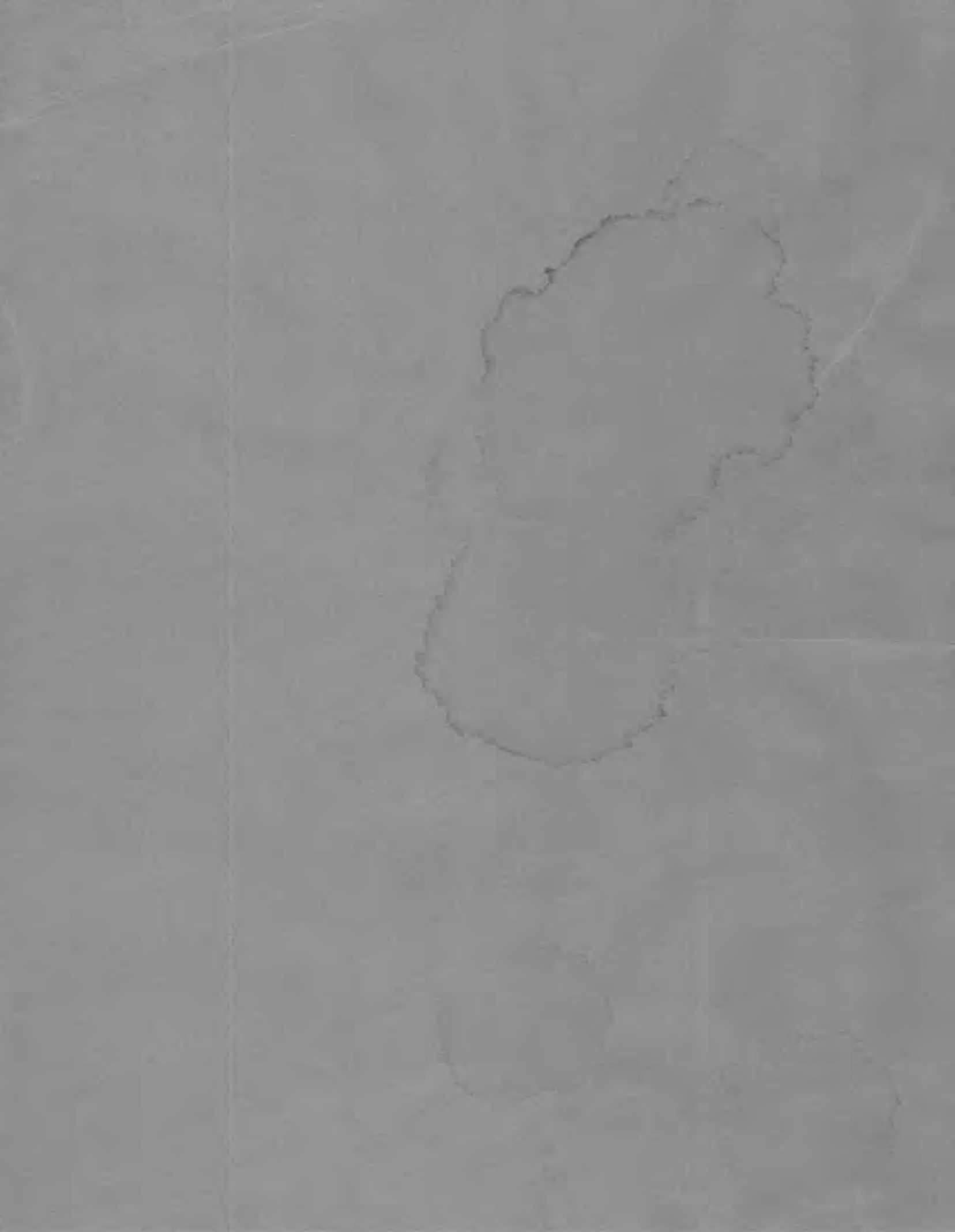
PACER—Data Entry, Retrieval, and Update for the National Coal Resources Data System (Phase I)

GEOLOGICAL SURVEY PROFESSIONAL PAPER 978



PACER—Data Entry, Retrieval, and Update for the National Coal Resources Data System (Phase I)

By S. M. CARGILL, A. C. OLSON, A. L. MEDLIN, *and* M. D. CARTER

GEOLOGICAL SURVEY PROFESSIONAL PAPER 978

A set of programs, written in FORTRAN IV, which extends the capability of GRASP and which has been developed in response to the need for a computer-based National Coal Resources Data System

UNITED STATES GOVERNMENT PRINTING OFFICE, WASHINGTON : 1976

UNITED STATES DEPARTMENT OF THE INTERIOR

THOMAS S. KLEPPE, *Secretary*

GEOLOGICAL SURVEY

V. E. McKelvey, *Director*

Library of Congress Cataloging in Publication Data
Main entry under title:

PACER.
(Geological Survey professional paper; 978)
Bibliography: p.
Supt. of Docs. SI 19.16:978
1. National Coal Resources Data System. 2. PACER (computer program) I. Cargill, S. M. II. Series: United States. Geological Survey. Professional paper; 978.
Z699.5.C57P3 029.7 75-619422

For sale by the Superintendent of Documents, U.S. Government Printing Office
Washington, D.C. 20402
Stock Number 024-001-02768-2

CONTENTS

ILLUSTRATIONS

TABLES

PACER—DATA ENTRY, RETRIEVAL, AND UPDATE FOR THE NATIONAL COAL RESOURCES DATA SYSTEM (PHASE I)

By S. M. CARGILL, A. C. OLSON, A. L. MEDLIN, and M. D. CARTER

ABSTRACT

PACER is a set of programs, written in FORTRAN IV, which extends the capability of GRASP and which has been developed in response to the need for a computer-based National Coal Resources Data System (NCRDS). PACER allows the user to enter data into one of three files, to search for and retrieve records using specific data elements, and to modify and update existing data records. All coal resource records west of the Mississippi River are grouped into WCOAL, whereas those east of the river are grouped into ECOAL. Each data record in WCOAL and ECOAL reflects a unique tonnage estimate of coal resource in a predefined category of thickness, overburden, and reliability of estimate. The USALYT file contains published coal analytical data and is structured to be as compatible as possible with the coal-resource tonnage files; however, it is not yet separated into east and west. A detailed description of the files is accompanied by user documentation for the use of the data files. A programmer's reference is also included to facilitate the installation and use of this software system on other computers.

INTRODUCTION

When probability of future energy shortages was recognized during the summer of 1973, considerable thought was directed to finding substitute sources of energy. Particular emphasis was given to studying coal in terms of that resource satisfying the energy needs of the United States during the interim period between the present dominant use of petroleum products and the advent of widespread use of fusion, solar, geothermal, and other forms of energy. The U.S. Geological Survey undertook to develop a computer-based National Coal Resources Data System (NCRDS) so that better estimates and evaluations could rapidly be made of this huge domestic resource. Development of the NCRDS began in July 1974 along two somewhat parallel paths which have become known as Phase I and Phase II.

Phase I includes retrieval and analysis software and three data bases:

1. WCOAL, which contains coal-resource tonnage estimates for deposits west of the Mississippi River;
2. ECOAL, which contains coal-resource tonnage estimates for deposits east of the Mississippi River;
3. USALYT, which contains published coal analytical data.

Information is stored in these files in an aggregated form, readily accessible for providing resource summaries throughout given geographic regions.

Phase I data records are stored in files which are accessed by a modified version of the Geological Retrieval and Synopsis Program (GRASP) (Bowen and Botbol, 1975) known as PACER: this is an acronym for Program to Analyze Coal Energy Resources. As of December 1975, the data files WCOAL and ECOAL contain nearly 25,000 logical records of coal-resource information for deposits east and west of the Mississippi River.

These two data files are from different sources. WCOAL consists of nearly 16,000 records and is a corrected and modified version of the Rocky Mountain Coal Reserve file received from the U.S. Bureau of Mines. ECOAL contains coal-tonnage records for the States east of the Mississippi River and probably will contain about 10,000 records when the file is completed in the spring of 1976.

Phase II data bases will include geologic information taken from point sources, such as core samples, field observations, and other forms of analysis. The software associated with Phase II will be designed to process these data so that they may be reduced to an aggregated or summary form amenable to Phase I applications.

PACER provides the user with access to the data in an interactive timesharing mode. The method of access to the master data records, using a remote terminal, is the main purpose of this report. Some new programming has been done to fill the specific requirements of the NCRDS and its principal users. These new programs are additions to GRASP, as it has been published, and, therefore, are included herein.

Examples of data are provided for clarity, and a sample session is included in a later section to help the user understand data relationships in the file. Because the file structures of ECOAL and WCOAL are identical, all references to WCOAL also apply to ECOAL, except where noted.

SCOPE AND PURPOSE

The scope of WCOAL, in terms of data elements, is limited to an estimate of coal tonnage for some defined area (State, county, township/range, and section) and contains specific definition of coal field, district, province, region, formation, coal bed, depth to coal coal (overburden), reliability of data, rank, and thickness of coal. Other incidental information includes the source document name from which the tonnage estimates were taken and its year of publication; topographic quadrangle name; base year of the tonnage estimate; and geologic age by System and Series. The scope of USALYT overlaps, to a large extent, the data in WCOAL. Replacing data such as depth, thickness, reliability, and tonnage estimates, are such data elements as sample identification and type, analysis identification and type, ash, moisture, fixed carbon, oxygen, hydrogen, and nitrogen values.

The purpose of this report is to provide a detailed description of how a user may access, enter, modify, and delete the data in the Phase I file. The design philosophy of PACER, as it pertains to WCOAL and USALYT, is also described in detail for the benefit of the programmer who wishes to modify the system or to install the software on his computer.

The purpose of the project, development of the NCRDS, has been to provide a means whereby estimates of coal resources, by a variety of characteristics, can be made accurately and quickly, and whereby the data may be updated easily with new information.

The PACER system was developed on Computer Science Corporation's INFONET timesharing system on a UNIVAC 1108 under the company's General Programming System (GPS). Development of PACER took approximately 2 man-years.

ACKNOWLEDGMENTS

Development of Phase I of the National Coal Resources Data System has been facilitated by the use of the Rocky Mountain Coal Reserve file from the U.S. Bureau of Mines MERIT (Mines Energy Resources, Information, and Transportation) System. This file of some 5,000 logical records formed the basis of WCOAL, and as such, has been the nucleus for Phase I data development.

The authors wish to express their thanks to Roger W. Bowen, particularly for his invaluable help in converting the original data into GRASP formats and generally for his assistance in understanding the GRASP system. We also wish to thank Joseph Moses Botbol for his continuing encouragement to bring the system into production.

DEFINITION OF TERMINOLOGY

Definitions of some computer terminology are given to clarify its use in this report.

Interactive. This term implies, in a broad sense, reciprocal activity between user and computer. It also implies a response to a user-initiated transaction to permit appropriate user action based upon timely computer response. The word is closely associated with a sense of immediacy of response time: interactive is a matter of degree; therefore, a system is less interactive the longer it takes to get a response.

Timesharing. This term, in its simplest definition, is a computing technique in which several users may utilize a computer concurrently for input, processing, and output functions.

Dictionary. The dictionary, or dictionary file, contains the alphanumeric entries which are associated with a data record. The data record contains not the value itself but a "pointer." The pointer is a number that indicates at which sequential position in the dictionary file may be found the correct character string value for that data field.

Master file. The master file contains the records of the data base. The fields within the record contain integer values (for integer data), floating-point values (for decimal data), and integer pointers (for alphanumeric data). The file is structured slightly differently from a conventional GRASP numeric master file in that multiple-choice-type items are not used, nor is the master file compressed.

Mask file. The "Mask file contains the item names, item types (integer, real, character string, multiple choice, and qualified real), and pointers to the first entry in the dictionary file for each character-type item" (Bowen and Botbol, 1975, p. 3).

Definitions file. This file contains the acronyms used for data variable and the meaning (definition) of each acronym. Other record length parameters are also defined in this file; however, the reader is referred to the GRASP report (Bowen and Botbol, 1975) for a detailed description.

Additional data processing definitions may be found in "Computer Dictionary" (Sippl and Sippl, 1974), or in Calkins and others (1973).

SYSTEM DESCRIPTION

A brief description of procedures and commands is provided below for the PACER version of the GRASP system. The reader is referred to the more detailed description of GRASP (Bowen and Botbol, 1975) for further clarification of procedures and commands. A detailed description of the data bases, WCOAL and USALYT, may be found following descriptions of the programs.

PACER SEARCH AND RETRIEVAL OPERATION

After a successful hookup and log-on procedure has been completed, the user requests the program by typing in the word PACER following the system prompt, !. At this point, PACER types a statement informing the user which data files are available for use and giving a brief description of the data contained therein, and requests the user to name the data file to be accessed.

The user may select one of several commands to perform various functions once the system has prompted with, ENTER COMMAND. Table 1 lists these commands and their meanings. The following is a detailed explanation of some of the commands.

COND (condition) requires three entries: acronym/relation/value. The acronym refers to the data variable name (for example, STATE), and value is the desired value of the variable (for example, ALASKA). The relation between these two may be expressed by one of the following seven operators:

EQ ------equal to.
NE ------not equal to.
GT ------greater than.
LT ------less than.
LE ------less than or equal to.
GE ------greater than or equal to.
BE ------between or equal to.

Examples: STATE EQ WYOMING
COUNTY EQ JOHNSON
YEAR BE 1967,1970
RANK LE SUBBIT
TONNAGE GT 350.00

Note that all the relational operators may be used for both alphabetic and numeric data. This is valid for alphabetic data only if they are ordered in such a way that their sequence in a list is significant. For example, rank of coal (RANK) may be logically ordered in a list so that anthracite (ANTH) tops the list, followed by semi-anthracite (SEMI ANTH), bituminous (BIT), low-volatile bituminous (LV BIT), medium-volatile bituminous (MV BIT), high-volatile bituminous (HV BIT), and lignite (LIGNITE). If the condition were set so that RANK EQ LIGNITE, then all lignite records would be retrieved; if RANK GT MV BIT were entered, then only LV BIT, SEMI ANTH, and ANTH records would be retrieved; if RANK BE MV BIT,ANTH were entered, then all records for anthracite, semi-anthracite, low-volatile bituminous, and medium-volatile bituminous coal would be retrieved.

Conditions may reflect repeating acronyms and relational operators, and each condition is prefaced by A, B, C, D, . . ., Z (up to 26 conditions). Thus, CONDition A might be STATE EQ MONTANA, and CONDition B, perhaps STATE EQ WYOMING. It is up to the user to associate these conditions with the proper logic to effect the desired retrieval, as described below in the command, LOGIC.

PACER will keep printing the next available alphabetic character as it expects another condition. If no more conditions are to be entered, the user strikes carriage return (CR) without any entry.

LOGI (logic) provides the user with a way to associate any two or more conditions specified in the CONDition command. Three Boolean logical operators are used to connect one condition with another:

.AND. or *
.OR. or +
.NOT. or −

Note that the word operators are bracketed by periods, but that the equivalent symbol operators are not. The structure of a LOGIc command allows the user to string together as many as 26 conditions (A,B,C, . . . ,Z) with operators as follows:

A.AND.B+C*D

TABLE 1.—*List of PACER commands*

```
ENTER COMMAND: HELP

THE COMMANDS WHICH MAY BE ISSUED (AND THEIR MEANING) ARE LISTED BELOW:


COND - INITIATES THE REQUEST FOR RETRIEVAL CRITERIA TO BE ENTERED IN
       THE FORM: NAME REL VALUE
LOGI - INITIATES THE REQUEST FOR A LOGICAL EXPRESSION TO BE ENTERED
       USING LOGICAL OPERATORS.
SEAR - INITIATES THE SEARCH OF A FILE BASED UPON PREVIOUSLY ENTERED
       CONDITIONS AND LOGIC.
LIST - ALLOWS THE USER TO LIST SELECTED VALUES (VARIABLE NAMES WILL
       BE ASKED FOR) IN A FILE.
FILE - ALLOWS THE USER TO SELECT OR CHANGE THE DATA BASE TO BE USED.

QUIT - TERMINATES THE SYSTEM.  ENTERING ! IN RESPONSE TO A PROMPT WILL
       ALSO STOP THE SYSTEM.
NAME - USED TO PRINT ITEM NAMES, THEIR TYPES AND DEFINITIONS IN A
       SELECTED SET OF GROUPS.
HELP - USED TO OBTAIN THE ABOVE COMMAND DEFINITIONS.

REVI - LISTS THE FILES WHICH HAVE BEEN USED AS WELL AS THE CONDITIONS
       AND LOGIC ENTERED.
DUMP - PRINTS ALL ITEMS OR SELECTED ITEMS PRESENT FOR EACH RECORD IN A
       SELECTED FILE.  WAITS AFTER EACH N LINES.
FUNC - PROVIDES FOR THE COMPUTATION OF FUNCTIONS ON ITEMS IN A DATA
       SET (OR FILE).
MERG - COMBINES THE CONTENTS OF SEVERAL SELECTED SUBFILES INTO A
       SINGLE SUBFILE.
TABL - PERMITS THE SELECTION OF SPECIALLY SORTED AND FORMATTED
       TABULAR OUTPUT DISPLAYS.
UPDA - PERMITS THE ADDITION, THE DELETION, OR THE MODIFICATION OF RECORDS
       OR PORTIONS OF RECORDS BELONGING TO THE MASTER FILE.
```

Parentheses may be used to group specific logical relationships, as in the following example:

(A.AND.B+C)*D

The order in which conditions are satisfied is given by the Boolean operators: .NOT., .AND., and .OR.

If parentheses are used, they take precedence over the order of the operators. In the example above, the expression (A.AND.B+C) is evaluated before it is combined with D.

SEAR (search) is a function that allows the user to actually retrieve the desired data records on the basis of the conditions that were logically connected. The user must define a new file name which will contain the retrieved records. The new data file thus created is a subfile of WCOAL or USALYT and may be saved for future study or may be deleted at the termination of the session using the QUIT command. If the file is to be deleted after conclusion of the session, the deletion is performed external to PACER execution under control of the operating system: in the case of INFONET, this is done with a !DROP "filename" command directive.

The user may access records in a previously created subfile. He may also combine records residing in two separate subfiles, each of which were retrieved by two separate searches of the main file, into a single subfile through the use of the MERGe command. The user may refine the data contained in a subfile by generating new conditions and logic and by performing a search of the subfile to satisfy those conditions and that logic. A new subfile is then created. Future use of any subfile must be preceded with

a reference to one of the available main files after the ENTER DATA BASE NAME command in order that all dictionaries are properly linked with the data.

The UPDA (update) command is described in detail in a later section because it is a new and extensive modification of the original GRASP system.

Other command options available (table 1) are self-explanatory at the time of operation. The user should note that the QUIT command terminates PACER sessions. The computer system will then prompt the user with the names of user files that have been created with a request to specify those files that are to be saved, perform the indicated file maintenance, and followed by a !. The response of OFF breaks the link between the terminal and the computer.

MERIT DATA FILE

The Rocky Mountain Coal Reserve data file of the MERIT System was acquired from the Bureau of Mines when it contained some 5,200 records of coal resources for States west of the Mississippi River. Data records are entered into the MERIT file in an 80-column card image format, and are put out also in an 80-column card image format. Figure 1 shows that the system requires the data to be entered on four different card types, which are identified in column 9 of the form. The first two cards contain data pertinent to location and stratigraphy, and the third card contains bibliographic data. The fourth card contains multiple DCT codes and tonnage estimates. In the DCT code, the D stands for depth to coal, the C stands for class, and the T stands for thickness of the coal seam; these codes are equivalent to Branch of Coal Resources classification nomenclature, ORT (overburden, reliability, and thickness). The four physical records (card types) constitute one logical record.

WCOAL DATA FILE

The WCOAL file is a modified version of the MERIT file in that several Bureau of Mines data elements have been removed and new Geological Survey data elements have been added. The following data fields have been retained from the indicated cards of the MERIT record:

Card No. 1 ---- P.M.
TOWNSHIP NO.
RANGE NO.
E/W
SEC. NO.
COAL FIELD NAME
FORMATION NAME

Card No. 2 ---- SEAM NAME
RANK NAME

Card No. 3 ---- DATA SOURCE
YR. OF PUBL.

Card No. 4 ---- DCT, TONNAGE } all values.

In addition, three other Bureau of Mines variables have been retained, the names of which do not appear in the data entry form shown in figure 1, but which are a part of the MERIT file. From card No. 3, Quadrangle Name (QUAD) and Base Year (BYEAR) have been retained. The first three variables of each card, FIPS State code, FIPS county code, and a record sequence number, have been lumped together to form a single WCOAL variable called ID.

All other data fields have been deleted from each of the MERIT records because either the data contained in the field are inconsistent, or the data are infrequently, if ever, entered for the variable, or they are anticipated not to be applicable to Phase I of the NCRDS. The remaining data fields and the additional new fields, when strung together, form a single physical and logical record. In the MERIT system, the State code, county code, and sequence number together are repeated from card to card as the logical link between the several physical records.

In WCOAL, however, the emphasis has changed from that of location to the use of ORT codes and the corresponding tonnage values as the basis for a unique record. Only one set of ORT codes and one tonnage value are provided per record. The effect of this change has been to create WCOAL as a file of 15,972 unique tonnage records.

Each record has been expanded from the abbreviated Bureau of Mines record to include eight new variables:

STATE ---- State name (not FIPS code).
COUNTY ---- County name (not FIPS code).
AAPGPRV ---- AAPG province number.
COALPRV ---- Coal province name.
REGION ---- Coal region name.
DISTRCT ---- Local area designator.
SYSTEM ---- Geologic age: System.
SERIES ---- Geologic age: Series.

WCOAL, therefore, is composed of records from Bureau of Mines and Geological Survey files. Each record is organized into four categories:

Location information.
Geologic designations.
Bibliographic information.
Resource delimiters.

Table 2, list of WCOAL record variables, provides a complete list of the variables, in their proper sequence, that make up one WCOAL record. Within each category is listed the variable name or mnemonic, its data type code, and a description. PACER allows one of the three data type codes to be associated with any variable:

I = whole numbers (integer values)
R = decimal numbers (real values)
A = alphanumeric strings

The appropriate type code for each item is shown in table 2. Although each variable is briefly described in table 2, some further elaboration of the data elements is useful, especially as to the purpose of some of the items. Therefore, each variable is described below, with specific reference to appendix A, which shows the names each variable can assume as listed in WCOAL dictionaries.

STATE name is used in preference to the FIPS (U.S. Natl. Bur. Standards, 1973) State code because the name is more readily known to the user.

COUNTY name is also used instead of the FIPS county code. A list of all county names and the associated State is available from a U.S. National Bureau of Standards (1973) publication and is not provided.

PMERID is the principal meridian from which a township/range survey was begun, and is given as a numeric code in WCOAL records. Appendix A.1 lists the meridians and baselines of the United States rectangular surveys, as provided by the Bureau of Land Management, as well as the Bureau of Mines code associated with each name.

TWNSHIP is the township number of the township/range survey. Three digits are provided for the town-ship number, allowing values from 001 to 999. However, no provision has been made for half townships.

NS is the township direction, north or south.

RANGE is the range number, and, like TWNSHIP, three digits are allowed with no provision for half ranges.

EW is the range direction east or west.

SECTION is the section number within a township/range unit. This is a two-digit value usually between 01 and 36.

AAPGPRV is the AAPG (American Association of Petroleum Geologists) province number (Meyer, 1970). The purpose of using this number is to provide a link between the National Coal Resources Data System and other national energy data systems. The associated geologic provinces, districts, basins, and so on, do not necessarily correspond from one resource to another.

COALPRV is the coal-province name. Figure 2 shows the coal provinces of the conterminous United States, Alaska being an additional province.

REGION is the coal-region name and is a subset of coal provinces. Figure 3 shows the coal regions of the conterminous United States, and figure 4 shows the coal regions for the Alaska province. The coal regions for the lower 48 States correspond to those given by Trumbull (1960), but no previous designations have been given to the regions of the Alaska province as shown in figure 4, although the base map from Barnes (1961) has been used. Appendix A.2 provides a list of the region names available for use with WCOAL.

FIELD is the coal-field name. A complete list of coal-field names used in WCOAL may be found in appendix A.3.

DISTRCT is a local area designator applicable mainly to the Alaskan areas. These names are, in many cases, interchangeable with field names, implying no hierarchical difference between district and field. Appendix A.4 lists all names which are used with WCOAL and which are in the coal district dictionary.

FORMATN is the formation name. A complete list of formation names used in WCOAL may be found in appendix A.5.

BED is the coal-bed name. A complete list of coal-bed names used in WCOAL is provided in appendix A.6.

Neither FORMATN nor BED contains a complete list of formation or coal-bed names. The list currently in use is predominantly that compiled by Bureau of Mines engineers using Geological Survey publications, and other documents, for the compilation. This list will be modified for use in NCRDS.

SYSTEM is the geological age designation. The following system names are in use with WCOAL: Tertiary, Cretaceous, Jurassic, Triassic, Permian, Pennsylvanian, and Mississippian.

SERIES is the name given to a divided system.

STATE _____ COUNTY _____ CODED BY _____ DATE _____

STATE CODE	COUNTY CODE	I.D.	CARD TYPE	CARD NO.	ACTION	P.M.	TOWNSHIP NO.	N/S	RANGE NO.	E/W	SEC. NO.	COAL FIELD NAME	IFOC CODE	FORMATION NAME	IFOC CODE	CO. PROP. FACTOR	ACREAGE	
			1															
1-2	3-5	6-8	9	10	11	12-13	14-16	17	18	19-21	22	23	24-25	26-40	41-43	44-55	59-61	62-64	65-72	73-80

					SEAM NAME	IFOC CODE	ARESCO CODE	RANK NAME	RANK CODE		SULFUR			WTD. AVG. THICKNESS	AVERAGE THICKNESS			
											AVG.	HI	LO		1st RANGE	2nd RANGE	3rd RANGE	
			2															
1-2	3-5	6-8	9	10	11	12-31	32-34	35-38	39-48	49-50	51	52-53	54-55	56-57	58	59-62	63-66	67-70	71-74	75-80

					DATA SOURCE	IFOC CODE	YR. OF PUBL.	FINDER CODE	
			3						
1-2	3-5	6-8	9	10	11	12-31	32-35	36-39	40-48	49-80

					DCT	TONNAGE	DCT	TONNAGE	DCT	TONNAGE	DCT	TONNAGE	DCT	TONNAGE	DCT	TONNAGE	
			4														
1-2	3-5	6-8	9	10	11	12-14	15-22	23-25	26-33	34-36	37-44	45-47	48-55	56-58	59-66	67-69	70-77	78-80

DCT CODES

Depth: 1. Ø-1ØØØ; 2. 1ØØØ-2ØØØ; 3. 2ØØØ-3ØØØ; 4. >3ØØØ; 5. Ø-2ØØØ; 6. Ø-3ØØØ; 7. Strippable; 8. Unclassified

Class: 1. Measured; 2. Indicated; 3. Inferred; 4. Measured and indicated; 5. Unclassified

Thickness: 1. 14"-28"; 2. 28"-42"; 3. >42"; 4. 2.5'-5'; 5. 5'-1Ø'; 6. > 1Ø'; 7. Unclassified; 8. Classified by zone

GPO 632-363

FIGURE 1.—MERIT data-entry form.

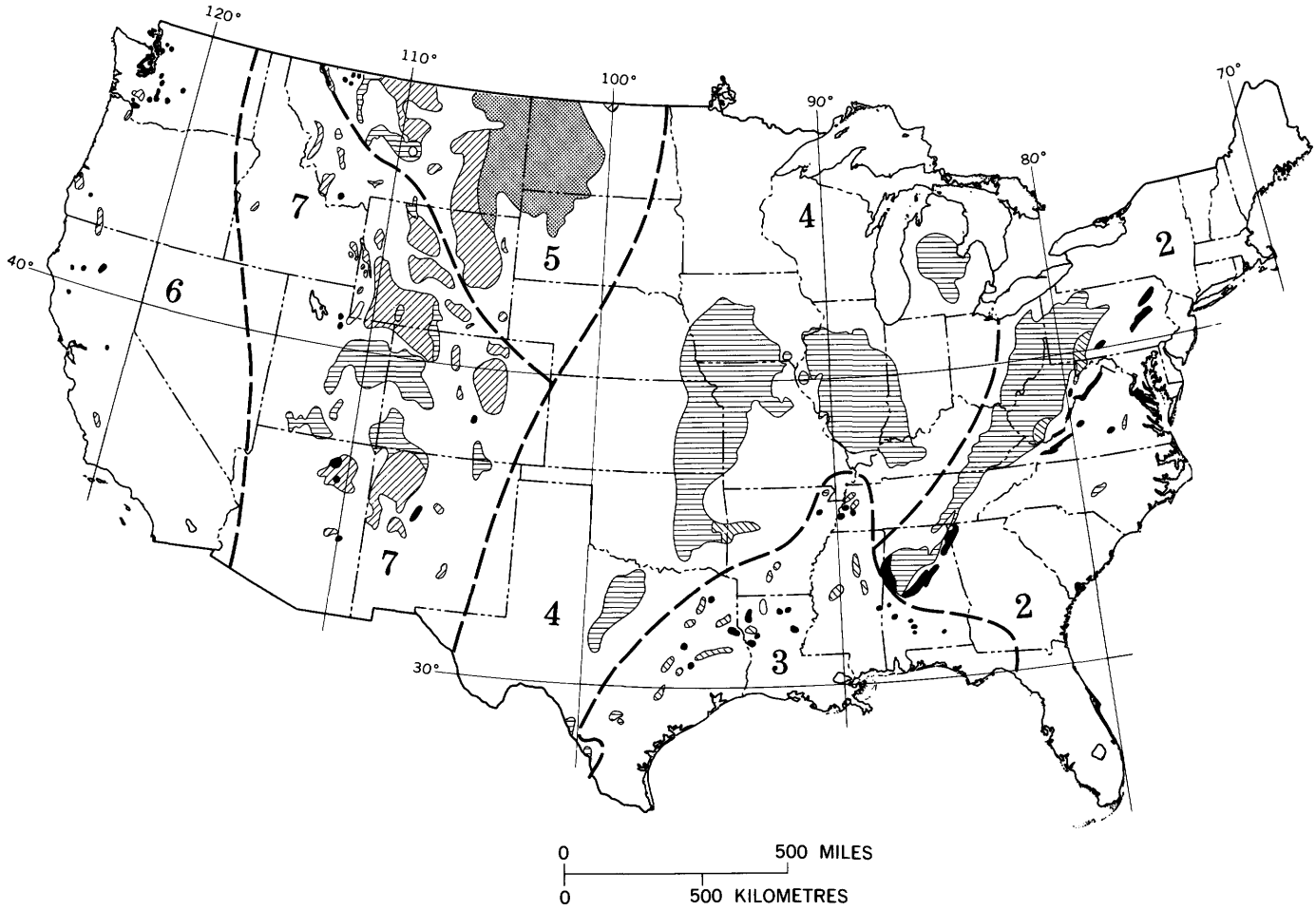
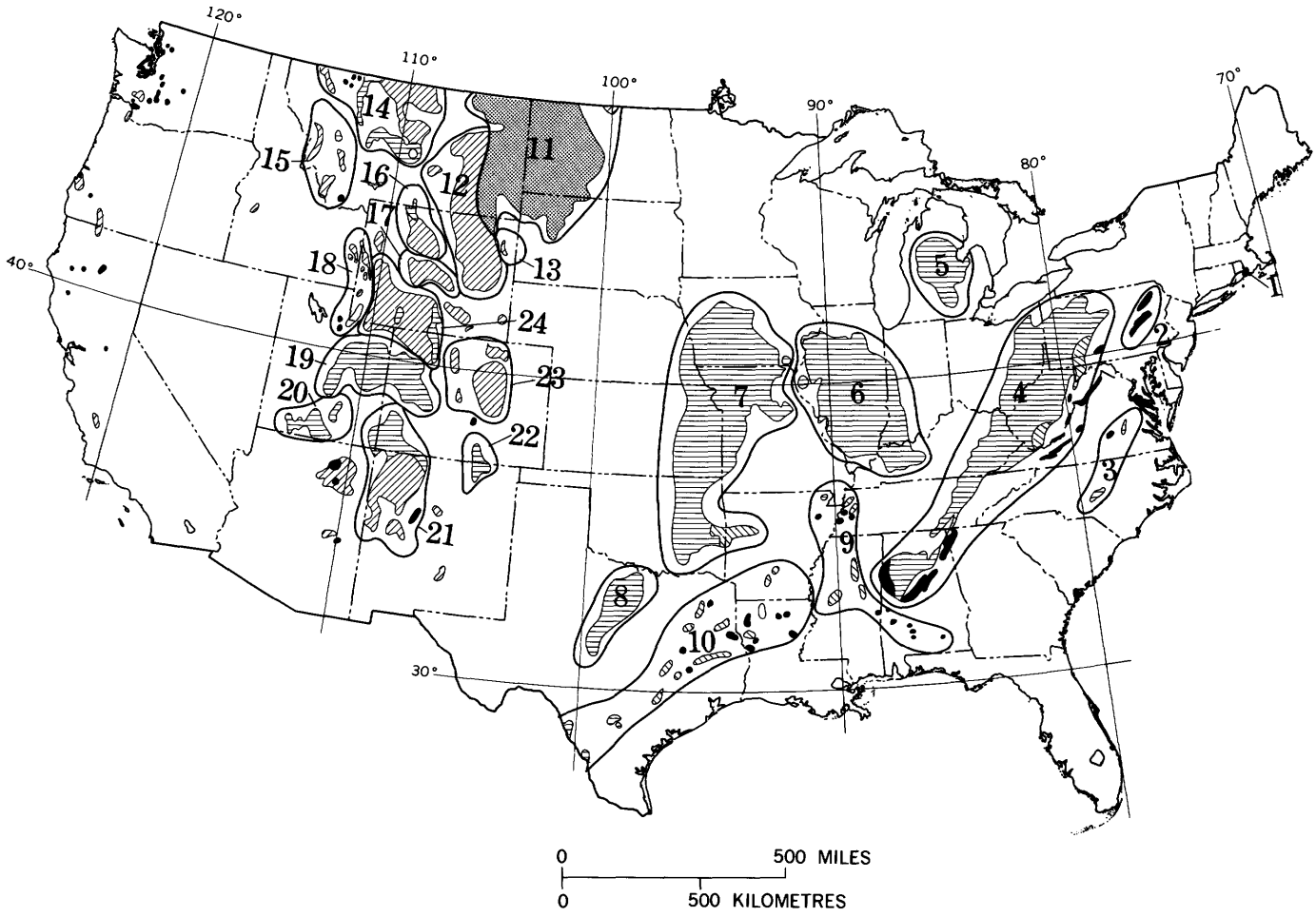


FIGURE 2.—Coal provinces of the conterminous United States. (Modified from Trumbull, 1960.)



EXPLANATION

Regions

1	Rhode Island meta-anthracite	13	Black Hills
2	Pennsylvania anthracite	14	North Central
3	Atlantic coast	15	Tertiary lake beds
4	Appalachian	16	Bighorn Basin
5	Northern	17	Wind River
6	Eastern	18	Hams Fork
7	Western	19	Uinta
8	Southwestern	20	Southwestern Utah
9	Mississippi	21	San Juan River
10	Texas	22	Raton Mesa
11	Fort Union	23	Denver
12	Powder River	24	Green River

See figure 2 for explanation of pattern areas

FIGURE 3.—Coal regions of the conterminous United States. (Modified from Trumbull, 1960.)

TABLE 2.—*List of WCOAL variables*

```
LOCATION INFORMATION
GEOLOGIC DESIGNATIONS
BIBLIOGRAPHIC INFORMATION
RESOURCE DELIMITERS

STATE    A  STATE NAME
COUNTY   A  COUNTY NAME
PMERID   I  PRINCIPAL MERIDIAN
TWNSHIP  I  TOWNSHIP NUMBER
NS       A  DIRECTION OF TOWNSHIP (N OR S)
RANGE    I  RANGE NUMBER
EW       A  DIRECTION OF RANGE (E OR W)
SECTION  I  SECTION NUMBER
AAPGPRV  I  AAPG PROVINCE NUMBER

COALPRV  A  COAL PROVINCE NAME
REGION   A  COAL REGION NAME
FIELD    A  COAL FIELD NAME
DISTRCT  A  LOCAL AREA DESIGNATOR
FORMATN  A  FORMATION NAME
BED      A  COAL BED NAME
SYSTEM   A  GEOLOGIC AGE: SYSTEM
SERIES   A  GEOLOGIC AGE: SERIES

QUAD     A  TOPOGRAPHIC QUADRANGLE NAME
BYEAR    I  BASE YEAR FOR TONNAGE ESTIMATES
               00 MEANS ORIGINAL DATA
               51 MEANS DATA TAKEN AS OF 1951
SOURCE   A  SOURCE DOCUMENT
YEAR     I  PUBLICATION YEAR OF SOURCE DOCUMENT

THICKNS  I  COAL BED THICKNESS CODE
               1 = 14 TO 28 INCHES
               2 = 28 TO 42 INCHES
               3 = GREATER THAN 42 INCHES
               4 = 2.5 TO 5 FEET
               5 = 5 TO 10 FEET
               6 = GREATER THAN 10 FEET
               7 = UNCLASSIFIED
               8 = CLASSIFIED BY ZONE
OVRBRDN  I  OVERBURDEN THICKNESS IN FEET
               1 =    0 TO 3000
               2 =    0 TO 2000
               3 =    0 TO 1000
               4 = 1000 TO 2000
               5 = 2000 TO 3000
               6 = GREATER THAN 3000
               7 = STRIPPABLE
               8 = UNCLASSIFIED
RELIABL  I  RELIABILITY CODE
               1 = MEASURED
               2 = MEASURED AND INDICATED
               3 = INDICATED
               4 = INFERRED
               5 = UNCLASSIFIED
RANK     A  RANK OF COAL
               ANTH       = ANTHRACITE
               SEMI ANTH  = SEMI-ANTHRACITE
               BIT        = BITUMINOUS
               LV BIT     = LOW-VOLATILE BITUMINOUS
               MV BIT     = MEDIUM-VOLATILE BITUMINOUS
               HV BIT     = HIGH-VOLATILE BITUMINOUS
               HV BIT A   = HIGH-VOLATILE BITUMINOUS A
               HV BIT B   = HIGH-VOLATILE BITUMINOUS B
               HV BIT C   = HIGH-VOLATILE BITUMINOUS C
               SUBBIT     = SUB-BITUMINOUS
               SUBBIT A   = SUB-BITUMINOUS A
               SUBBIT B   = SUB-BITUMINOUS B
               SUBBIT C   = SUB-BITUMINOUS C
               LIGNITE    = LIGNITE
TONNAGE  I  COAL RESOURCE IN MILLIONS OF SHORT TONS. ... A
            TONNAGE RECORD EXISTS FOR EVERY UNIQUE COMBINA-
            TION OF THICKNESS CODE, OVERBURDEN CODE, RELI-
            ABILITY CODE, RANK CODE, AND LOCATION CATEGORY,
            AS WELL AS CERTAIN STRATIGRAPHIC DESIGNATIONS.
```

Series names in use with WCOAL are Eocene, Lower, Middle, Miocene, Oligocene, Paleocene, Pliocene, and Upper.

Table 3 shows the ordered relationship between System and Series. Note that the repetition of Series names from one System to another has no effect on their use, because each requires only one dictionary entry.

TABLE 3.—*Geological age names used with WCOAL and ECOAL*

System	Series
Tertiary	Pliocene Miocene Oligocene Eocene Paleocene
Cretaceous	Upper Lower
Jurassic	Upper Middle Lower
Triassic	Upper Middle Lower
Permian	Upper Lower
Pennsylvanian	Upper Middle Lower
Mississippian	Upper Lower

QUAD is the topographic quadrangle name and refers to the quadrangle for which the resource tonnage was made. Often the quad name is not known or is not unique to the tonnage record. Therefore, the present list of quadrangle names given in appendix A.7 is very brief.

BYEAR is the base year for which estimates were made of the tonnages of coal. As an example, 51 indicates that the estimates are for remaining resources as of 1951; 00 indicates that the tonnage estimate is of original coal resources.

SOURCE is the publication from which the data were taken. Appendix A.8 lists these publications.

YEAR is the publication year of the source document.

THICKNS is the coal thickness code. The name of this variable corresponds to the Bureau of Mines "T" for thickness in the DCT code. The possible values of THICKNS are shown in table 2.

OVRBRDN is the overburden thickness code. The name of this variable corresponds to the Bureau of Mines "D" for depth in the DCT code, but the categories have been renumbered to permit a search over a range of overburden depths. The possible values of OVRBRDN are shown in table 2.

RELIABL is the reliability code given to a tonnage estimate. The name of this variable corresponds to the Bureau of Mines "C" for class in the DCT code. It also has been renumbered to permit a search over a given range. The possible values of RELIABL are shown in table 2.

RANK is the name given for the quality of coal in terms of energy content. Fourteen rank categories are provided in the list; they range from anthracite to lignite. The order of these ranks is such that the user may enter ranges of rank (for example, RANK BE HV BIT,SUBBIT) as a condition.

TONNAGE is the estimated coal resource in millions of short tons to two decimal places. A tonnage value exists for every unique combination of thickness code, overburden code, reliability code, and location.

USALYT DATA FILE

A review of the published coal analytical data revealed that it was not feasible to add this type of data to the existing area/tonnage records of the WCOAL file, as there were few, if any, areas of direct correlation. Therefore, a separate data base of published coal analytical data is maintained as the file USALYT on the PACER system; it is accessed by specifying USALYT for the ENTER DATA BASE NAME command. This file uses the same dictionaries as WCOAL; however, it has separate mask and definitions files for the 46 data items, as shown in table 4.

The two files were structured as closely alike as possible to facilitate retrieval and correlation of data by the user. A comparison of tables 2 and 4 shows that the location, geologic, and bibliographic fields are the same as in WCOAL, with the exception of the item BYEAR (base year). BYEAR is not applicable to the type of data in USALYT. For this reason, the resource delimiter fields also have been deleted, with the exception of RANK. The following is an expansion of the brief definitions given in table 4 of the additional data items for USALYT:

ANIDA is the alphabetic part of the analysis identification number. Some of the reported analyses have alphanumeric identification numbers. Because PACER treats alphanumeric data as dictionary items, the identification number is recorded in two parts. This dictionary contains the 26 letters of the alphabet.

ANIDN is the numeric part of the reported analysis identification number.

SAMPTYP is a numeric code to indicate the type of sample on which the analysis was done.

ANLYTYP is a numeric code to represent the condition of the sample when it was analyzed.

VALREP is a numeric code indicating the type of data that the individual values represent, such as an average of several samples or a single sample.

TRACE indicates by Y=yes or N=no whether the analysis included trace-element values. These values are not included in the data file, because their occurrence in the Phase I published analytical data is rare but could be retrieved from the original source.

HGRIND represents the result of a Hardgrove grindability test on the sample.

OTHER indicates by Y=yes or N=no whether other types of tests were reported in the analysis. Results of these tests are not in the data file but could be retrieved from the data source.

BTU is the energy value of the coal reported in British thermal units. If the value is reported in other units it has been converted.

ASH: [DEFORM], [SOFT], [FLUID] indicate the temperature in degrees Fahrenheit at which the ash deforms, softens, or fluidizes. If the reported values were in another form, they have been converted.

The final 12 items are the proximate and ultimate analysis values. All items are reported in percent to one decimal place.

ENTRY OF NEW DATA

Since the creation of the initial WCOAL file from the U.S. Bureau of Mines data tape, data entry has been accomplished using a SYCOR model 340 programmable terminal. The data entry format for WCOAL (fig. 5) is displayed on the terminal CRT, and the data items are entered and checked by programs, as described below. These data-entry controls are indicated by the character or number immediately following the data label, which is enclosed in brackets. The SYCOR programming system reserves symbols W, X, Y, and Z to call user-writer field programs. Therefore, these symbols below may call different programs, according to the data-field name. The data fields have been arranged to allow all data belonging to one record to be displayed on the screen at one time. The data-entry format is programmed to move the cursor to each data field according to groups of related information. Data-entry control characters for the WCOAL format include:

STATE and COUNTY --5] Alphabetic data must be entered.

PMD ----------X] The cursor skips to the APG data item, if a blank or zero (0) is entered, because it is assumed that no township and range information is available if the principal meridian is unknown.

TWN and RNG ---------N] A numeric entry is required.

NS and EW ----------W] The value entered is compared with a table of acceptable values, and an error message is displayed if no match is found.

PRV, FLD, FMN -------A] The data must be alphabetic.

RGN, DST, BED -------W] The cursor skips to the next line in order to maintain logical data entry from reporting forms.

THK, OVB, REL -------W] The cursor skips intervening data fields so that these items may be entered consecutively. Format control then passes to the next entry on the reporting form.

SYS, SER, RNK -------W] The cursor skips fields on the screen so that these items can be entered consecutively. Each of these items is also compared with a table of acceptable values in order to reduce data-entry errors.

QDR ----------M] An alphanumeric value may be entered.

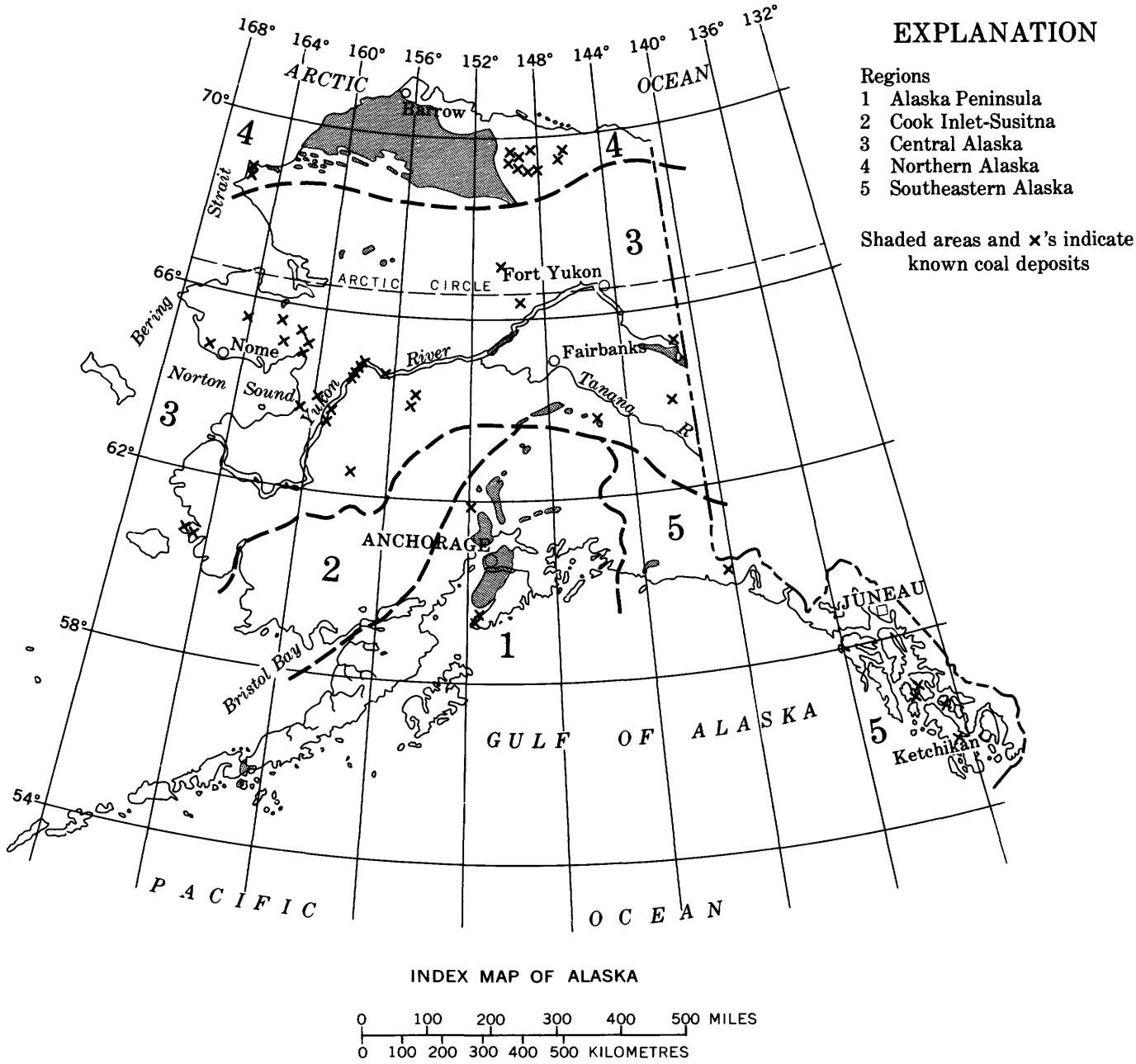


FIGURE 4.—Coal regions of the Alaska coal province. (Modified from Barnes, 1961.)

BYR ----------X] The data value is checked against an acceptable range of values, and an error message is printed if the comparison fails.

SRC ----------A] Some alphanumeric data must be entered in this field.

YR ----------X] The cursor skips to the ID field, after the user presses the TAB/SKIP key.

ID ----------X] The cursor skips to the TON field, after the user press the TAB/SKIP key.

TON ----------Z] The data entry will be checked for a value with two decimal places. Any other value range will cause an error message to be printed.

[*], [C] These symbols are constant field indicators which contain a line feed character for continuous transmission, because the record exceeds the 256-character buffer available in the SYCOR terminal.

Data entry to file USALYT requires a two-page SYCOR format shown in figure 6. Page 1 has data-entry restrictions similar to the WCOAL format. Page 2 data fields include the following controls:

ANIDA ----------A] Only alphabetic data entry is acceptable.

ANIDN, S-TYPE -----N] Only numeric data can be entered.

A-TYPE, VAL\REP, HRDGV-AGRIND, TRACE, OTHER. A] Alphabetic data only.

BTU,ASH (DEF SOF FLD), FREE-SWELL. N] Numeric data.

MOISTUR, VOL-MAT, FIXEDC, ASH, HYDROGN, CARBON, NITROGN, OXYGEN, SULFUR, SULFATE, SULFPYR, SULFORG. Z] The format checks for a numeric entry with one decimal place. Any other entry will result in an error message. The \ character indicates succeeding data fields with the same "Z" control.

USE OF THE UPDATE OPTION

The update software permits the user to edit and update data in the master file while operating in an interactive mode from a remote terminal. The following update procedures are available for the user to:

1. Add new records to the master file;
2. Delete existing records from the master file;
3. Sequentially review and modify records existing in a subfile of the master file and, upon completion of the review, post the subfile onto the master file;
4. Select, by a key, a record residing in a subfile of the master file, review the contents, modify the data, and post the individual record onto the master file;
5. Modify the value of a given data element for all the records existing in a specific subfile and then post the contents of this subfile onto the master file.

To use the update system, the user must first access the host computer from his terminal. After the appropriate identification has been entered, the computer will begin a new line with the character ! which is the prompt for the user to type in the program name, PACER, followed by a carriage return. This command accesses the PACER program and begins execution with the message:

WELCOME TO THE USGS "PACER" VERSION OF THE "GRASP" RETRIEVAL SYSTEM. THE FOLLOWING DATA BASES ARE AVAILABLE:

ECOAL—USGS EASTERN US COAL RESOURCES DATA
WCOAL—USGS WESTERN US COAL RESOURCES DATA
USALYT—PUBLISHED US COAL ANALYTICAL DATA

The user must specify one of the above data bases at the prompt for a data base name, ENTER DATA BASE NAME. A new master file data base may be selected at any time during a user session by entering the word, FILE, following the prompt, ENTER COMMAND.

When the prompt, ENTER COMMAND, is again displayed on the terminal, the user may proceed immediately to update, by entering UPDAte, if he has a new data file or a subfile of the master file available for editing. If not, he may select a subfile of records to be edited by proceeding through the standard search procedure, specifying the sequence of commands CONDitions, LOGIc, and SEARch, each followed by the appropriate information as documented in the GRASP literature.

Assuming that the user either has a new data file or has generated a subfile of records which require updating, he will respond to the next ENTER COMMAND prompt with the entry of UPDAte. The message, PLEASE PRESS CARRIAGE RETURN, will then be displayed. If the user is familiar with the procedures and has knowledge of the code for suppressing the output of user instructions (see programmer's reference), he may enter it at this point before pressing the carriage return. If not, he will be presented with instructions for the updating procedure as well as special

TABLE 4.—*List of USALYT record variables*

```
LOCATION INFORMATION
GEOLOGIC DESIGNATIONS
BIBLIOGRAPHIC INFORMATION
RANK INFORMATION
ANALYSIS INFORMATION

STATE    A  STATE NAME
COUNTY   A  COUNTY NAME
PMERID   I  PRINCIPAL MERIDIAN
TWNSHIP  I  TOWNSHIP NUMBER
NS       A  DIRECTION OF TOWNSHIP (N OR S)
RANGE    I  RANGE NUMBER
EW       A  DIRECTION OF RANGE (E OR W)
SECTION  I  SECTION NUMBER
AAPGPRV  I  AAPG PROVINCE NUMBER

COALPRV  A  COAL PROVINCE NAME
REGION   A  COAL REGION NAME
FIELD    A  COAL FIELD NAME
DISTRCT  A  LOCAL AREA DESIGNATOR
FORMATN  A  FORMATION NAME
BED      A  COAL BED NAME
SYSTEM   A  GEOLOGIC AGE: SYSTEM
SERIES   A  GEOLOGIC AGE: SERIES

QUAD     A  TOPOGRAPHIC QUADRANGLE NAME
SOURCE   A  SOURCE DOCUMENT
YEAR     I  PUBLICATION YEAR OF SOURCE DOCUMENT

RANK     A  RANK OF COAL
                ANTH      = ANTHRACITE
                SEMI ANTH = SEMI-ANTHRACITE
                BIT       = BITUMINOUS
                LV BIT    = LOW-VOLATILE BITUMINOUS
                MV BIT    = MEDIUM-VOLATILE BITUMINOUS
                HV BIT    = HIGH-VOLATILE BITUMINOUS
                HV BIT A  = HIGH-VOLATILE BITUMINOUS A
                HV BIT B  = HIGH-VOLATILE BITUMINOUS B
                HV BIT C  = HIGH-VOLATILE BITUMINOUS C
                SUBBIT    = SUB-BITUMINOUS
                SUBBIT A  = SUB-BITUMINOUS A
                SUBBIT B  = SUB-BITUMINOUS B
                SUBBIT C  = SUB-BITUMINOUS C
                LIGNITE   = LIGNITE

ANIDA    A  REPORTED ANALYSIS IDENTIFICATION (ALPHABETIC)
ANIDN    I  REPORTED ANALYSIS IDENTIFICATION (NUMERIC)
SAMPTYP  I  SAMPLE TYPE
                1 = CHANNEL
                2 = RUN OF MINE
                3 = DRILL CORE
                4 = OTHER
ANLYTYP  I  ANALYSIS TYPE

                1 = AS RECEIVED
                2 = AIR DRIED
                3 = MOISTURE FREE
                4 = MOISTURE AND ASH FREE
                5 = OTHER
VALREP   I  VALUES REPRESENT
                1 = SINGLE SAMPLE
                2 = AVERAGE OF MORE THAN ONE SAMPLE
                3 = RANGE OF SAMPLE VALUES
                4 = OTHER
TRACE    A  SAMPLE ANALYZED FOR TRACE ELEMENTS(Y=YES N=NO)
HGRIND   I  HARDGROVE GRINDABILITY INDEX
OTHER    A  RESULTS OF OTHER TESTS SHOWN ON ANALYSIS
BTU      I  BTU VALUE
ASHDEF   I  ASH DEFORMATION TEMPERATURE IN FAHRENHEIT
ASHSOF   I  ASH SOFTENING TEMPERATURE IN FAHRENHEIT
ASHFLD   I  ASH FLUID TEMPERATURE IN FAHRENHEIT
FRESWEL  R  FREE-SWELLING INDEX
MOISTUR  R  MOISTURE VALUE IN PERCENT
VOLMAT   R  VOLATILE MATTER VALUE IN PERCENT
FIXEDC   R  FIXED CARBON VALUE IN PERCENT
ASH      R  ASH VALUE IN PERCENT
HYDROGN  R  HYDROGEN VALUE IN PERCENT
CARBON   R  CARBON VALUE IN PERCENT
NITROGN  R  NITROGEN VALUE IN PERCENT
OXYGEN   R  OXYGEN VALUE IN PERCENT
SULFUR   R  TOTAL SULFUR VALUE IN PERCENT
SULFATE  R  SULFATE VALUE IN PERCENT
SULFPYR  R  PYRITIC SULFUR IN PERCENT
SULFORG  R  ORGANIC SULFUR IN PERCENT
```

instructions for any of the five procedures he may select. If the user does not enter the instruction-suppression code, the following message will be displayed:

THIS IS "UPDATE"
IT IS DESIGNED TO PERMIT THE USER TO ADD RECORDS TO OR DELETE RECORDS FROM THE MASTER FILE, OR TO CHANGE RECORDS OR PORTIONS OF RECORDS IN A SUBFILE OF THE MASTER FILE AND TO POST THESE CHANGES ONTO THE MASTER FILE.

THE FIVE PROCEDURES USED TO UPDATE THE MASTER FILE ARE:

1. THE ADDITION OF NEW RECORDS ALREADY WRITTEN INTO A TEMPORARY FILE. UNLIKE THE OTHER UPDATE PROCEDURES WHICH OPERATE ON RECORDS IN THE MASTER FILE FORMAT, THIS PROCEDURE OPERATES ON RAW DATA RECORDS AND CONVERTS THESE RECORDS TO THE MASTER FILE FORMAT FOR INSERTION INTO THE MASTER FILE.
2. THE DELETION, BY KEY, OF RECORDS ALREADY EXISTING IN THE MASTER FILE.
3. THE SEQUENTIAL REVISION OF RECORDS FROM A SELECTED SUBFILE (I.E. SELECTED THROUGH A LOGICAL SEARCH) FROM THE MASTER FILE. THIS SUBFILE MAY THEN BE POSTED ONTO THE MASTER FILE AFTER THE DESIRED REVISONS ARE COMPLETED.
4. THE SELECTION, BY KEY, OF ANY RECORD BELONGING TO THE SUBFILE FOR REVISION OF ANY SELECTED DATA ELEMENT. THE DATA MANAGER HAS THE OPTION OF POSTING THE SELECTED RECORD ONTO THE MASTER FILE OR LEAVING IT, AS REVISED, IN THE SUBFILE FOR FURTHER REVISION.
5. THE BATCH REVISION OF A GIVEN DATA ELEMENT WHICH WILL BE THE SAME VALUE FOR ALL RECORDS IN THE SELECTED SUBFILE.

WHEN REVISION HAS BEEN COMPLETED ON ANY SELECTED SUBFILE, THE USER MAY THEN ELECT TO POST THE REVISED SUBFILE ONTO THE MASTER FILE, OR SAVE THE SUBFILE FOR REVIEW AND POSSIBLE FURTHER REVISION BY ANY OF THE 2-5 UPDATE PROCEDURES. RECORDS DELETED FROM THE MASTER FILE ARE SAVED IN THE "SAVDEL" (SAVE DELETION) FILE FOR FUTURE RECOVERY IF THERE SHOULD ARISE A NEED TO RECONSTITUTE THESE RECORDS.

An output display of each record reviewed is provided automatically for the deletion procedure, the sequential revision procedure, and the keyed revision procedure. The display of each record for correct data is optional when adding new records, but need not be used because dictionary nonmatches will be displayed to the data manager at the time of record entry. No display is associated with the batch revision because normally this revision will be made to a larger number of records, and such a display would inhibit the speed of the batch revision. If a display is desired following the batch revision, the data manager can choose to omit posting the revised subfile to the master file. He may then select the sequential or the keyed revision procedure to review the record and perhaps further update the subfile before posting it onto the master file.

The output display for the editing of records is formatted so that the data-element positions are similar to those positions in the SYCOR formats for data input, as discussed in the section on entry of new data. The example below is the formatted display which is used for the interactive edit and update procedures of PACER when used to operate on records from the WCOAL file:

```
* * * * * * * * * *
  STATE: WYOMING                COUNTY: CAMPBELL
 PMERID:  6 TWNSHIP: 58 NS: N  RANGE: 76 EW: W SECTION: 0  AAGPRV: 0

COALPRV: NO DATA ENTERED      REGION: NO DATA ENTERED      THICKNS: 4
  FIELD: SPOTTED HORSE       DISTRICT: NO DATA ENTERED     OVRBRDN: 3
FORMATN: FORT UNION             BED: CANYON                RELIABL: 2

 SYSTEM: NO DATA ENTERED       QUAD:                        BYEAR: **
 SERIES: NO DATA ENTERED      SOURCE: USGS BULL 1050   YEAR: 1957

   RANK: SUBBIT   ID: 56005005 . . . KEY: KEY:   13159   TONNAGE: 11.80

* * * * * * * * * *
```

The display of records from the USALYT file is shown in the following example:

```
* * * * * * * * * *
  STATE: NORTH DAKOTA          COUNTY: ADAMS
 PMERID:  5 TWNSHIP: 121 NS: N  RANGE: 95 EW: W SECTION:  10 AAPGPRV: 395

COALPRV: N GREAT PLAINS         REGION:  FORT UNION
  FIELD: NO DATA ENTERED      DISTRICT:  NO DATA ENTERED
FORMATN: FORT UNION              BED:  NO DATA ENTERED
 SYSTEM: TERTIARY                QUAD:  NO DATA ENTERED
 SERIES: PALEOCENE             SOURCE:  NDU DIV MIN CIRC 8 YEAR: 1934

   RANK: LIGNITE      ANID: NO.  10316     SAMPTYP: 1 ALYTYP: 1 VALREP: 1
  TRACE: N            HGRIND: 0           OTHER TESTS: N

    BTU: 6820  ASH: (DEFORM)  0  (SOFT)  0  (FLUID) 2280  FRESWEL: 0.

MOISTUR   VOLMAT    FIXEDC     ASH     HYDROGN   CARBON   NITROGN
   35.6     29.6      26.9       0           0        0         0

OXYGEN    SULFUR    SULFATE    SULFPYR    SULFORG    KEY:
     0.       2.7        0.         0.         0.     529

* * * * * * * * * *
```

Procedure 1 is selected for the addition of new records to the master file. Before this procedure can be used, a file of new data must be entered into the host computer system. Although there are many ways to create a file of new data, the method used predominantly will be to enter the data records onto a tape cassette through the SYCOR terminal and then to use the terminal to transmit these records to the host computer.

The file containing the new data is structured so that each record is an unformatted, but fixed length, string of characters (including blanks). This character string must be translated through a format to obtain the internal machine language values from the numerical data and to determine whether there is a dictionary match for the alphanumeric data. If a match is found, the numerical pointer to that dictionary entry is determined. If there is no dictionary entry match for the given item of alphanumeric data, the user is given the option of adding that data value to the dictionary or of correcting the data input (that is, a spelling error) so that it will match an existing dictionary entry.

After the user has completed the translation phase for the new data, he may elect either to post the translated file directly to the master file or to save it for further editing and revision by means of procedures 3 through 5 before posting it to the master file.

UPDATE PROCEDURE 1

Selection of update procedure 1 will produce the following message on the user terminal:

THE ADD RECORD PROCEDURE IS DESIGNED TO READ A RAW DATA INPUT FILE AND CONVERT THE RECORDS TO THE RECORD STRUCTURE THAT IS COMPATIBLE WITH THE MASTER FILE OF "PACER," CHECKING FOR CORRECT DICTIONARY ENTRIES, AND PROMPTING THE DATA MANAGER TO REQUEST ADDITION OF THE NONMATCHING DICTIONARY ENTRIES TO THE DICTIONARY LIST OR TO CORRECT THE INPUT ENTRY SO THAT IT MATCHES A VALUE ALREADY IN THE DICTIONARY LIST.
IF THE INPUT FILE DOES NOT CONTAIN RAW DATA, THE DATA MANAGER CAN EXIT THIS REVISION PROCEDURE TO SELECT A DIFFERENT PROCEDURE (3–5) BY ENTERING "QUIT," WHEN PROMPTED FOR A FILE NAME.

If the appropriate instruction-suppression code has previously been entered, this message will also be suppressed. Next, the user will be prompted with NAME OF RAW DATA INPUT FILE. The response to this prompt must be either the name of the newly created raw-data file or the word QUIT which will terminate execution of the current update procedure and will permit the user to

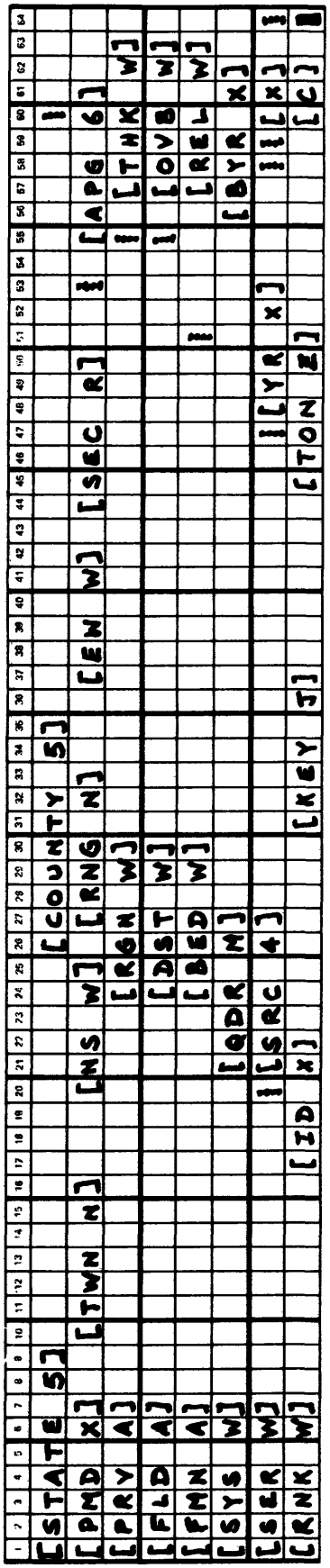


FIGURE 5.—Data-entry format for WCOAL.

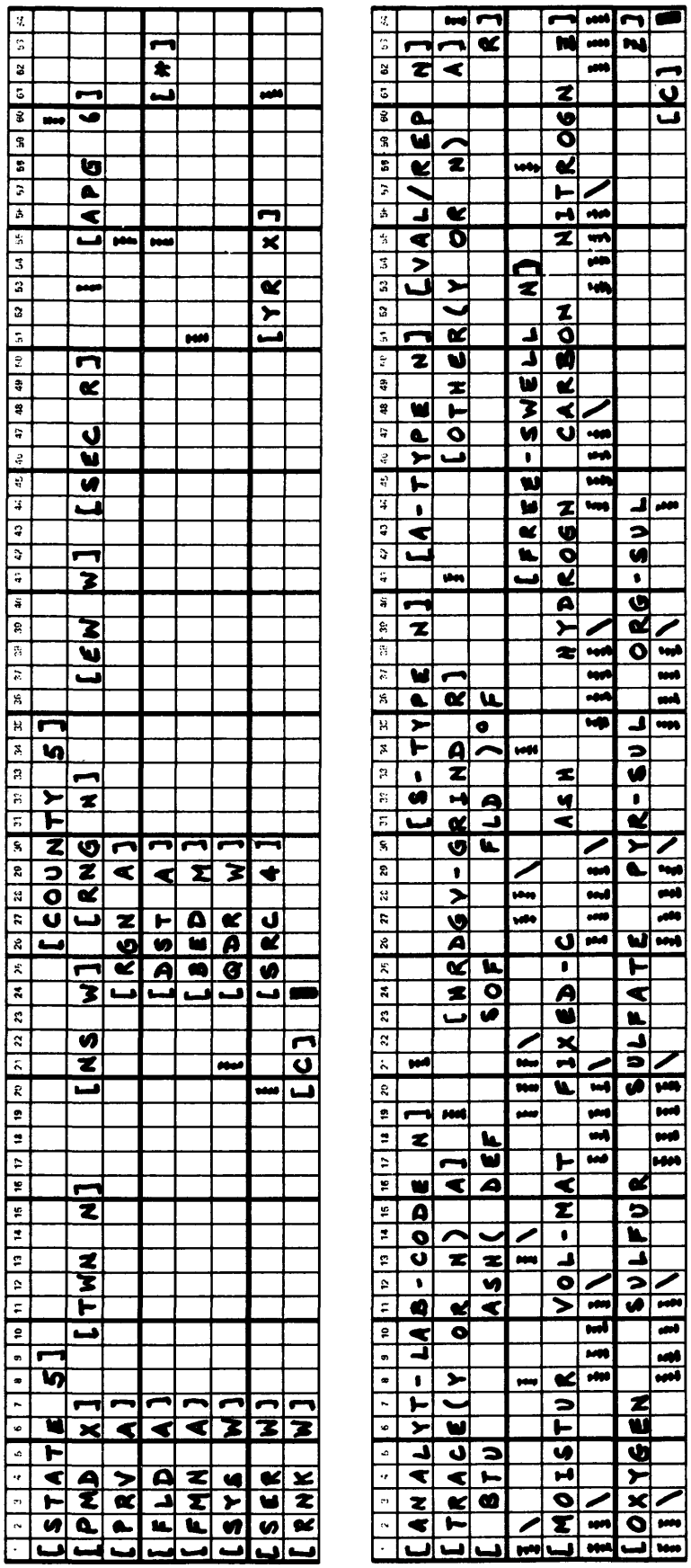


FIGURE 6.—Data-entry format for USALYT.

select an alternative procedure or return to the search and retrieval operation. If the name of the raw-data file is properly entered, the user will then be prompted for the NAME OF TRANSLATED FILE. This name is selected at the user's discretion, and care must be taken to ensure that it does not coincide with the name of another file already existing on that user account number.

Next, the user will be asked if he wishes to have the contents of every input record, as corrected for any dictionary mismatch, displayed on the terminal. A response of YES will display every record, and a response of NO will suppress the display. If an entry on the record does not match the existing entries in that dictionary, the message, THERE IS NO DICTIONARY MATCH FOR DATA NAME is displayed, followed by the name of that data element, the message ALPHANUMERIC DATA, and the displayed data value. The user is prompted with DO YOU WISH TO ENTER THIS DATA IN THE DICTIONARY? "YES" OR "NO." A YES will place that data value in its dictionary, and subsequent records containing that data value will obtain a dictionary match. If the response is NO, the user is prompted with the message, ENTER DATA VALUE FOR DATA NAME, in order to correct the data value, followed by the data name and the message, DATA VALUE. The newly entered data value is again tested for a dictionary match, and if found, the record translation procedure continues. If the new data value does not have a dictionary match, the user is again queried to determine whether he intends to enter that data value in the dictionary, or if he wishes to re-enter the data value. This cycle will continue until a dictionary entry exists that will match that data value.

When all the new data records have been translated, the following message will be displayed:

THE INPUT FILE IS NOW READY FOR POSTING ONTO THE MASTER FILE. IF YOU WISH TO MAKE FURTHER CHANGES, TO THIS FILE BEFORE IT IS POSTED, SELECT THE SEQUENTIAL, BATCH OR KEYED REVISION PROCEDURE AND SPECIFY THE NAME OF THIS FILE.

FILE "TEST" HAS BEEN REVISED.
DO YOU WISH TO WRITE THIS FILE ONTO THE MASTER FILE? "YES" OR "NO":

where TEST, in this case, is the name of the translated file. If the response is NO, the file is saved, and the prompt, UPDATE PROCEDURE (1–5), is again displayed. If the response is YES, the translated file is added to the master file and the user is queried with, DO YOU WISH TO SAVE THIS FILE? "YES" OR "NO." A YES will save the translated file, where a NO will cause the file to be dropped. Finally, the message:

UPDATE OPERATIONS HAVE BEEN COMPLETED. IF YOU WISH TO CONTINUE WITH THE UPDATE PROCEDURE, ENTER THE NUMBER (1–5). ENTERING A "0" FOR THE PROCEDURE PROMPT OR "QUIT" FOR THE FILE PROMPT WILL RETURN CONTROL TO THE SEARCH AND RETRIEVAL PORTION OF "PACER."

is displayed followed by the prompt, UPDATE PROCEDURE (1–5), to permit further update operations (1–5), or to return to search and retrieval (0).

UPDATE PROCEDURE 2

Update procedure 2 is the deletion procedure, which is accomplished by specifying the record key of records from the master file. The prompt, DATA FILE TO BE REVISED, will be for a file that has already been selected by a prior logical search for records that are to be deleted from the master file. If the instruction-suppression code has not been set previously, the following message will be displayed on the terminal:

THIS IS THE MASTER FILE RECORD DELETION PROCEDURE. WHEN PROMPTED, THE DATA MANAGER WILL SPECIFY THE KEY NUMBER OF THE RECORD TO BE DELETED. THE DELETED RECORD WILL BE WRITTEN, FOR PRESERVATION, ONTO THE "ESAVE," "WSAVE," OR "SAVUSA" FILE. THEN, THE DATA ELEMENTS IN THE MASTER FILE WILL BE BLANKED, AND A NEW IDENTIFICATION NUMBER WILL BE WRITTEN ONTO THE MASTER FILE. THIS NUMBER WILL BE ENCODED TO CONTAIN THE DATE OF DELETION AND THE IDENTIFICATION NUMBER OF THE DATA MANAGER RESPONSIBLE FOR EXECUTING THE DELETION. IN ADDITION, A DELETION MESSAGE WILL BE SUPERIMPOSED OVER SEVERAL OF THE DATA FIELDS TO NOTE TO THE USER THAT THAT KEY NUMBER NO LONGER HAS A VALID RECORD IN THE MASTER FILE.

WHEN THE DATA MANAGER HAS CONCLUDED THE DELETION PROCEDURE,

ENTRY OF A "–1" AT THE PROMPT FOR "KEY" WILL END THE PROCESS.

It is presumed that the user has determined, by examination of the selected file, which record keys will be in the deletion process. He will enter the key number following the prompt, KEY NUMBER OF RECORD TO BE DELETED. This will cause the data elements to be blanked out in that record of the master file having the specified key number. A deletion message is inserted in the REGION, DISTRCT, and BED data fields along with a coded ID number giving the year, month, and day of deletion as well as the data manager identification, as in the following example:

```
* * * * * * * * * * *
   STATE:                          COUNTY:
  PMERID: **  TWNSHIP: *** NS:    RANGE: *** EW:  SECTION: ** AAPGRV: ***

 COALPRV:          REGION: ** RECORD DELETED. . . ** THICKNS  : *
   FIELD:        DISTRICT: ** SEE "ID" FOR DATE   ** OVRBRDN : *
FORMATN:             BED: ** AND MANAGER CODE.    RELIABL  : *

  SYSTEM:            QUAD:                          BYEAR    : **
  SERIES:          SOURCE:                    YEAR: ***

    RANK:     ID: 75090299...KEY: 13158      TONNAGE: .00
* * * * * * * * * * *
```

For example, the date of deletion is 75 (year), 09 (month), and 02 (date), and 99 is the data manager identification. The corresponding record in the selected subfile is eliminated entirely. This prevents that record from being posted onto the master file again after the subfile has been subsequently subjected to any of the other (3–5) updating procedures. To terminate the deletion procedure, a negative entry, such as –1, following the prompt, KEY NUMBER OF RECORD TO BE DELETED, will permit the data manager either to select another update procedure or to return to the search and retrieval activity of PACER.

UPDATE PROCEDURE 3

Update procedure 3 is designed for the sequential review and revision of records belonging to a subfile selected from the master file. After the sequential revision has been completed, the data manager may elect to post the subfile back onto the master file or to save the subfile for further update operations.

If the user has not previously entered the instruction-suppression code, the first message displayed will be the following instructions:

RECORDS FROM THE DESIGNATED SUBFILE WILL BE PRESENTED SEQUENTIALLY FOR REVIEW AND UPDATE. AFTER ALL RECORDS HAVE BEEN EXAMINED BY THE REVIEWER, HE MAY THEN ELECT TO POST THE RECORDS IN THIS SUBFILE ONTO THE MASTER FILE.

WHENEVER YOU WISH TO LEAVE A SELECTED DATA ELEMENT UNCHANGED, ENTER AN ASTERISK, *, FOLLOWED BY A CARRIAGE RETURN.

IF YOU WISH TO PROCEED TO THE NEXT RECORD IN THE FILE, ENTER THE CHARACTERS "NEXT" FOLLOWING THE PROMPT: "NAME OF DATA ELEMENT TO BE CHANGED." THE "NEXT" COMMAND WILL LEAVE THAT RECORD IN ITS ORIGINAL, UNREVISED STATE AND THE NEXT RECORD WILL BE DISPLAYED, IN SEQUENCE, FROM THE SUBFILE. IF AT ANY TIME YOU DO NOT WISH TO REVIEW THE REMAINDER OF THE FILE, ENTER "QUIT."

Following this message will be a display of the first record in the subfile, as shown in an earlier example. This display will be followed by the prompt, NAME OF DATA ELEMENT TO BE CHANGED. The data manager may either enter a valid data-element name or specify the command NEXT or QUIT. If a proper data-element name has been entered, the next prompt will be either, ENTER VALUE, if it is a whole number, ENTER DECIMAL VALUE, if it is decimal data, or ENTER DATA, if the value is an alphanumeric value. At this point, the data value may be left unchanged by entering an asterisk, *, for the alphanumeric data, or an asterisk enclosed by apostrophes, '*', for the integer and decimal data. If the value is alphanumeric, a search is made to determine whether the data entry matches an already existing dictionary entry. If there is a matching dictionary entry, or if the proper type of numerical data has been entered, the data manager will be prompted with,

DO YOU WISH TO CHANGE ANY MORE DATA ELEMENTS BELONGING TO THIS RECORD? ENTER "YES" OR "NO." A response of NO will cause all preceding changes to that record to be posted back onto the subfile. A response of YES will again display the prompt, NAME OF DATA ELEMENT TO BE CHANGED. The user can then select another data element to revise, or he may enter NEXT or QUIT. The NEXT command nullifies all data changes made to that record and proceeds to display the next record from the subfile. The QUIT command presents the user with the option of posting the subfile onto the master file or leaving the subfile available for further activity.

If the alphanumeric input does not match a dictionary entry, the message, THERE IS NO DICTIONARY MATCH FOR DATA NAME, is displayed, followed by the name of the data element, and the prompt, DO YOU WISH TO ENTER THIS DATA IN THE DICTIONARY? "YES" OR "NO." An entry of YES will make that alphanumeric value a permanent dictionary entry and then query for further data changes to the record. A NO entry will bring up the prompt, NAME OF DATA ELEMENT TO BE CHANGED, giving the data manager the opportunity to re-enter the correct alphanumeric value for that data element, or to select a different data-element name for data revision.

After all records in the subfile have been reviewed or after QUIT has been entered in response to the prompt, NAME OF DATA ELEMENT TO BE CHANGED, the prompt, FILE "(subfile name)" HAS BEEN REVISED. DO YOU WISH TO WRITE THIS FILE ONTO THE MASTER FILE? "YES" OR "NO" will appear. A response of NO returns control for selection of another update procedure. A response of YES posts the subfile onto the master file, following with the prompt, DO YOU WISH TO SAVE THIS FILE? "YES" or "NO." If NO, the subfile is dropped; if YES, the subfile is saved for further use. In either case, the message:

UPDATE OPERATIONS HAVE BEEN COMPLETED.
IF YOU WISH TO CONTINUE WITH THE UPDATE PROCEDURE ENTER THE NUMBER (1–5). ENTERING A "0" FOR THE PROCEDURE PROMPT OR "QUIT" FOR THE FILE PROMPT WILL RETURN CONTROL TO THE SEARCH AND RETRIEVAL PORTION OF "PACER."

is printed followed by the prompt, UPDATE PROCEDURE (1–5), for selection of further update operations.

UPDATE PROCEDURE 4

Update procedure 4 is designed to permit the user to access a given record from a previously selected subfile by means of the record key number. The data contained in that record is displayed in the same format as illustrated earlier. If the instruction-suppression code has not been set, the following instructions will precede the first prompt, KEY.

RECORDS FROM THE DESIGNATED SUBFILE WILL BE PRESENTED, AS SPECIFIED BY KEY NUMBER, FOR REVIEW AND UPDATE. AFTER THE REVIEWER HAS EXAMINED THE CONTENTS OF THE RECORD OF INTEREST, HE MAY ELECT TO POST THAT RECORD ONTO THE MASTER FILE.
REGARDLESS OF WHETHER OR NOT THE RECORD IS POSTED TO THE MASTER FILE, IT WILL REMAIN, AS REVISED, IN THE SUBFILE.
TO ACCESS THE DESIRED RECORD, RESPOND TO THE PROMPT "KEY" BY ENTERING THE RECORD'S KEY NUMBER. . . . ENTERING A "–1" WILL CONCLUDE THE KEYED ACCESS PROCEDURE.
IF YOU WISH TO GO ON TO ANOTHER RECORD IN THE FILE, ENTER THE CHARACTERS "NEXT" FOLLOWING THE PROMPT: "NAME OF DATA ELEMENT TO BE CHANGED." THE "NEXT" COMMAND WILL LEAVE THAT RECORD IN ITS ORIGINAL, UNREVISED STATE AND PROMPT FOR THE NEXT RECORD KEY.
WHENEVER YOU WISH TO LEAVE A SELECTED DATA ELEMENT UNCHANGED, ENTER AN ASTERISK, *.

After the key number has been specified for the KEY prompt, the selected record is then displayed, followed by the prompt, NAME OF DATA ELEMENT TO BE CHANGED. The data modification procedure for this record is identical with the sequential revision procedure (update procedure number 3). The NEXT and the QUIT commands are also used in the same context as in the sequential revision procedure. The asterisk response again is used to leave the data fields unchanged.

After revision of the record has been completed, the prompt, DO YOU WISH TO POST THIS RECORD TO THE MASTER FILE? "YES" OR "NO," is displayed. If the response is YES, it is posted to the master file as well as to the subfile. If the response is NO, it is posted to the subfile only. After

the keyed revision is concluded, the data manager can, by means of procedure 3 or 5, make further corrections to the subfile and then post the entire subfile onto the master file.

The next prompt after the one for posting the record onto the master file is, DO YOU WANT TO REVIEW ANY MORE RECORDS? "YES" OR "NO." A response of YES will produce the prompt, KEY, whereas a response of NO concludes the keyed update procedure and returns the prompt, UPDATE PROCEDURE (1–5). Entry of a negative integer value following the KEY prompt will also conclude the keyed update procedure.

UPDATE PROCEDURE 5

Update procedure 5 is the batch revision procedure. It is used to change all subfile records to the same specified data value for a selected data element. If the instruction-suppression code has not been set, the following message will be displayed:

THIS IS THE BATCH UPDATE PROCEDURE.
GIVEN A SPECIFIED DATA ELEMENT NAME AND A SPECIFIED DATA ELEMENT VALUE, THIS PROCEDURE CHANGES ALL RECORDS IN THE GIVEN SUBFILE TO THE DATA VALUE SPECIFIED FOR THAT DATA ELEMENT NAME.
THE BATCH EDIT/REVISION PROCEDURE CAN BE TERMINATED BY ENTERING "QUIT" WHEN A PROMPT FOR THE NAME OF THE DATA ELEMENT IS ENCOUNTERED.

This message will be followed by the prompt, NAME OF DATA ELEMENT TO BE CHANGED. The user will respond to this prompt and the ones that follow for the data values in the same manner as when using update procedures 3 and 4.

Because of the mass record revision capability of this updating procedure, the batch procedure does not produce a display of any of the records changed. If it is desired to review any records in this subfile before posting them onto the master file, the sequential or the keyed revision procedure may be selected after the batch revision process has been concluded. To review records with the sequential revision procedure, enter NEXT following the prompt, NAME OF DATA ELEMENT TO BE CHANGED, and the next record from the subfile will be displayed. Entering QUIT will conclude the review. If the keyed revision procedure is selected for review, enter the record key number following the prompt, KEY. If the key number is unknown, enter the value 1. The keyed read accesses the first record with a key greater than or equal to 1. To access the next record in the subfile enter NEXT following the prompt, NAME OF DATA ELEMENT TO BE CHANGED, and then a key value exceeding by one the key value of the previous record. This will access the record in the subfile with the next higher key. A continuation of this process will have the same effect as a sequential review. The user can terminate this procedure at any time, by entering a negative key value for the key prompt, or QUIT in place of NEXT for the data-element name.

After the update and posting of a subfile onto the master file, the data manager can conclude the update process by entering a 0 (zero) when prompted with UPDATE PROCEDURE (1–5). At this point, the prompt, ENTER COMMAND, will be displayed, and the user may respond with any one of the standard PACER commands (see table 1). It is also possible at this point to select another subfile to be used for the editing and correcting of the master file.

If several corrected subfiles have been saved for future posting to the master file, care must be taken to ensure that if any of the subfiles contain overlapping records, they will be posted in proper sequence to avoid negating previously posted record corrections.

USE OF TABULAR SUMMARY OPTION

Data may be retrieved from either USALYT or WCOAL in a predefined tabular output form on a *wide carriage* terminal (135 characters per line, or more) by entering the command TABLE. A data file retrieved from WCOAL will be listed showing tonnage of coal in various thickness categories by rank, coal bed, and overburden. Because the width of paper is a limiting factor and because of the natural break in thickness categories for varying ranks of coal, two summary tables are automatically printed, if required by the data, for thickness categories given in inches or feet.

The user is prompted to enter the data file to be printed and a description of the area searched to be printed as a title. As each data record is read in, it is checked for THICKNS equal to "unclassified" or "classified by zone," or RELIABL equal to "unclassified." If any of these conditions are true, the tonnage is added to the appropriate one of three subtotals, and the record is skipped. The other data records are then written to one of two files, depending upon whether THICKNS is in feet or inches.

After all records are read, the inches file is sorted by county, rank, coal bed, overburden, thickness, and reliability. A table is printed in the form shown in figure 8, new lines of subtotals being printed at every change in overburden, bed, rank, or county. A line showing the totals for all thickness columns is printed at the end of the table. If no data are reported in inches, or after the first end of file, the program reads, sorts, and prints the table for THICKNS reported in feet, also shown in figure 8.

Upon reaching an end of file for the second, or for the only data file, a total line is printed for all tables shown. If any input records have been skipped because of THICKNS or RELIABL conditions, a message is printed showing the total tonnage omitted, followed by the tonnage of coal excluded by each THICKNS or RELIABL condition. A final total tonnage for the area under consideration is then printed.

When a file of analytical data has been retrieved from USALYT, entering the command TABLE from a *wide carriage* terminal (135 characters per line, or more) will produce a tabular output as shown in figure 9. The user is prompted to enter the name of the data file to be printed and to enter a description of the area searched, which is to be printed in the title. The input data file is sorted by county, rank, and coal bed. Data records are read and a subtotal computed until a change in coal bed, rank, or county occurs. A separate counter is kept for each variable because a number of incomplete records are anticipated. The data are then averaged and printed according to the format shown in figure 8.

EXAMPLE OF INTERACTIVE SESSION

The questions and requests in figures 7, 8, and 9 provide examples of actual use of the USGS National Coal Resources Data System. Inputs required and prompts and responses by the PACER system utilized in Phase I studies are illustrated.

Questions:

1. What is the total tonnage of bituminous and subbituminous coal in T5–6N, R89–90W of the Yampa coal field in Colorado?
2. What coal chemical analyses are available in this area, and what are the average sulfur, ash, and Btu values?

Requests:

1. Tabulate the tonnage of coal in this area by county, rank, coal bed, and thickness of coal bed and overburden.
2. Tabulate the chemical analyses of the coal in this area.

PROGRAMMER'S REFERENCE

This section includes two topics:

1. Modifications to the GRASP system which are required in order to provide updating and editing capabilities in PACER;
2. New programs which constitute the updating and editing capabilities of PACER, called by the UPDATE command.

The user should note that PACER is presently operational on the Computer Science Corp. (INFONET) timesharing UNIVAC 1108 system in El Segundo, Calif. Some INFONET-dependent programs have been used with the PACER subroutines, and are, therefore, not transferrable to other computers. Similar or equivalent programs must be available or written for PACER to operate in its present form. The INFONET-dependent programs are noted at the end of discussion for each PACER subroutine in appendix C.

MODIFICATIONS TO GRASP

As discussed earlier, PACER is a modified version of the GRASP search and retrieval program. The modifications to the search and retrieval system are minor and involve the tailoring of GETREC, the input and output subroutine, to obtain greater operating efficiency with the particular file structures of WCOAL, ECOAL, and USALYT.

Because of the needed capability for editing and updating these three master files, they were created as keyed record files. This permits the random access of records for editing and updating. In addition, very few of the data fields in an NCRDS master record contain blank information; instead, they contain integer data or integer pointers for alphanumeric data contained in the dictionaries. Little is to be gained in terms of storage by packing these records. However, a significant proportion of CPU time is saved by storing them in the unpacked mode. Thus, the PACER version of GRASP is structured to handle keyed and unpacked master file records, but all other major features of GRASP are retained. The GRASP documentation, therefore, can be utilized as a user's guide for the search and retrieval operation of PACER.

The principal distinction between PACER and GRASP is the addition of considerable software programming to permit user updating and editing of

```
!PACER
WELCOME TO THE USGS "PACER" VERSION OF THE "GRASP" RETRIEVAL SYSTEM.
THE FOLLOWING DATA BASES ARE AVAILABLE:

ECOAL  - USGS EASTERN US COAL RESOURCES DATA

WCOAL  - USGS WESTERN US COAL RESOURCES DATA

USALYT  - PUBLISHED US COAL ANALYTICAL DATA

THE USER MUST SPECIFY ONE OF THE ABOVE DATA BASES FOLLOWING THE PROMPT.

A CHANGE OF DATA BASES CAN BE MADE AT ANY TIME BY ENTERING THE WORD "FILE"
FOLLOWING THE PROMPT TO "ENTER COMMAND."

ENTER DATA BASE NAME: WCOAL



ENTER COMMAND:  COND
A.  STATE EQ COLORADO
B.  FIELD EQ YAMPA
C.  TWNSHIP BE 5,6
D.  NS EQ N
E.  RANGE BE 89,90
F.  EW EQ W
G.


ENTER COMMAND:  LOGIC
ENTER LOGIC:   A*B*C*D*E*F


ENTER COMMAND:  SEARCH
ENTER INPUT FILE NAME:  WCOAL
ENTER OUTPUT FILE NAME:  RYAMPA
ALL  15972 RECORDS OF WCOAL  SEARCHED.
       125 RECORDS FOUND WHICH SATISFY THE REQUEST.
THEY HAVE BEEN STORED IN RYAMPA
SRU'S:  140.6


ENTER COMMAND:  FUNC
ENTER NAME OF FILE:  RYAMPA


FUNCTIONS AVAILABLE AT THIS TIME ARE:
    MEAN      FIT
```

FIGURE 7.—Example of an interactive session using PACER.

```
ENTER FUNCTION NAMES AND CORRESPONDING ARGUMENTS.
1.  MEAN TONNAGE
2.

MEAN STATISTICS FOR TONNAGE WITH   125  ITEM(S).
 MIN=       .24  MAX=    570.34  MEAN=     31.24  ROOT MEAN SQ.=6.40E+03
 SUM=     3904.83         SUM OF SQUARES=   7.996+005
 STD. DEV.=   73.923       VARIANCE=   5464.6


ENTER COMMAND: COND
A.  RANK BE BIT, HV BIT C
B.  RANK BE SUBBIT, SUBBIT C
C.



ENTER COMMAND:  LOGIC
ENTER LOGIC:    A


ENTER COMMAND:  SEARCH
ENTER INPUT FILE NAME:  RYAMPA
ENTER OUTPUT FILE NAME:  BITYAM
ALL   125 RECORDS OF RYAMPA SEARCHED.
       77 RECORDS FOUND WHICH SATISFY THE REQUEST.
THEY HAVE BEEN STORED IN BITYAM
SRU'S:1.4


ENTER COMMAND:  FUNC
ENTER NAME OF FILE:  BITYAM

FUNCTIONS AVAILABLE AT THIS TIME ARE:
    MEAN    FIT

ENTER FUNCTION NAMES AND CORRESPONDING ARGUMENTS.
   1.  MEAN TONNAGE
   2.

MEAN STATISTICS FOR TONNAGE WITH    77 ITEMS(S).
 MIN=       .30  MAX=    570.34  MEAN=     35.31  ROOT MEAN SQ.= 7.48E+03
 SUM=     2718.72         SUM OF SQUARES=   5.762+005
 STD.  DEV.=   79.490       VARIANCE=   6318.7


ENTER COMMAND:  LOGIC
ENTER LOGIC:    B
```

FIGURE 7.—Example of an interactive session using PACER—Continued

```
ENTER COMMAND:  SEARCH
ENTER INPUT FILE NAME:  RYAMPA
ENTER OUTPUT FILE NAME:  SUBYAM
ALL   125 RECORDS OF RYAMPA SEARCHED.
       48 RECORDS FOUND WHICH SATISFY THE REQUEST.
THEY HAVE BEEN STORED IN SUBYAM

SRU'S:1.3



ENTER COMMAND:  FUNC
ENTER NAME OF FILE:  SUBYAM


FUNCTIONS AVAILABLE AT THIS TIME ARE:
    MEAN    FIT


ENTER FUNCTION NAMES AND CORRESPONDING ARGUMENTS.
   1.  MEAN TONNAGE
   2.

MEAN STATISTICS FOR TONNAGE WITH    48 ITEM(S).
 MIN=       .24  MAX=   448.49  MEAN=    24.71  ROOT MEAN SQ.= 4.65E+03
 SUM=     1186.11         SUM OF SQUARES=   2.234+005
 STD.  DEV.=   64.257       VARIANCE=   4129.0


SRU'S:7.1



ENTER COMMAND:  FILE
ENTER DATA BASE NAME:  USALYT


ENTER COMMAND:  COND
A.   STATE EQ COLORADO
B.   FIELD EQ YAMPA
C.   TWNSHIP BE 5,6
D.   NS EQ N
E.   RANGE BE 89,90
F.   EW EQ W
G.



ENTER COMMAND:  LOGIC
ENTER LOGIC:    A*B*C*D*E*F



ENTER COMMAND:  SEARCH
ENTER INPUT FILE NAME:  USALYT
ENTER OUTPUT FILE NAME:  AYAMPA
ALL   666 RECORDS OF USALYT SEARCHED.
       62 RECORDS FOUND WHICH SATISFY THE REQUEST.
THEY HAVE BEEN STORED IN AYAMPA
```

FIGURE 7.—Example of an interactive session using PAWER—Cont.

```
SRU'S:7.1


ENTER COMMAND:  COND
A.   RANK BE BIT,HV BIT C
B.   RANK BE SUBBIT,SUBBIT C
C.


ENTER COMMAND:  LOGIC
ENTER LOGIC:    A


ENTER COMMAND:  SEARCH
ENTER INPUT FILE NAME:  AYAMPA
ENTER OUTPUT FILE NAME:  ABTYAM
ALL    62 RECORDS OF AYAMPA SEARCHED.
       50 RECORDS FOUND WHICH SATISFY THE REQUEST.
THEY HAVE BEEN STORED IN ABTYAM

SRU'S:1.2


ENTER COMMAND:  FUNC
ENTER NAME OF FILE:  ABTYAM


FUNCTIONS AVAILABLE AT THIS TIME ARE:
    MEAN    FIT


ENTER FUNCTION NAMES AND CORRESPONDING ARGUMENTS.
   1.  MEAN SULFUR, ASH, BTU
   2.

MEAN STATISTICS FOR SULFUR WITH    50 ITEM(S).
 MIN=       .30  MAX=      3.60  MEAN=       .71  ROOT MEAN SQ.=   .80
 SUM=       35.50         SUM OF SQUARES=    39.9
 STD.  DEV.=    .54782      VARIANCE=    .30010

MEAN STATISTICS FOR ASH WITH    50 ITEM(S).
 MIN=      2.70  MAX=     26.90  MEAN=      8.11  ROOT MEAN SQ.=  85.
 SUM=       405.60         SUM OF SQUARES=   4.257+003
 STD.  DEV.=    4.4410      VARIANCE=    19.722

MEAN STATISTICS FOR BTU      WITH    50 ITEM(S).
 MIN=  8390.00  MAX= 11920.00  MEAN= 11080.20  ROOT MEAN SQ.=1.23E+08
 SUM=   554010.00         SUM OF SQUARES=   6.156+009
 STD.  DEV.=    603.16      VARIANCE=  3.63798E+05


ENTER COMMAND:  LOGIC
ENTER LOGIC:    B
```

FIGURE 7.—Example of an interactive session using PACER—Continued

```
ENTER COMMAND:  SEARCH
ENTER INPUT FILE NAME:  AYAMPA
ENTER OUTPUT FILE NAME:  ASBYAM
ALL    62 RECORDS OF AYAMPA SEARCHED.
       12 RECORDS FOUND WHICH SATISFY THE REQUEST.
THEY HAVE BEEN STORED IN ASBYAM

SRU'S:.7


ENTER COMMAND:  FUNC
ENTER NAME OF FILE:  ASBYAM


FUNCTIONS AVAILABLE AT THIS TIME ARE:
    MEAN    FIT


ENTER FUNCTION NAMES AND CORRESPONDING ARGUMENTS.
   1.  MEAN SULFUR, ASH, BTU
   2.

MEAN STATISTICS FOR SULFUR  WITH    12 ITEM(S).
 MIN=        .30  MAX=      1.20  MEAN=       .63  ROOT MEAN SQ.=   .46
 SUM=        7.60         SUM OF SQUARES=      5.58
 STD.  DEV.=   .26400       VARIANCE=  6.96970E-02

MEAN STATISTICS FOR ASH WITH    12 ITEM(S).
 MIN=      3.20  MAX=      7.50  MEAN=      4.64  ROOT MEAN SQ.=  23.
 SUM=       55.70         SUM OF SQUARES=     273.
 STD.  DEV.=   1.1357       VARIANCE=    1.2899

MEAN STATISTICS FOR BTU     WITH    12 ITEM(S).
 MIN=  9730.00  MAX=  11290.00  MEAN= 10564.17  ROOT MEAN SQ.= 1.12E+08
 SUM=   126770.00         SUM OF SQUARES=   1.341+009
 STD.  DEV.=   433.75       VARIANCE=  1.88136E+05
```

FIGURE 7.—Example of an interactive session using PACER—Continued

```
ENTER COMMAND:  TABLE
ENTER UP TO 6 CHARACTER NAME OF DATA FILE TO BE PRINTED: RESYAM
ENTER NAME OF AREA SEARCHED (UP TO 28 CHARACTERS): T5-6N,R89-90W,YAMPA FIELD,CO
```

SUMMARY OF COAL RESOURCES FOR AREA = T5-6N,R89-90W,YAMPA FIELD,CO

(IN MILLIONS OF SHORT TONS)

RANK / COAL BED / OVERBURDEN	MEASURED AND INDICATED (INCHES) 14 - 28	28 - 42	OVER 42	TOTAL	INFERRED RESOURCES (INCHES) 14 - 28	28 - 42	OVER 42	TOTAL	TOTAL RESOURCES, BY CATEGORY 14 - 28	28 - 42	OVER 42	TOTAL
				MOFFAT COUNTY								
BIT												
A												
0-1000	.00	5.88	.00	5.88	.00	.00	.00	.00	.00	5.88	.00	5.88
B												
0-1000	.00	.00	.56	.56	.00	.00	.00	.00	.00	.00	.56	.56
BD GROUP												
0-1000	1.37	6.34	25.43	33.14	.00	.00	.00	.00	1.37	6.34	25.43	33.14
1-2000	.00	.00	4.18	4.18	.00	.00	.00	.00	.00	.00	4.18	4.18
D												
0-1000	3.57	1.79	13.10	18.46	.00	.00	.00	.00	3.57	1.79	13.10	18.46
1-2000	.00	1.37	17.56	18.93	.00	.00	.00	.00	.00	1.37	17.56	18.93
E												
0-1000	.00	.00	8.55	8.55	.00	.00	.00	.00	.00	.00	8.55	8.55
F ZONE												
0-1000	18.05	.00	.00	18.05	.00	.00	.00	.00	18.05	.00	.00	18.05
1-2000	19.56	.00	.00	19.56	.00	.00	.00	.00	19.56	.00	.00	19.56
FF GROUP												
0-1000	4.62	14.94	80.10	99.66	.00	.00	.00	.00	4.62	14.94	80.10	99.66
1-2000	.30	1.50	28.16	29.96	.00	.00	.00	.00	.30	1.50	28.16	29.96
G ZONE												
0-1000	.94	1.34	67.84	70.12	.00	.00	.00	.00	.94	1.34	67.84	70.12
1-2000	3.36	.00	17.90	21.26	.00	.00	.00	.00	3.36	.00	17.90	21.26
H ZONE												
0-1000	3.02	6.47	75.41	84.90	.00	.00	.00	.00	3.02	6.47	75.41	84.90
1-2000	1.62	9.40	90.88	101.90	.00	.00	.00	.00	1.62	9.40	90.88	101.90
J												
0-1000	.00	3.70	6.16	9.86	.00	.00	.00	.00	.00	3.70	6.16	9.86
1-2000	.00	8.34	.00	8.34	.00	.00	.00	.00	.00	8.34	.00	8.34
LOWER GR												
0-1000	6.40	9.79	.00	16.19	.00	.00	.00	.00	6.40	9.79	.00	16.19
WADGE												
1-2000	.00	.00	.00	.00	.00	.00	77.71	77.71	.00	.00	77.71	77.71
COUNTY TOTAL =	62.81	70.86	435.83	569.50	.00	.00	77.71	77.71	62.81	70.86	513.54	647.21
				ROUTT COUNTY								
BIT												
A												
0-1000	.00	.00	4.25	4.25	.00	.00	.00	.00	.00	.00	4.25	4.25

B													
	0-1000	1.23	.00	1.60	2.83	.00	.00	.00	.00	1.23	.00	1.60	2.83
BELOW D													
	1-2000	.00	.00	2.42	2.42	.00	.00	.00	.00	.00	.00	2.42	2.42
C													
	0-1000	.75	.00	14.65	15.40	.00	.00	.00	.00	.75	.00	14.65	15.40
	1-2000	.00	1.38	12.05	13.43	.00	.00	.00	.00	.00	1.38	12.05	13.43
D													
	0-1000	.00	.83	18.17	19.00	.00	.00	.00	.00	.00	.83	18.17	19.00
	1-2000	.00	.00	9.14	9.14	.00	.00	.00	.00	.00	.00	9.14	9.14
E													
	0-1000	.00	.00	2.74	2.74	.00	.00	.00	.00	.00	.00	2.74	2.74
F ZONE													
	0-1000	5.00	7.20	57.54	69.74	.00	.74	4.68	5.42	5.00	7.94	62.22	75.16
	1-2000	7.00	5.84	36.74	49.58	.00	12.13	50.09	62.22	7.00	17.97	86.83	111.80
G ZONE													
	0-1000	12.16	22.20	158.04	192.40	.00	.00	.00	.00	12.16	22.20	158.04	192.40
	1-2000	13.96	30.45	139.39	183.80	.00	.00	251.05	251.05	13.96	30.45	390.44	434.85
H ZONE													
	0-1000	21.74	.00	148.72	170.46	.00	.00	.00	.00	21.74	.00	148.72	170.46
	1-2000	.00	.00	100.30	100.30	.00	.00	.00	.00	.00	.00	100.30	100.30
J													
	0-1000	.00	.00	71.05	71.05	.00	.00	.00	.00	.00	.00	71.05	71.05
WADGE													
	1-2000	.00	.00	.00	.00	.00	.00	21.30	21.30	.00	.00	21.30	21.30
COUNTY TOTAL =		61.84	67.90	776.80	906.54	.00	12.87	327.12	339.99	61.84	80.77	1103.92	1246.53
TOTAL =		124.65	138.76	1212.63	1476.04	.00	12.87	404.83	417.70	124.65	151.63	1617.46	1893.74

SUMMARY OF COAL RESOURCES FOR AREA = T5-6N,R89-90W,YAMPA FIELD,CO

(IN MILLIONS OF SHORT TONS)

RANK		MEASURED AND INDICATED (FEET)				INFERRED RESOURCES (FEET)				TOTAL RESOURCES, BY CATEGORY			
COAL BED	OVERBURDEN	2.5 - 5	5 - 10	OVER 10	TOTAL	2.5 - 5	5 - 10	OVER 10	TOTAL	2.5 - 5	5 - 10	OVER 10	TOTAL
					MOFFAT	COUNTY							
SUBBIT													
K													
	0-1000	.00	57.07	.00	57.07	.00	.00	.00	.00	.00	57.07	.00	57.07
N													
	0-1000	.00	38.67	.00	38.67	.00	.00	.00	.00	.00	38.67	.00	38.67
Q													
	0-1000	35.43	4.85	.00	40.28	.00	.00	.00	.00	35.43	4.85	.00	40.28
	1-2000	9.00	.00	.00	9.00	.00	.00	.00	.00	9.00	.00	.00	9.00
R													
	0-1000	.00	2.72	.00	2.72	.00	.00	.00	.00	.00	2.72	.00	2.72

FIGURE 8.—Example of a tabular summary of coal resources.

S													
	0-1000	14.33	.00	.00	14.33	.00	.00	.00	.00	14.33	.00	.00	14.33
TM GROUP													
	0-1000	1.86	15.82	16.48	34.16	.00	.00	.00	.00	1.86	15.82	16.48	34.16
UPPER GR													
	0-1000	.00	5.72	.00	5.72	.00	.00	.00	.00	.00	5.72	.00	5.72
COUNTY TOTAL =		60.62	124.85	16.48	201.95	.00	.00	.00	.00	60.62	124.85	16.48	201.95
					ROUTT	COUNTY							
SUBBIT													
K													
	0-1000	6.51	27.35	.00	33.86	.00	29.42	.00	29.42	6.51	56.77	.00	63.28
	1-2000	.00	.00	.00	.00	.00	.00	12.00	12.00	.00	.00	12.00	12.00
L													
	0-1000	.00	44.12	.00	44.12	.00	.00	.00	.00	.00	44.12	.00	44.12
M													
	0-1000	1.80	59.56	.00	61.36	.00	.00	.00	.00	1.80	59.56	.00	61.36
	1-2000	.00	.00	.00	.00	.00	76.60	.00	76.60	.00	76.60	.00	76.60
N													
	0-1000	10.93	57.03	.00	67.96	.00	.00	.00	.00	10.93	57.03	.00	67.96
	1-2000	.00	.00	.00	.00	.00	22.22	.00	22.22	.00	22.22	.00	22.22
O													
	0-1000	13.28	19.51	.00	32.79	.00	.00	.00	.00	13.28	19.51	.00	32.79
P													
	0-1000	.00	8.28	.00	8.28	.00	.00	.00	.00	.00	8.28	.00	8.28
Q													
	0-1000	18.35	1.38	.00	19.73	.00	.00	.00	.00	18.35	1.38	.00	19.73
R													
	0-1000	1.13	.00	.00	1.13	.00	.00	.00	.00	1.13	.00	.00	1.13
S													
	0-1000	12.92	.50	.00	13.42	.00	.00	.00	.00	12.92	.50	.00	13.42
UNCORREL													
	0-1000	28.21	56.19	.00	84.40	.00	.00	.00	.00	28.21	56.19	.00	84.40
UPPER GR													
	0-1000	.00	5.17	.00	5.17	.00	.00	.00	.00	.00	5.17	.00	5.17
COUNTY TOTAL =		93.13	279.09	.00	372.22	.00	128.24	12.00	140.24	93.13	407.33	12.00	512.46
TOTAL =		153.75	403.94	16.48	574.17	.00	128.24	12.00	140.24	153.75	532.18	28.48	714.41
GRAND TOTAL =		278.40	542.70	1229.11	2050.21	.00	141.11	416.83	557.94	278.40	683.81	1645.94	2608.15

TONNAGE NOT INCLUDED: 1296.68
BECAUSE:
THICKNESS = CLASSIFIED BY ZONE: 1296.68

TOTAL TONNAGE FOR AREA = T5-6N,R89-90W,YAMPA FIELD,CO 3904.83

STOP
SPU'S:20.4

FIGURE 8.—Example of a tabular summary of coal resources—Continued

```
ENTER COMMAND:  TABLE
ENTER UP TO 6 CHARACTER NAME OF DATA FILE TO BE PRINTED: ANAYAM
ENTER NAME OF AREA SEARCHED (UP TO 28 CHARACTERS): T5-6N,R89-90W,YAMPA FIELD,CO
```

AVERAGE ANALYSES OF COAL IN T5-6N,R89-90W,YAMPA FIELD,CO

RANK / COAL BED	PROXIMATE ANALYSIS (PERCENT) MOISTUR	VOL MAT	FIXED C	ASH	ULTIMATE ANALYSIS (PERCENT) HYDROGN	CARBON	NITROGN	OXYGEN	SULFUR (PERCENT)	BTU	ASH SOF TEMP(F)	MAX. NO. ANALYSES AVERAGED
				MOFFAT	COUNTY							
BIT												
ABOVE H	9.7	36.3	44.1	9.9	5.7	61.3	1.4	21.2	.5	10770	****	1
BELOW H	9.7	36.8	45.9	7.6	5.6	62.8	1.2	22.1	.6	11130	****	2
F ZONE	10.0	34.9	46.6	8.8	5.5	62.0	1.3	20.6	1.7	11033	****	3
H ZONE	10.9	37.2	48.3	3.7	5.8	65.6	1.3	23.2	.5	11570	****	2
LOWER GR	9.2	36.5	43.9	10.4	5.4	62.2	1.4	19.4	1.2	11076	****	5
COUNTY AVERAGE	9.8	36.3	45.5	8.5	5.6	62.7	1.3	20.8	1.1	11126	****	13
SUBBIT												
MOORE	12.4	34.1	48.0	5.5	5.5	62.3	1.3	24.9	.5	11110	****	1
UNCORREL	15.5	33.8	46.0	4.6	5.9	62.7	1.6	25.3	.7	10654	2218	7
COUNTY AVERAGE	15.1	33.9	46.3	4.7	5.8	62.6	1.5	25.1	.7	10711	2218	8
				ROUTT	COUNTY							
BIT												
B	11.2	35.8	50.3	2.7	5.8	68.2	1.3	21.5	.5	11790	****	1
BELOW B	10.3	34.9	49.3	5.6	5.7	66.1	1.5	20.5	.6	11580	****	3
BELOW G	10.0	36.1	40.0	13.9	5.3	57.6	1.2	21.2	.8	10100	****	1
BELOW H	11.9	37.2	47.7	3.3	5.8	65.4	1.4	23.7	.5	11400	****	2
C	9.8	35.6	50.5	4.1	5.7	68.4	1.3	20.0	.5	11920	****	1
D	9.9	35.3	47.3	7.5	5.8	64.3	1.4	20.4	.6	11370	****	1
F ZONE	9.6	31.0	42.5	17.0	5.1	56.7	1.2	19.5	.6	9855	****	2
G ZONE	10.1	35.5	45.4	9.0	5.5	62.5	1.2	21.3	.6	10904	****	16
H ZONE	11.2	35.9	46.5	6.4	5.7	63.3	1.4	22.6	.5	11040	****	6
LOWER GR	9.3	34.4	49.6	6.8	5.6	66.0	1.4	19.6	.7	11555	****	4
COUNTY AVERAGE	10.3	35.3	46.5	8.0	5.6	63.4	1.3	21.2	.6	11063	****	37
SUBBIT												
L	16.9	32.2	45.5	5.4	0.	0.	0.	0.	.4	10360	2480	1
UNCORREL	17.1	32.9	45.8	4.1	5.7	62.6	1.6	13.8	.5	10240	2263	3
COUNTY AVERAGE	17.1	32.7	45.7	4.4	5.7	62.6	1.6	13.8	.5	10270	2317	4

SWI'S: 25.8

FIGURE 9.—Example of a tabular summary of coal analyses.

```
ENTER COMMAND: QUIT


THE FOLLOWING FILES HAVE BEEN CREATED DURING THIS SESSION:
   1      RYAMPA
   2      BITYAM
   3      SUBYAM
   4      AYAMPA
   5      ABTYAM
   6      ASBYAM


DO YOU WISH TO SAVE ANY OF THEM?
(ENTER Y FOR YES OR NO FOR NO)  Y
ENTER A LIST OF NUMBERS CORRESPONDING TO
THOSE FILES YOU WISH TO SAVE (IE. 1-3,5).
1,4
 STOP
SRU'S:8.5

!OFF
SRU'S:236.1      ELAPSED TIME: 00:49:54
GOOD BYE
```

FIGURE 9.—Example of tabular summary of coal analyses—Continued

the WCOAL, ECOAL, and USALYT master files. Although the updating software was designed to be as general as possible, certain peculiarities in each of the NCRDS master files make it necessary to add some unique, file-dependent programming steps. Addition of new master files to the PACER system will necessitate some additional software changes. The parts of the software logic where such changes must be made will be discussed more fully in the section describing the update software.

There are also some minor modifications to several of the GRASP routines used by PACER. These changes were implemented to adapt PACER better to the requirements of the National Coal Resources Data System. Certain of these changes were made for cosmetic purposes in the presentation of the data output. Listing of the modified GRASP subroutines appears in appendix B. These changes include modification of subroutine COLPNT, the data output routine that prints selected data-element values from a selected file in columnar form. The GRASP version permits the variable decimal selection of fixed-point real fields, based on tests for the number of significant digits in the real value. Because the number of significant digits for all real data, currently existing or contemplated in the NCRDS, is suitable for constant field output with two significant decimal digits (F8.2), the column print (COLPNT) logic was modified so that the decimal points for the real data values would be alined vertically in the output.

Subroutine MEAN was modified to ensure agreement between the total tonnages as output from MEAN and the actual total tonnages for a large number of records from a retrieval subfile. Before modification, discrepancies were caused by the summation of these tonnages in a single-precision register. The resultant summation of single-precision "noise" for several thousand records reduced the accuracy of tonnage totals to three or four significant digits. By making the appropriate variables (SUMX and SUMXS) double-precision quantities, this "noise" was removed, and the tonnage totals for a large number of records now have the same degree of precision as the tonnage entries on the individual records.

As previously mentioned, GETREC, the input and output subroutine for reading and writing records in the master-file formats, has been modified to permit the use of keyed master-file records. Keyed master-file records are essential for the development of an efficient editing and updating software package to be appended to the original GRASP system.

A recent modification of GRASP (since publica-

tion by Bowen and Botbol, 1975), permitting the merging of records from two subfiles having identical record structure, has also been implemented in the PACER version. This subroutine is called MERGE.

A final modification was made to the GRASP main program, DRIVER, to permit the user to specify UPDAte, TABLe, and MERGe as three additional commands now available with PACER. The GRASP subroutines containing these modifications are listed in appendix B. All other GRASP subroutines are used by PACED as they exist in their original form on the host computer system.

UPDATE PROGRAMS OF PACER

Exercise of the UPDAte option of PACER requires programming logic heretofore unavailable with the GRASP program software. This section will outline the overall flow of subroutine activities and summarize the functional logic inherent in each subroutine. Special note will be made of the data-dependent as well as the machine-dependent peculiarities in case the user wishes to modify these routines to operate on other data sets or has a need to transfer this software to computer systems other than the UNIVAC 1108-based INFONET system for which these routines were designed.

Figure 10 provides an outline of the flow of subroutine activities throughout the data-editing and updating process. Subroutine REVISE is the controlling subroutine that is called from the main program DRIVER when the UPDAte command is specified. Many of the overall housekeeping tasks are performed through this subroutine: tasks such as selection of the desired update procedure, specification of the subfile to be updated, opening files in the update mode, and posting the updated subfile onto the master file. Any of the five updating subroutines, ADDREC, DELREC, SEQREV, KEYREV, and BATCH may be selected while under the control of REVISE.

Subroutine ADDREC processes the new data-input files, using subroutine MATCH to check the dictionaries for matches to the alphanumeric data entries and prompting the data manager to either correct the data entries for the mismatches or to request that the mismatched data be added as an entry in the appropriate dictionary through subroutine ADDICT. If all data items on the raw data record are satisfactory, the record is rewritten in the master-file record format, integer dictionary pointers replacing the alphanumeric entries. This secondary data file is saved for future posting onto the master file at some other time when control is again returned to REVISE. Two data-dependent subroutines are currently available for reading the new data-input file and for formatting the data to permit the proper assignment of the different data strings to each of the data elements. These subroutines are called REDUSA, to read data for the USALYT file, and REDUSC, to read data for the WCOAL and ECOAL files. There are also two data-dependent subroutines which, at the option of the user, can be used to put out corrected data in a format comparable to the input formats used with the SYCOR data-entry device. These subroutines are OUTUSA to display USALYT records and OUTUSC to display WCOAL and ECOAL records.

Subroutine DELREC is used to delete records from the master file. It uses one of the data-dependent output routines, OUTUSA or OUTUSC, to produce a hard-copy listing of each record that has been deleted before eliminating that record from the master file.

Subroutine SEQREV selects, in sequence, each record from the update subfile. Subroutine MODIFY displays each record by calling either subroutine OUTUSA or OUTUSC and then selects the required dictionary data values by passing the integer dictionary pointers to subroutine SCANDC which performs the dictionary lookup for the appropriate alphanumeric data string. If the data manager wishes to change an alphanumeric data value, subroutine MATCH is called to compare the new data value with existing dictionary entries. If a match is found, the new dictionary pointer replaces the old pointer in the record. If there is no match, but the data manager wishes to have the alphanumeric data added as a dictionary entry, subroutine ADDICT is then called to perform this task.

Subroutine KEYREV selects records by key number from the update subfile. After the record has been selected, subroutine MODIFY is called, and the data-update process proceeds in the same manner as described for the SEQREV routine.

The BATCH subroutine performs batch data-value entry for all records in a given subfile. This procedure utilizes subroutine MATCH to locate the dictionary entry, if it exists, which matches the input alphanumeric data value. If there is no dictionary match, the alphanumeric data entry may be added to the dictionary through subroutine ADDICT.

The individual subroutine logic is summarized in

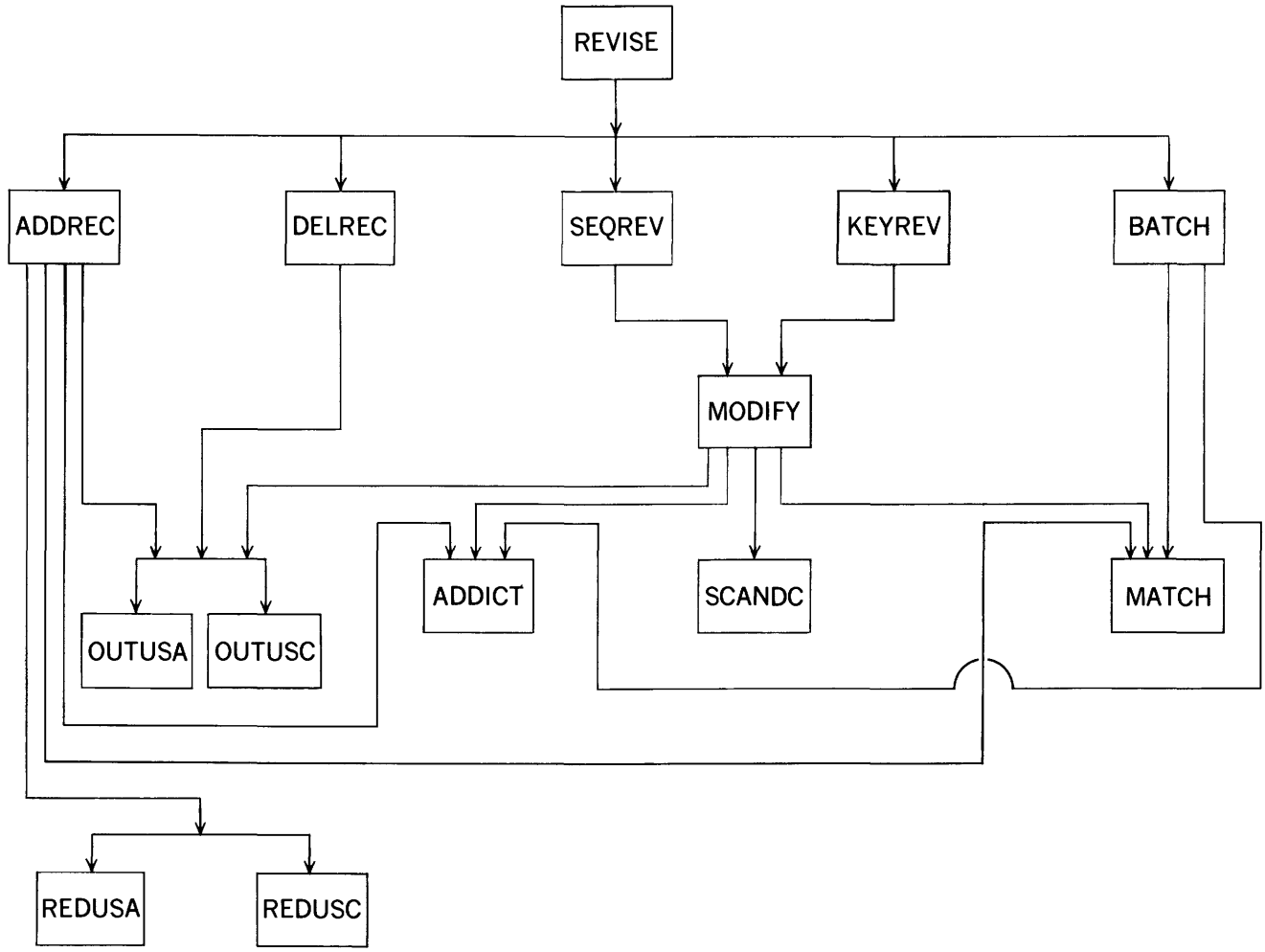


FIGURE 10.—Subroutine activity flow.

appendix C, and particular attention is directed to those peculiarities which (1) make the logic dependent upon a specific record data structure, or (2) make the logic dependent on a specific computer system. The READ parameters END, ERR, KEY, and PROMPT are not discussed in the subroutine descriptions found in appendix C. However, it should be noted that these parameters are nonstandard and not necessarily available on all hardware systems. The program listings that accompany the discussion are commented on, together with a brief description of the subroutine logic and a definition of variables important to that subroutine.

REFERENCES CITED

Barnes, F. F., 1961, Coal fields of the United States—Sheet 2, Alaska: Washington, D.C., U.S. Geol. Survey, scale 1:5,000,000.

Bowen, R. W., and Botbol, J. M., 1975, The Geologic Retrieval and Synopsis Program (GRASP): U.S. Geol. Survey Prof. Paper 966, 87 p.

Calkins, J. A., Kays, Olaf, and Keefer, E. K., 1973, CRIB—The Mineral Resources Data Bank of the U.S. Geological Survey: U.S. Geol. Survey Circ. 681, 39 p.

Meyer, R. F., 1970, Geologic provinces code map for computer use: Am. Assoc. Petroleum Geologists Bull., v. 54, no. 7, p. 1301–1305.

Sippl, C. J., and Sippl, C. P., 1974, Computer dictionary: Indianapolis, Ind., Howard W. Sams & Co., 488 p.

Trumbull, J. V. A., 1960, Coal fields of the United States—Sheet 1: Washington, D.C., U.S. Geol. Survey, scale 1:5,000,000.

U.S. Bureau of Land Management, 1973, Manual of instructions for the survey of the public lands of the United States: U.S. Bur. Land Management Tech. Bull. 6, 382 p.

U.S. Bureau of the Budget, 1965, Automatic data processing glossary Washington, D.C., U.S. Govt. Printing Office, 62 p.

U.S. National Bureau of Standards, 1973, Counties and county equivalents of the States of the United States: U.S. Bur. Standards, Federal Inf. Processing Standards (FIPS) Pub. 6–2, 35 p.

APPENDICES

APPENDIX A. NAMES USED WITH NCRDS FILES

APPENDIX B. PROGRAM LISTINGS OF MODIFIED GRASP ROUTINES

APPENDIX C. PROGRAM LISTINGS OF PACER SUBROUTINES

APPENDIX A, NAMES USED WITH NCRDS FILES

A.1.—List of meridian codes in the PMERID file with names.

01	1st Principal
39	1st Scioto River
02	2d Principal
40	2d Scioto River
03	3d Principal
41	3d Scioto River
04	4th Principal
05	5th Principal
06	6th Principal
07	Black Hills
08	Boise
09	Chickasaw
10	Choctaw
11	Cimarron
12	Copper River
42	Ellicott's Line
13	Fairbanks
14	Gila and Salt River
36	Great Miami River
15	Humboldt
16	Huntsville
17	Indian
44	Kateel River
18	Louisiana
19	Michigan
20	Montana Principal
21	Mount Diablo
37	Muskingum River
22	Navajo
23	New Mexico
35	Ohio
38	Ohio River
26	Salt Lake
27	San Bernardino
28	Seward
24	St. Helena
25	St. Stephens
29	Tallahassee
43	Twelve Mile Square
30	Uintah Special
45	Umiat
31	Ute
32	Washington
33	Willamette
34	Wind River
96	Principal meridian and base line unknown
99	Various principal meridians and base lines

A.2.—List of names in the REGION dictionary

```
ALASKA PENINSULA
APPALACHIAN
ATLANTIC COAST
BIGHORN BASIN
BLACK HILLS
CENTRAL ALASKA
COOK INLET-SUSITNA
DENVER
EASTERN
FORT UNION
GREEN RIVER
HAMS FORK
MISSISSIPPI
NORTH CENTRAL
NORTHERN
NORTHERN ALASKA
PENNSYLVANIA ANTHRACITE
POWDER RIVER
RATON MESA
RHODE ISLAND META-ANTH
SAN JUAN RIVER
SOUTHEASTERN ALASKA
SOUTHWESTERN
SOUTHWESTERN UTAH
TEXAS
TERTIARY LAKE BEDS
UINTA
WESTERN
WIND RIVER
```

A.3.—List of names in the FIELD dictionary

ALADDIN
ALKALAI BUTTE
ALTON
ARKANSAS VALLEY
BARBER
BASIN
BAYFIELD
BISTI AREA
BLACK MESA
BOOK CLIFFS
BOULDER WELD
BP COSTELLO CREEK
BRIGGSDALE
BROAD PASS
BROKEN ARROW D
BUFFALO DE SMET
BUICK MATHESON
C IOWA
CAMBRIA
CANNONBALL RV
CANON CITY
CARBONDALE
CART JOR DELMUE
CAVE HILLS
CEDAR MTN
CENTER POWDER
CENTRAL REGION
CENTRALIA CH
CERRILLOS
CHACO CANYON A
CHECOTAH
COALVILLE
COASTAL PLAIN
COLLINSVILLE D
COLORADO SPGS
COOS BAY
CORTEZ
CRESTED BUTTE
CROWNPT CANYN A
DANFORTH HILLS
DATIL MOUNTAIN
DEER CREEK
DRY CHEYENNE
DURANGO
EAGLE PASS DIST
EAST CENTRAL
EAST CENTRAL RG
EATON
EMERY
EVANSTON
FAIRFAX M
FOOTHILLS
FRUITLAND AREA
GALLUP
GARLAND
GEBO
GILLETTE
GLENROCK
GRAND HOGBACK
GRAND MESA F
GRASS CREEK
GREAT DIVIDE B
GREEN RIVER D
GREYS RIVER
HANNA
HARMONY
HENRY MOUNTAINS
HORSESHOE CREEK
HOWE WILBURTON
HUDSON
ISABELFIRESTEEL
JACKSON HOLE
JARVIS CREEK
KAIPAROWITS
KELSO CASTLE R
KEMMERER
KENAI
KINDIT BASIN
KOLOB
LA VENTANA A
LABARGE RIDGE
LEHIGH DISTRICT
LEXINGTON RG
LIGHTNING CREEK
LITTLE POWDER R
LITTLE SNAKE R
LOS PINOS RIVER
LOST CREEK
LOST SPRINGS
LOWER WHITE RIV
LTTL MISSOURI R
MA CHICKALOON R
MA WISHBONE H
MCALESTER DIST
MCDOUGAL
MEETEETSE
MELMONT
MENDOTA REGION
MORTON
MT PLEASANT
MUDDY CREEK
MUSKOGEE DIST
N CENTRAL
NA COLVILLE R

A.3.—List of names in the FIELD dictionary—Continued

NA CRWN BF C BT
NA IKPIKPUK R
NA KOKOLIK R
NA KUGRUA R
NA KUK R
NA KUKPOWRUK R
NA MEADE R
NA UTUKOK R
NAVAJO AREA
NEWCASTLE GR
NEWCOMB AREA
NIBLOCK
NN CALIF CREEK
NN HEALY CREEK
NN LIG CREEK
NN REX CREEK
NN SAVAGE R
NN TATLANIKA
NN WOOD R
NORTH PARK
NORTHEASTERN
NORTHEASTERN RG
NORTHERN OKLA
NORTHWESTERN RG
NUCLA NATURITA
OKMULGEE DIST
OREGON BASIN
PAGOSA JUNCTION
PAGOSA SPRINGS
PILOT BUTTE
PINEDALE
PORTER DISTRICT
POWDER R DIST
POWDER RIVER
PUMPKIN BUTTE
RAMAH FONDIS
RATON
RENTON
ROCK CREEK
ROCK SPRINGS
ROSLYN
S MNT TAYLOR A
SALINA CANYON
SALLISAW DIST
SAN CARLOS DIST
SAN JUAN RIVER
SAN MATEO AREA
SANTO TOMAS
SC IOWA
SCRANTON
SE IOWA
SEGO
SHERIDAN
SIERRA BLANCA
SILVER TIP
SKAGIT COUNTY
SKULL CREEK
SLIM BUTTES
SOMERSET
SOUTH CENTRALRG
SOUTH PARK
SOUTHEASTERN
SOUTHERN OKLA
SPIKETON
SPOTTED HORSE
STANDING ROCK A
STAR LAKE AREA
STERLING
STIGLERDISTRICT
STONEVILLE
SU BELUGA R
SU CAPPS GLAC D
SU CHUITNA R
SU SW OF TYONEK
SU YENTNA R
SUMMIT CREEK
SUNDANCE
SUSSEX
SW IOWA
TABBY MOUNTAIN
TABLE MOUNTAIN
TANEUM
TAYLOR
TIGER MTN
TIJEVAS
TONGUE MESA
TRINIDAD
UNA DEL GATO
UNCORRELATED
VERNAL
W IOWA
WALES
WASATCH PLATEAU
WELLINGTON
WEST CENTRAL RG
WESTERN REGION
WHATCOM COUNTY
WILKESON
YAMPA
ZUNI AREA
BEULAH ZAP
NEW SALEM
NOONAN
SENTINEL BUTTE

A.4.—List of district names in the DISTRCT dictionary

ANTHRACITE RIDGE
BAYFIELD-YELLOW JACKET
BELUGA LAKE
BRIGGSDALE (AREA)
BROAD PASS
BUICK-MATHESON (AREA)
CALIFORNIA CREEK
CAPPS GLACIER
CHICKALOON
CHUITNA RIVER
COLVILLE RIVER
CORTEZ (AREA)
CORWIN BLUFF-CAPE BEAUFT
COSTELLO CREEK
EAGLE-CIRCLE
EAGLE PASS
EAGLE SPRING (AREA)
EATON (AREA)
FOOTHILLS
HEALY CREEK
HOMER
KENAI
KOBUK RIVER
KOKOLIK-UTOKOK RIVERS
KOOTZNAHOO INLET
KOYUKUK RIVER
KUK-KUGRUA RIVERS
KUKPOWRUK RIVER
KUSKOKWIM
LIGNITE CREEK
LITTLE SUSITNA
MEADE-IKPIKPUK RIVERS
MESA VERDE (AREA)
NORTH CENTRAL
PAGOSA JUNCTION
RAMAH-FONDIS (AREA)
RAMPART
RED MESA (AREA)
REX CREEK
RUBY-ANVIK
SAN CARLOS
SANTO THOMAS
SAVAGE RIVER
SCRANTON
SEWARD PENINSULA
SKWENTNA RIVER
TATLANIKA CREEK
TERLINGUA
TYONEK
UNALAKLEET
WELLINGTON (AREA)
WISHBONE HILL
WOOD RIVER
YENTNA RIVER

A.5.—List of formation names in the FORMATN dictionary[1]

ADAVILLE
AGUJA
BANDERA SHALE
BLACKHAWK
BOGGY
CHANDLER
CHEROKEE
CHICKALOON
COALEDO
COALMONT
COWLITI
CHUCKANUT
DAKOTA
DAKOTA SS
DENVER
DOMENGINE
DOUGLAS
EAGLE SS
EMERY SS
EVANSTON
FERRON SS
FORT UNION
FRONTER ADAVLLE
FRONTIER
FRUITLAND
FTLD MNE DAKSS
FTUNION HELLCRK
FTUNION WASATCH
GARNER
GRAFORD
HARTSHORNE SS
HELL CREEK
ILES
ILES WILLIAMSFK
ILSWILFKLNCFTUN
JACKSONGROUP
KANSAS CITY
KENAI
KNIGHT
LABETTE SHALE
LAKOTA SS
LARAMIE
LNCE FTUN WSTCH
LONE
LOWER CHEROKEE
MARMATON
MCALESTER
MENEFEE
MESA MEET
MESA WASATCH
MESA MEET FTUN
MESA VERDE
MESAVERDE
MESAVERDE GROUP
MESAVERDE HANNA
MESVDE FRONTIER
MILLSAP LAKE
MOUNT SELMAN
NORTH HORN
OLMOS
PALO PINTO
PERMIAN AGE
PRICE RIVER
PRINCE CREEK
PUEBLO
PUGET GROUP
RATON
ROSLYN
SAVANNA
SEVERY SHALE
SEMINOLE
SENORA
SIGMILE CANYON
SKOOKUMCHUCK
STRAIGHT CLIFFS
TEMBLER
THRIFTY
TOREVA
TOUTLE
TROPIC DAKOTA
UNCORRELATED
UPPER CHEROKEE
VERMEJO
VERMEJO RATON
WABAUNSEE
WASATCH
WEPO
WILCOG GROUP
WILLIAMS FORK
LANCE

[1] The stratigraphic nomenclature used in this report is from many sources and may not necessarily follow U.S. Geological Survey usage.

A.6.—List of names in the BED dictionary [1]

A
A BD
A RATON
AB
ABBOTT
ABC
ALFREDA
ALLEN
ALVEY COAL ZONE
ANCHOR
ANDERSON
ANOBASWF
APACHE
ARVADA
B
B BD
B RATON
BAGLEY
BALD KNOLL COAL ZONE
BALLARD COAL ZONE
BASWF
BD 1
BD 10
BD 2
BD 3
BD 4
BD 5
BD 9
BD GROUP
BEAR CANYON
BEAR CANYON NO 6
BEAR RIVER
BEAVER HILL
BECKWITH
BED 1
BED 2
BED 3
BED 4
BED 5
BED A
BED ABOVE CAMERON
BED ABOVE HEALY
BED ABOVE WALTERS
BED BELOW HEALY
BED BELOW MURRAY
BED BELOW SCHUMAN
BED BELOW UCROSS
BED D
BED E
BED H
BED J
BELLINGHAM NUMBER 1
BELLINGHAM NUMBER 2
BELOW RIACH
BELOW RIACH DOUBLE
BERWIND
BEVIER
BEVIER WHEELER
BIG
BIG BEN
BIG DIRTY
BIG ELK
BLACK BEAR
BLACK CARBON
BLACKSMITH
BLEVINS GROUP
BLIND CANYON
BLUE CANYON
BLUE MOUND
BOB WRIGHT
BOISE
BONCARBO
BRIDGEPORT
BROOKS
BROOKSIDE
BROWN BEAR
BULL CREEK
BUNKER HILL
BURNT
C
C BD
CAMEO
CAMERON
CANDLAND
CANYON
CAPRON
CARBON
CARBON BAHNE NO 1
CARBON BAHNE NO 2
CARBON BAHNE NO 3
CARBONADO NUMBER 5
CARBONADO NUMBER 8

[1] The stratigraphic nomenclature used in this report is from many sources and may not necessarily follow U.S. Geological Survey usage.

A.6.—List of names in the BED dictionary [1]—Continued

CARBONERA
CARBONERO
CASS
CASTLEGATE A
CAVANAL
CAVANAUGH NUMBER 2
CEDAR CREEK NO 1
CEDAR CREEK NO 2
CEDAR CREEK NO 3
CEDAR MTN NO 1
CEDAR MTN NO 2
CHANDLER
CHAFFIN
CHARLESTON
CHERRY CREEK
CHESTERFIELD
CHRISTENSEN ZONE
CIRUELA
CLIFFLAND
COAL NO 1
COAL NO 2
COAL NO 3
COAL NO 4
COAL NO 5
COAL NO 6
COAL NO 7
COAL NO 8
COKEDALE
COLUMBUS
COMO
COLBORN
COTTONWOOD
CROCKER
CROWEBURG
D
D AND F
D BD
DALE NUMBER 4
DALE NUMBER 7
DALTON
DAWSON
DE
DELAGUA NO 1
DIETI NO 1
DISCOVERY
DOLLY VARDEN
DRY CREEK
DURHAM NUMBER 2
DUTCH
E
E BD
EIGHT FOOT
ELK NUMBER 1
ELK NUMBER 2
ELMO
EMERY COAL ZONE
EMPIRE
ERAM
EUREKA
F
F BD
FELIX
FERRON COAL ZONE
FF 1
FF 10
FF 11
FF 12
FF 13
FF 14
FF 15
FF 16
FF 17
FF 18
FF 19
FF 2
FF 20
FF 21
FF 22
FF 23
FF 24
FF 3
FF 4
FF 5
FF 6
FF 7
FF 8
FF 9
FF GROUP
FIRESTEEL
FISH CREEK

[1] The stratigraphic nomenclature used in this report is from many sources and may not necessarily follow U.S. Geological Survey usage.

A.6.—List of names in the BED dictionary [1]—Continued

FLEMING
FLORESTA
FRANKLIN NUMBER 10
FRAZIER
FREDERICK
FRONTIER BED 1
FRONTIER BED 2
FRONTIER BED 3
FRONTIER BED 4
FRONTIER COAL ZONE
FULTON NUMBER 12
G
GEM
GILSON
GLACIER
GOFF 1
GOFF 10
GOFF 11
GOFF 12
GOFF 13
GOFF 14
GOFF 15
GOFF 16
GOFF 17
GOFF 18
GOFF 19
GOFF 2
GOFF 20
GOFF 21
GOFF 22
GOFF 3
GOFF 4
GOFF 5
GOFF 6
GOFF 7
GOFF 8
GOFF 9
GOLDEN GLOW
GREEN NUMBER 7
HARRIS
HARTSHORNE
HASTIE
HASTIE PLUS
HASTINGS
HEALY
HENDERSON COAL ZONE
HENDERSON ZONE
HENRYETTA
HIAWATHA
IVIE
JEFF HILL
JOHN HENRY MEMBER
JONES
K BED
KEBLER NO 2
KENILWORTH
KEYSTONE
KUMMER NUMBER 0
KUMMER NUMBER 1
KUMMER NUMBER 4
L BED
LADD NUMBER 2
LADD NUMBER 3
LADD NUMBER 4
LADDSDALE
LAKE WHATCOM
LANDSBURG NO 1
LAY SECTION
LEAVELL
LENNOX
LENOX
LEXINGTON
LION CANYON 1
LION CANYON 10
LION CANYON 11
LION CANYON 12
LION CANYON 13
LION CANYON 14
LION CANYON 15
LION CANYON 16
LION CANYON 17
LION CANYON 18
LION CANYON 19
LION CANYON 2
LION CANYON 20
LION CANYON 21
LION CANYON 22
LION CANYON 23
LION CANYON 24
LION CANYON 3
LION CANYON 4
LION CANYON 5

[1] The stratigraphic nomenclature used in this report is from many sources and may not necessarily follow U.S. Geological Survey usage.

A.6.—List of names in the BED dictionary [1]—Continued

LION CANYON 6
LION CANYON 7
LION CANYON 8
LION CANYON 9
LION CANYON A
LION CANYON B
LION CANYON C
LION CANYON D
LITTLE DIRTY
LONSDALE
LOW CARBONERA
LOW GROUP A BED
LOW GROUP B BED
LOW GROUP C BED
LOW GROUP D BED
LOW GROUP E BED
LOW HOLGATE
LOWER
LOWER ALAMO
LOWER BUNKER HILL
LOWER CAMERON
LOWER COAL FORD COAL
LOWER COAL ZONE
LOWER CULVER ZONE
LOWER DIRTY
LOWER GROUP
LOWER HARTSHORNE
LOWER LUDLOW
LOWER MEMBER
LOWER MEMBER RATON
LOWER MESAVERDE
LOWER MYSTIC
LOWER PIEDMONT
LOWER ROBINSON
LOWER RUGBY
LOWER SOPRIS
LOWER STARKVILLE
LOWER SUNNYSIDE
LOWER THOMPSON
LOWER WITTEVILLE
LOWER ZONE ORD AREA
LOWTHIRWF
LUCAS CREEK
MAJESTIC
MAMMOTH
MANBECK
MARTINEI
MAY CREEK
MC ALESTER
MCKAY
MCNEILL
MESAVERDE COAL ZONE
MEXICAN CREEK
MENDOTA
MID CARBONERA
MID GROUP F ZONE
MID GROUP G ZONE
MID GROUP H ZONE
MID GROUP J BED
MID HOLGATE
MIDDLE
MIDDLE GROUP
MIDDLE MEMBER
MIDDLE MEMBER RATON
MIDDLE MESAVERDE
MIDWF
MINERAL
MITCHELL MINE
MONAHAN
MONTERVILLE
MONUMENT PEAK
MORGAN NUMBER 7
MORLEY
MORRIS
MUDDY NO 1
MUDDY NO 2
MUDDY NOS 1 AND 2
MULBERRY
MULDOON
MULKY
MURRAY
MUTUAL
MYSTIC
N BED
NEW LAKE HOUNGS NO 2
NEW ROUSE
NEWCASTLE
NEWENHAM
NISQUALLY
NO 1

[1] The stratigraphic nomenclature used in this report is from many sources and may not necessarily follow U.S. Geological Survey usage.

A.6.—List of names in the BED dictionary [1]—Continued

NO 1 BED
NO 2
NO 2 BED
NO 3
NO 3 AND 4
NO 3 BED
NAVY NUMBER 4
NAVY NUMBER 6
NO 4
NO 5
NO 6
NO 7
NODAWAY
NONAC
NUMBER 1
NUMBER 2
NUMBER 3
NUMBER 4
NUMBER 5
NUMBER 6
NUMBER 10
NUMBER 11
NUMBER 12
NUMBER 2 5
NUMBER 4 5
NUMBER 7
NUMBER 8
O BED
OCCIDENTAL
OCCIDENTAL NO 1
OCCIDENTAL NO 2
OCCIDENTAL NO 3
OCCIDENTAL NO 6
OCCIDENTAL NO 14
OCEAN WAVE
OTTAWA
P BED
PALISADE
PARIS
PENITENTIARY
PIEDMONT
PLACITA
PLANT NUMBER 6
PLANT NUMBER 7
POCAHONTAS
PRETTY
PRIMERO
PROGRESSIVE
PRYOR
Q BED
R BED
RADIANT
RAINBOW
RAPSON
RAVENSDALE NO 3
RAVENSDALE NO 4
RAVENSDALE NO 5
RAVENSDALE NO 9
RED ASH
REES COAL ZONE
RIACH
RIDER
ROCK CANYON
ROCKVALE
ROLAND
ROSLYN NUMBER 5
ROWE
ROYAL GORGE
RYAN NUMBER 1
S BED
SAN PEDRO
SANTO TOMAS
SCHUMAN
SCOTT
SECOR
SENIOR
SHOO FLY
SIBLEY
SILVER LAKE
SIX FOOT
SLIDE HOLLOW
SMIRL COAL ZONE
SMITH
SMOKY HOLLOW MEMBER
SNELL
SOPRIS
SPRINGBROOK
STIGLER
STRAIGHT CLIFFS ZONE
SUDDUTH

[1] The stratigraphic nomenclature used in this report is from many sources and may not necessarily follow U.S. Geological Survey usage.

A.6.—List of names in the BED dictionary [1]—Continued

SUMMIT
SUNBEAM
SUNDAY CREEK
SUNNYSIDE
SUNSET NUMBER 1
SUNSET NUMBER 2
SUNSET NUMBER 7
TANK
TAHLOR MINE
TEBO
THAYER
THOMAS
THURBER
THREE PINES
TYSON
TIMAR
TM GROUP
TONO NUMBER 1
TONO NUMBER 2
TOPWF
TROPIC OAK INTERVAL
UCROSS
ULM NO 2
UNCORRELATED
UP CARBONERA
UP HOLGATE
UPPER
UPPER ALAMO
UPPER AND LOWER
UPPER BEAR CANYON
UPPER BUNKER HILL
UPPER CAMERON
UPPER CULVER ZONE
UPPER HARTSHORNE
UPPER HIAWATHA
UPPER IVIE
UPPER LUDLOW
UPPER MEMBER
UPPER MEMBER RATON
UPPER MESAVERDE
UPPER PART
UPPER ROBINSON
UPPER RUGBY
UPPER STARKVILLE
UPPER SUNNYSIDE
UPPER THOMPSON
VICTORY
WADGE
WALKER
WALL
WALSEN
WALTERS
WASATCH
WATTIS
WEIR PITTSBURG
WHEELER
WHITEBREAST
WILEY
WILKESON NUMBER 1
WILKESON NUMBER 2
WILKESON NUMBER 3
WILKESON NUMBER 4
WILKESON NUMBER 5
WILKESON NUMBER 7
WILLIAMSBURG
WINCHESTER
WOLF CREEK
WRIGHT NUMBER 8
BEULAH ZAP
COTEAU
DUNN CENTER
FRYBURG
HANKS
HAYNES
HARMON
MANHAVEN
MIDDLE WILLISTON
NOONAN
SCRANTON
MEDORA
T-CROSS
UPPER WILLISTON
WILLISTON
WILTON

[1] The stratigraphic nomenclature used in this report is from many sources and may not necessarily follow U.S. Geological Survey usage.

A.7.—List of quadrangle names in the QUAD dictionary

ACORD LAKES NE QD
ACORD LAKES NW QD
ACORD LAKES SE QD
ACORD LAKES SW QD
ALTON
BALD KNOLL
BRYCE POINT
CAINEVILLE
CANAAN CR QD
CARCASS CANYON QD
CASTLE DALE NE QD
CASTLE DALE NW QD
CASTLE DALE SW QD
CASTLEGATE KHUNE QDS
CASTLEGATE MATTS SUMMIT QDS
CEDAR MOUNTAIN
CLIFF RIDGE QD
COAL CR TO ORDERVILLE CANYON
COAL CREEK
COALVILLE QD
COLLET TOP QD
COW FLAT
CURRANT CR
DAVE CANYON QD
DEATH RIDGE QD
DONKEY FLAT QD
DRY FORK VERNAL NW QDS
EAST OF NAVAJO QD
EMERY EAST
EMERY THREE NE
EMERY THREE NW
EMERY THREE SE
EMERY THREE SW
FACTORY BUTTE NW
FERRON CANYON QD
FLAGSTAFF PEAK QD
FLOY CANYON SE SW
GRIFFIN POINT QD
GUNSIGHT BUTTE NE
GUNSIGHT BUTTE NW
HENRIEVILLE QD
HIAWATHA NE QD
HIAWATHA NW QD
HIAWATHA SE QD
HIAWATHA SW QD
KOLOB PK ORDERVILLE CANYON N
LAKE MOUNTAIN QD
LOST CREEK
MESA BUTTE
MOUNT ELLEN NW SW
MOUNT PENNELL NW NE
MT PLEASANT HUNTINGTON RES
NAPLES QD
NEEDLE EYE POINT QD
NIPPLE BUTTE NE
NIPPLE BUTTE SE
NOTOM NE SE
ORDERVILLE CANYON NE
ORDERVILLE CANYON SE
ORDERVILLE NE SE
ORDERVILLE SW
PAGE RANCH STODDARD MOUNTAI
PARIA NW
PETES COVE QD
PINE LAKE QD
RAINBOW POINT
RASBERRY KNOLL
RASMUSSEN HOLLOW QD
ROCK CREEK
SALINA CANYON
SCOFIELD NE QD
SCOFIELD NW QD
SCOFIELD SE QD
SCOFIELD SW QD
SEEP FLAT
SEGO CANYON SE
SEGO CANYON SW
SHIP MOUNTAIN POINT QD
SKUTUMPAH CREEK PODUNK CREEK
SLICK ROCK BENCH-BUTLER VALL
SNAKE JOHN REEF QD
SOLDIER SUMMIT SE QD
STEINAKER RES VERNAL NE QDS
STERLING
SUNNYSIDE NW QD
SUNNYSIDE SE QD
SUNNYSIDE SW QD
TABBY MTN
TABIONA QD
TROPIC CANYON QD
UPPER VALLEY QD
WAGON BOG MESA NE CAVE POINT
WALES
WELLINGTON MINNIE MAUD W QD
WELLNGTN NE MINNIE MAUD CR E
WESTWATER CR N
WOODSIDE NE QD
WOODSIDE SE QD

A.8.—List of publications in SOURCE dictionary

ALTON OPEN FILE REPT
ARIZONA BOM BULL 182
BOOK CLIFFS OPENFILE
BULL 70 SGS KANSAS
BVILLE STRIP REPT
COAL RESERVES OF WA
COAL RESOURCES IOWA
COALVILLE TABBY SEGO
EMERY OPEN FILE REPT
HENERY MTN COAL FIELD
IDAHO MCGEOL PHAM 92
KAIPAROWITS OPENFILE
KOLOB HARMONY FIELDS
LANDIS OPEN FILE
MIN & HOH RES OF ARI
MISSOURI GS RI NO48
NMBMMR MEMOIR 25
OKLA GS TBL 40 TO 58
SEVIER GOOSE LOST OF
TEX B ECON GEO RI 50
THE COMPASS SIGGAMEP
USGS BULL 1042 J
USGS BULL 1042 O
USGS BULL 1050
USGS BULL 1051
USGS BULL 1072 C
USGS BULL 1072 G
USGS BULL 1072P
USGS BULL 1078
USGS BULL 1242D
USGS BULL 9828
USGS BULLETIN 1242B
USGS CIRCULAR 159
USGS CIRCULAR 226
USGS CIRCULAR 53
USGS CIRCULAR 81
USGS CIRCULAR 89
USGS MAP C 20
USGS MAP C 26
USGS MAP C 4
USGS MAP OM 109
USGS MAP OM 138
USGS MAP OM 149
UTAH MONO SERIES 1
UTAH MONO SERIES 2
VERNAL OPEN FILE RPT
WASATCH OPENFILE RPT
NDGS BULL 4
NDU DIV MIN CIRC 2
NDU DIV MIN CIRC 5
NDU DIV MIN CIRC 8
NDU DIV MIN CIRC 11
USBM TECH PAPER 700

APPENDIX B, PROGRAM LISTINGS OF MODIFIED GRASP ROUTINES

```
      SUBROUTINE HELP(WORDS)
      INTEGER WORDS(13)
      REAL*8 TEXT(11,13)
      DATA TEXT/%
       '- INITIA','TES THE ','REQUEST ','FOR RETR','IEVAL CR','ITERIA T'%
         ,'O BE ENT','ERED IN ','THE FORM',': NAME R','EL VALUE',%
       '- INITIA','TES THE ','REQUEST ','FOR A LO','GICAL EX','PRESSION'%
         ,' TO BE E','NTERED','USING LO','GICAL OP','ERATORS.',%
       '- INITIA','TES THE ','SEARCH O','F A FILE',' BASED U','PON PREV'%
         ,'IOUSLY E','NTERED','CONDITIO','NS AND L','OGIC.',%
       '- ALLOWS',' THE USE','R TO LIS','T SELECT','ED VALUE','S (VARIA'%
         ,'BLE NAME','S WILL','BE ASKED',' FOR) IN',' A FILE.',%
       '- ALLOWS',' THE USE','R TO SEL','ECT OR C','HANGE TH','E DATA B'%
       ,'ASE TO B','E USED.',3*'      ',%
       '- TERMIN','ATES THE',' SYSTEM.',' ENTERIN','G ! IN R','ESPONSE '%
         ,'TO A PRO','MPT WILL','ALSO STO','P THE SY','STEM.',%
       '- USED T','O PRINT','ITEM NAM','ES, THEI','R TYPES','AND DEFI',%
        'NITIONS','IN A','SELECTED',' SET OF','GROUPS.','%
       '- USED T','O OBTAIN',' THE ABO','VE COMMA','ND DEFIN','ITIONS.',%
         5*' ',%
       '- LISTS','THE FILE','S WHICH','HAVE BEE','N USED A','S WELL A',%
        'S THE CO','NDITIONS','AND LOGI','C ENTERE','D.',%
       '- PRINTS',' ALL ITE','MS PRESE','NT FOR E','ACH RECO','RD IN A',%
        'SELECTED',' FILE.','WAITS AF','TER EACH',' N LINES',%
      '- PROVID','ES FOR T','HE COMPU','TATION O','F FUNCTI',%
       'ONS ON I','TEMS IN','A DATA','SET (OR','FILE).','%
       '        ',%
       '- COMBIN','ES THE C','ONTENTS ','OF SEVER','AL SELEC',%
       'TED SUBF','ILES INT','O A     ','SINGLE S','UBFILE. ',%
       '        ',%
       '- PERMIT','S THE SE','LECTION ','OF SPECI','ALLY SOR',%
       'TED AND ','FORMATTE','ED      ','TABULAR ','OUTPUT D',%
       'ISPLAYS.'/
      PRINT 501,(WORDS(J),(TEXT(I,J),I=1,11),J=1,13)
      501 FORMAT('0THE COMMANDS WHICH MAY BE ISSUED ',%
        '(AND THEIR MEANING) ARE LISTED BELOW:'///%
        ('0',A4,8A8/7X,3A8))
          PRINT 1001
      1001 FORMAT(' UPDA- PERMITS THE ADDITION, THE DELETION, OR THE'%
           ' MODIFICATION OF RECORDS'/'       OR PORTIONS OF RECORDS'%
           ' BELONGING TO THE MASTER FILE.'//)
      RETURN
      END
```

```
4100  SUBROUTINE COLPNT(NPAGE,*)
4101  DOUBLE PRECISION DBLNK,AREA,LINE(20),NAMES,LABEL,%
4102   VNAMES(20),BUFFER(15,20),FMT(3),FMTS(8),ONAME
4103  LOGICAL DISK
4104  COMMON NAMES,ITYPE,PNTS,IDIM
4105  COMMON /EXPRNS/ POLISH,ICODE,LPS
4106  INTEGER PNTS(400),BLANK,TANK,USED(20)
4107  DIMENSION ITYPE(400),BITEM(15,25),ITEMS(20),LASTDX(15,20),%
4108   IREC(400),REC(400),NAMES(400),TANK(30),LABEL(25),LIST(25),%
4109   POLISH(15,8),ICODE(15,8),LPS(8),EQUATE(5)
4110  EQUIVALENCE (REC(1),IREC(1)),(IVAL,VAL),(TANK(1),AREA)
4111  DATA NDICT,NBIN,NFILE/9,9,11/,BLANK,DBLNK/' ','      '/
4112  DATA FMT,FMTS/'(',' ',')','F8.6','F8.5','F8.4',%
4113   'F8.3','F8.2','F8.1','F8.0','1PE8.1'/
4114  DATA EQUATE/'EQUA','TE 1','5',2*' '/
4115  INTEGER SPACE(3187)
4116  COMMON /SCRTCH/ SPACE
4117  EQUIVALENCE (SPACE(1),BUFFER),(SPACE(601),LASTDX),%
4118   (SPACE(901),BITEM),(EQUATE(4),ONAME)
4119  KOUNT=0
4120  DO 1 K=1,20
4121  USED(K)=0
4122  DO 1 I=1,15
4123  1 LASTDX(I,K)=-999999
4124  46 PRINT 50
4125  50 FORMAT(' ITEMS MAY BE PRESENTED IN A SORTED ORDER.')
4126  READ(5,51,PROMPT='ENTER 1 FOR A SORT, 0 OTHERWISE.: ',END=995)%
4127  ISORT
4128  51 FORMAT(I1)
4129  PRINT 53
4130  53 FORMAT(' THIS LIST CAN BE DIRECTED TO'/%
4131   ' YOUR TERMINAL OR TO A SYSTEM DISK.'/%
4132   ' ENTER 1 FOR FORMATTED DISK ONLY, OTHERWISE ENTER 0')
4133  READ(5,51,END=995) I
4134  DISK=I.NE.0
4135  IF(.NOT.DISK) GO TO 49
4136  IF(ISORT.NE.1) GO TO 48
4137  PRINT 47
4138  47 FORMAT(' GRASP IS UNABLE TO SORT OUTPUT ',%
4139   'DIRECTED TO A DISK. REENTER YOUR CHOICE.')
4140  GO TO 46
4141  48 READ(5,54,PROMPT='ENTER NAME OF DISK FILE: ',%
4142   END=995) ONAME
4143  54 FORMAT(A6)
4144  CALL OBEY(EQUATE,5)
4145  49 CALL VLIST(VNAMES,ITEMS,NUM,&990)
4146  IF(NUM.EQ.0) GO TO 995
4147  NUM=MIN0(NUM,20)
4148  CALL OBEY('USAGE,COPY',3)
4149  IF(ISORT.EQ.0) GO TO 52
4150  ISORT=1
4151  CALL RSZ(8*NUM)
4152  CALL KEYC(1,8*NUM,7,1)
```

```
52 IF(DISK) GO TO 900
PRINT 44,(VNAMES(I),I=1,NUM)
44 FORMAT(//1X,20A10)
900 IF(ISORT.LT.2) CALL GETREC(IREC,&999)
IF(ISORT.EQ.1) GO TO 2
IF(DISK) GO TO 8
KOUNT=KOUNT+1
IF(KOUNT.LE.NPAGE) GO TO 8
KOUNT=0
CALL PAUSE(&999)
PRINT 44,(VNAMES(I),I=1,NUM)
8 IF(ISORT.NE.2) GO TO 2
CALL SRTRET(LINE,L)
IF(L.EQ.0) GO TO 998
GO TO 801
2 DO 800 JJ=1,NUM
AREA=DBLNK
II=ITEMS(JJ)
IF(II.GT.0) GO TO 9
II=-II
VAL=EVAL(IREC,ICODE(1,II),POLISH(1,II),LPS(II),&800)
GO TO 20
9 IVAL=IREC(II)
IF(IVAL.EQ.BLANK) GO TO 800
KIND=ITYPE(II)
IF(KIND.LT.10) GO TO 7
KIND=KIND-10
GO TO 28
7 GO TO (10,20,28,40),KIND
10 ENCODE(AREA,11) IVAL
GO TO 800
20 IF(VAL.EQ.0.) GO TO 22
A=ALOG10(ABS(VAL))
IF(A.GE.5.) GO TO 23
IF(A.LE.-4.) GO TO 23
22 LK=5
GO TO 24
23 LK=8
24 FMT(2)=FMTS(LK)
ENCODE(AREA,FMT) VAL
GO TO 800
28 IF(USED(JJ).EQ.0) GO TO 30
DO 29 K=1,15
IF(IVAL .EQ. LASTDX(K,JJ)) GO TO 27
29 CONTINUE
GO TO 30
27 AREA=BUFFER(K,JJ)
GO TO 800
30 IKEY=10000*KIND+IVAL
READ(NDICT,KEY=IKEY) J,(TANK(I),I=1,J)
USED(JJ)=MOD(USED(JJ),15)+1
NUSED=USED(JJ)
BUFFER(NUSED,JJ)=AREA
LASTDX(NUSED,JJ)=IVAL
```

```
4207  GO TO 800
4208  40 READ(NBIN,KEY=II) K,M,(LABEL(J),(BITEM(I,J),I=1,K),J=1,M)
4209  CALL BLIST(LIST,NUMS,IVAL)
4210  AREA=LABEL(LIST(1))
4211  800 LINE(JJ)=AREA
4212  IF(ISORT.NE.1) GO TO 801
4213  CALL SRTREL(LINE,8*NUM)
4214  GO TO 900
4215  801 IF(.NOT.DISK) GO TO 802
4216  WRITE(15,5) (LINE(JJ),JJ=1,NUM)
4217  GO TO 900
4218  802 PRINT 5,(LINE(JJ),JJ=1,NUM)
4219  GO TO 900
4220  999 REWIND NDICT
4221  IF(ISORT.NE.1) GO TO 998
4222  ISORT=2
4223  GO TO 900
4224  998 CALL OBEY('USAGE,CHANGE',3)
4225  IF(DISK) REWIND 15
4226  995 REWIND NFILE
4227  RETURN
4228  990 REWIND NFILE
4229  REWIND NDICT
4230  RETURN 1
4231  5 FORMAT(1X,20A10)
4232  11 FORMAT(I8)
4233  END
```

```
3800  SUBROUTINE MEAN
3801  COMMON NAMES,ITYPE,IPTS,IDIM
3802  COMMON /FTNCOM/ TAGS,IREC,ARGS,NARGS,IFTN,NFTN
3803  DIMENSION ARGS(6,5),NARGS(5),IFTN(5),ITYPE(400),IPTS(400),%
3804   IREC(400),NSUM(5),VMAX(5),VMIN(5)
3805  DOUBLE PRECISION NAMES(400),TAGS(5,5),SUMX(5),SUMXS(5)
3806  INTEGER ARGS
3807  EQUIVALENCE (IVAL,VAL)
3808  DATA IBLNK/' '/
3809  %
3810  ENTRY MEANI(J)
3811  K=NARGS(J)
3812  DO 11 I=1,K
3813  SUMX(I)=0.
3814  VMAX(I)=-1.E30
3815  NSUM(I)=0
3816  VMIN(I)=1.E30
3817  11 SUMXS(I)=0.
3818  RETURN
3819  %
3820  ENTRY MEANB(J)
3821  K=NARGS(J)
3822  DO 21 I=1,K
3823  IVAL=IREC(ARGS(I+1,J))
3824  IF(IVAL.EQ.IBLNK) GO TO 21
3825  NSUM(I)=NSUM(I)+1
3826  VALUE=IVAL
3827  IF(ITYPE(ARGS(I+1,J)).EQ.2) VALUE=VAL
3828  IF(VALUE.LT.VMIN(I)) VMIN(I)=VALUE
3829  IF(VALUE.GT.VMAX(I)) VMAX(I)=VALUE
3830  SUMX(I)=SUMX(I)+VALUE
3831  SUMXS(I)=SUMXS(I)+VALUE*VALUE
3832  21 CONTINUE
3833  RETURN
3834  %
3835  ENTRY MEANO(J)
3836  K=NARGS(J)
3837  DO 40 I=1,K
3838  IF(NSUM(I).EQ.0) GO TO 33
3839  PRINT 31,TAGS(I,J),NSUM(I)
3840  31 FORMAT(/' MEAN STATISTICS FOR ',A8,'WITH',I6,' ITEM(S).')
3841  AMEAN=SUMX(I)/NSUM(I)
3842  RMS=SUMXS(I)/NSUM(I)
3843  IF(ITYPE(ARGS(I+1,J)).EQ.2) GO TO 32
3844  MIN=VMIN(I)
3845  MAX=VMAX(I)
3846  PRINT 35,MIN,MAX,AMEAN,RMS,SUMX(I),SUMXS(I)
3847  GO TO 34
3848  33 PRINT 36,TAGS(I,J)
3849  36 FORMAT(/' NO VALUES PRESENT FOR ',A8)
3850  GO TO 40
3851  32 PRINT 35,VMIN(I),VMAX(I),AMEAN,RMS,SUMX(I),SUMXS(I)
3852  35 FORMAT('  MIN=',F9.2,'  MAX=',F9.2,'  MEAN=',F9.2,%
```

```
3853     '  ROOT MEAN SQ.=',1PG9.2/%
3854     '  SUM=',0PF12.2,'         SUM OF SQUARES=',1PG12.3)
3855   34 IF(NSUM(I).LE.1) GO TO 40
3856   V=(SUMXS(I)-SUMX(I)*SUMX(I)/NSUM(I))/(NSUM(I)-1)
3857   SD=0.
3858   IF(V.GT.0.) SD=SQRT(V)
3859   PRINT 37,SD,V
3860   37 FORMAT('  STD. DEV.=',1PG13.5,'  VARIANCE=',G13.5)
3861   40 CONTINUE
3862   RETURN
3863   END
```

```
1800   SUBROUTINE START
1801   COMMON /FILNAM/ MASTER,MASK,DEFTN,DFILE,NUMF,IPACK
1802   DOUBLE PRECISION MASTER(10),MASK(10),DEFTN(10),DFILE(10)
1803   COMMON NAMES,ITYPE,IPTS,IDIM,VNAMES
1804   DOUBLE PRECISION NAMES(400),VNAMES(400),CONTNT(5)
1805   DIMENSION ITYPE(400),IPTS(400),IPACK(10)
1806   IDIM=400
1807   CALL OBEY('EQUATE 8 FILES',4)
1808   NUMF=1
1809   PRINT 1
1810   1 FORMAT(' WELCOME TO THE USGS "PACER" VERSION OF THE'%
1811    ' "GRASP" RETRIEVAL SYSTEM.'/' THE FOLLOWING DATA BASES'%
1812    ' ARE AVAILABLE:')
1813   10 READ(8,11,END=20) MASTER(NUMF),CONTNT,MASK(NUMF),DEFTN(NUMF),%
1814    DFILE(NUMF),IPACK(NUMF)
1815   11 FORMAT(A6,1X,5A8,3(1X,A6),I4)
1816   PRINT 12,MASTER(NUMF),CONTNT
1817   12 FORMAT(/1X,A7,'- ',5A8)
1818   NUMF=NUMF+1
1819   GO TO 10
1820   20 NUMF=NUMF-1
1821   REWIND 8
1822   PRINT 2
1823   2 FORMAT(/' THE USER MUST SPECIFY ONE OF THE ABOVE DATA BASES'   %
1824     ' FOLLOWING THE PROMPT.'//' A CHANGE OF DATA BASES CAN BE MADE'%
1825     ' AT ANY TIME BY ENTERING THE WORD "FILE"'/' FOLLOWING THE'    %
1826     ' PROMPT TO "ENTER COMMAND."'/)
1827       CALL FILE($20)
1828       RETURN
1829       END
```

```
100   INTEGER WORDS(13),COMAND,NAMEPT(26),RCODE(26),IVAL(26),POLISH(60)
101   REAL*8 NAMES(400),IFILES(20),OFILES(20),DFAULT,FILEID
102   COMMON NAMES,ITYPE,IPTS,IDIM
103   DIMENSION ITYPE(400),IPTS(400),IMAGE(4)
104   DATA DFAULT/'MASTER'/,MINE/'UPDA'/
105   DATA WORDS/'COND','LOGI','SEAR','LIST','FILE','QUIT','NAME',%
106    'HELP','REVI','DUMP','FUNC','MERG','TABL'/
107   NFILES=0
108   LPS=0
109   CALL OBEY('FORM WIDTH:80',4)
110   CALL START
111   2 PRINT 801
112   801 FORMAT(//)
113   READ(5,1,END=999,PROMPT='ENTER COMMAND: ') IMAGE
114   1 FORMAT(4A1)
115   ENCODE(COMAND,1) IMAGE
116   DO 6 I=1,11
117   IF(COMAND.EQ.WORDS(I)) GO TO 7
118   6 CONTINUE
119   IF(COMAND.EQ.MINE) GO TO 800
120   PRINT 3,COMAND
121   3 FORMAT(1X,A4,' ILLEGAL COMMAND. ENTER HELP IF YOU WISH TO SEE',%
122    ' THE LEGAL COMMANDS.'/)
123   GO TO 2
124   7 GO TO (5,10,15,610,400,999,700,500,450,550,300,810,820),I
125   5 CALL READER(NAMEPT,RCODE,IVAL,NREXP,&999)
126   GO TO 2
127   10 CALL PPARSE(POLISH,LPS,NREXP,&999)
128   GO TO 2
129   15 CALL RETRVE(IFILES,OFILES,NFILES,POLISH,LPS,NAMEPT,RCODE,IVAL%
130    ,NREXP,&10,&999)
131   GO TO 2
132   300 FILEID=DFAULT
133   IF(NFILES.GT.0) FILEID=OFILES(NFILES)
134   CALL FTNC(FILEID,&999)
135   GO TO 2
136   400 CALL FILE(&2)
137   GO TO 2
138   450 CALL CONDS(NREXP,LPS)
139   IF(NFILES.GT.0) GO TO 410
140   PRINT 409
141   409 FORMAT(' NO FILES HAVE BEEN USED AT THIS TIME.')
142   GO TO 2
143   410 PRINT 401,(IFILES(I),OFILES(I),I=1,NFILES)
144   401 FORMAT(' INPUT:    OUTPUT:'/(2X,A8,2X,A8/))
145   GO TO 2
146   500 CALL HELP(WORDS)
147   GO TO 2
148   550 CALL DUMPIT
149   GO TO 2
150   700 CALL NAME(&999)
151   GO TO 2
152   610 CALL LIST(&999)
```

```
153   GO TO 2
154   800 CALL REVISE(&999)
155   GO TO 2
156   810   CALL MERGE
157         GO TO 2
158   820   CALL TABLE
159         GO TO 2
160   999   CALL QUIT (OFILES,NFILES)
161         STOP
162         END
```

APPENDIX C, PROGRAM LISTINGS OF PACER SUBROUTINES

SUBROUTINE REVISE

Subroutine REVISE is the controlling subroutine for the five available updating procedures, and a listing may be found following this discussion. This subroutine initializes the needed control parameters, calls the updating subroutines, and posts the updated subfile onto the master file upon conclusion of the update procedure.

The unlabeled COMMON statement (line number 153) is used to transmit the data values for ITYPE, VLIST, NELEM, and FILENM (as defined in the listing) from the original GRASP program logic for use by the updating subroutines. The remaining parameters in this unlabeled COMMON are not used by any of the updating subroutines and are, therefore, treated simply as dummy variables.

The labeled COMMON statement, LOG (line 154) is used to transmit data between the updating subroutines. The variables in this statement are defined in the first part of listing except for MONTH, IDAY, and IYEAR which come from the INFONET subroutine MMDDYY (line 208) for obtaining the date of execution from the computer clock.

The DATA statements (lines 155–159) are used to initialize data values for QQ, NO, IYES, DROP, QUIT, K1, and K2. The array QQ is used to store part of the alphanumeric data string for the dynamic assignment of the update subfile to file number 21. The data values for IYES and NO are used for alphanumeric comparisons of user responses to computer prompts. The alphanumeric initialization of DROP is used for the dynamic release of a previously assigned file. The data for QUIT is used for comparison with the alphanumeric quit command. The data for arrays K1 and K2 are used to denote the maximum number of computer words reserved for each of the data elements. Array K1 is substituted into array KWORD when the WCOAL and ECOAL master files are being used, and array K2 is placed into KWORD for the USALYT master file. If another master file is added to the PACER system, it must have an appropriate array defining the number of computer words reserved for each data element available for substitution into KWORD.

The instruction-suppression code is input (line 160) and tested (line 162), followed by the output of the instructions (lines 163–198), if not suppressed by the code.

The input master-file name is tested against the available selection of master files (lines 200–202) to select the corresponding value of KDIC. After the value of KDIC is selected, the corresponding array defining the file structure is loaded into KWORD. Again, if new master files are added, the FORTRAN code must be modified to permit the definition of another KDIC value as well as to permit the transfer of the related file-structure word-length array into KWORD. The value of KDIC is also used for a keyed read of the dictionary file (line 207) to obtain the total number of records, MAXREC, residing in the master file.

Subroutine MMDDYY (line 208) is a subroutine in the INFONET library which is used to read the month, day, and year from the computer clock. This is combined with the data manager identification number, IDM (line 210), to construct the log number, ILOG (line 212). This log number is placed into the master-file ID number when a record has been deleted, or when a record has been added to the master file with the ID field left blank.

The user is then prompted to select an update procedure, IREV (line 214), with the test for zero (line 215) to exit from the update procedure and the test for the ADDREC procedure (line 216) to skip unnecessary file management. File management for modification of the subfile with any of update procedures 2 through 5 begins with a prompt for the subfile name (lines 219–220) and a test for the characters QUIT (line 221), which will cause a return from the REVISE subroutine. The remaining lines (lines 222–227) encode the file name with the rest of the alphanumeric string, ALFA, to equate the subfile name to file 21. This equivalence is performed dynamically during execution through the INFONET-dependent subroutine, OBEY (line 225). The INFONET-dependent subroutine, UPDATE, is called (line 227) to permit keyed records to be written onto file 21 without disturbing the other records residing on that file. The preceding REWIND statement (line 226) is used to ensure that the file is properly closed before it is opened in the update mode.

The OBEY subroutine is called (line 228) to equivalence file 9 with the dictionary file, USCDIC. This dictionary file is used with the three existing master files of the NCRDS. If other dictionary files were to be required with the addition of other master files, the appropriate modifications to the source code would be required.

The OBEY subroutine is also used to equivalence

file 20 with the master-file name selected following the response to the FILE command (lines 229–231). This permits the record changes and additions to be posted to the proper master file. If new master files are added to the data base, a similar statement must be added to this part of the logic. File 20 is then rewound (line 232) so that it can be placed in the update mode without conflict with prior master-file activity. Subroutine UPDATE (line 233) is an INFONET-supplied subroutine which permits the writing of records onto file 20 without the destruction of all data previously residing on that file. If the file is not placed in the update mode, the write operation creates a new file having the same name but consisting only of those records written onto the file during program execution. The execution of read operations is also permitted on this file while it is in the update mode.

The selection of an update procedure (lines 234–238) is based on the value of IREV entered for the prompted read (line 214). The $100 parameter returns execution to statement 100 which initiates the prompt for a new update procedure, while $200 shifts control to statement 200 which rewinds file 21.

The posting of the revised subfile onto the master file is accomplished by a series of prompt commands and write statements (lines 240–267). Subroutine GETKEY (lines 259 and 264) is used to obtain the record key for the data read from the subfile on file 21 (lines 258 or 263). The tests for MAXREC (lines 260 and 265) are used to update the maximum record key number so that it will be greater than or equal to the largest key valve for all records added to the master file. The next block of statements (lines 268–271) closes the dictionary file, places it in the update mode, writes MAXREC onto the line with the same key value as KDIC, and closes the file once again to the read-only mode. After the subfile has been posted onto the master file, the master file is also closed by rewinding file 20 (line 273).

Of particular importance to the programmer are the modifications to this program which must be made when: (1) the program is to be adapted for other data bases; (2) the program is to be adapted to other computer systems. These changes are noted in this documentation to enable the programmer to modify these subroutines to suit his available computer system and the different data-base structures.

A change in the data bases will require the addition of an array for storing the maximum number of computer words reserved for each raw-data element. This data array should be similar to the arrays K1 and K2 (lines 158–159). The master-file index, KDIC, must be defined in a manner similar to that in lines 200–202. The logic for specifying KWORD may be encoded as shown in lines 203–206. A MAXREC value specifying the maximum record key used in the master file must be saved in some manner. In this listing, it is stored in the KDIC-valued line of the dictionary file, read in (line 207) for test and update, and then rewritten (line 270) over its original position. The master-file name must be equivalenced to device 20; this is achieved by the OBEY subroutine in lines 229–231.

The INFONET system-dependent subroutines used for dynamic file management and requiring reprogramming for other systems are OBEY, UPDATE, and MMDDYY. The OBEY subroutine (lines 225 and 228–231) is used to equivalence the file names to the input-output device and to drop (line 280) the subfile from the computer system. The system routine, UPDATE, is used to place files in the update mode (lines 227, 233, and 269). The MMDDYY subroutine is also INFONET dependent and is used to obtain the month, day, and year value, which are used by the log number. If this software is used with a computer system other than INFONET, the system-dependent logic must be reprogrammed to make it compatible with the new computer system.

```
100   SUBROUTINE REVISE(*)
101   %     THIS IS SUBROUTINE ''REVISE.''
102   %     IT IS DESIGNED TO ADD, DELETE, OR CHANGE RECORDS
103   %     OR PORTIONS OF RECORDS BELONGING TO THE MASTER
104   %     FILE.
105   %
106   %     FILENM= THE NAME OF THE MASTER FILE.
107   %     VLIST = THE ALPHANUMERIC LIST OF THE VARIABLES
108   %             AS THEY APPEAR IN THE MASK FILE.
109   %     ITYPE = THE VARIABLE TYPE OR THE DICTIONARY
110   %             POINTER AS LISTED IN THE MASK FILE.
111   %     QQ    = DUMMY INPUT FILL FOR PLACING THE FILE
112   %             TO BE UPDATED INTO THE 'OBEY' STATEMENT.
113   %     ALFA  = REFORMATTED ALPHANUMERIC STRING FOR
114   %             PLACING THE NAME OF THE FILE TO BE
115   %             UPDATED INTO CORRESPONDING  'OBEY'
116   %             STATEMENT.
117   %     NELEM = NUMBER OF DATA ELEMENTS IN ONE
118   %             MASTER RECORD.
119   %     KWORD = AN ARRAY SPECIFYING THE MAXIMUM NUMBER OF
120   %             COMPUTER WORDS RESERVED FOR EACH RAW DATA
121   %             ELEMENT.
122   %     MAXREC= THE MAXIMUM RECORD KEY VALUE FOR THE MASTER FILE.
123   %     KDIC  = INDEX TO IDENTIFY THE MASTER FILE BEING USED AND TO
124   %             LOCATE THE PROPER MAXREC VALUE FROM THE DICTIONARY
125   %             FILE.
126   %     NREC  = NUMBER OF RECORDS REVIEWED IN THE SELECTED SUBFILE.
127   %     K1    = DATA TO BE TRANSFERRED INTO THE "KWORD" ARRAY FOR
128   %             USE WITH MASTER FILES "WCOAL" AND "ECOAL."
129   %     K2    = DATA TO BE TRANSFERRED INTO THE "KWORD" ARRAY FOR
130   %             USE WITH MASTER FILE "USALYT."
131   %     IRITE = THE INSTRUCTION SUPPRESSION CODE.  AN INPUT
132   %             VALUE OF 23 WILL SUPPRESS THE INSTRUCTION
133   %             PRINTS FROM SUBROUTINES "REVISE," "ADDREC,"
134   %             "DELREC," "SEQREV," "KEYREV," AND "BATCH."
135   %     IDM   = THE DATA MANAGER ID NUMBER (1-99).
136   %     IREV  = THE UPDATE PROCEDURE SELECTION CODE.
137   %     ILOG  = THE LOG NUMBER FOR RECORDS ADDED WITHOUT AN
138   %             ID NUMBER AND FOR RECORDS DELETED FROM THE
139   %             MASTER FILE.  THIS NUMBER IS ASSIGNED AS THE
140   %             NEW ID NUMBER.  IT IS A CODED, EIGHT DIGIT
141   %             NUMBER WITH THE TWO LEFTMOST DIGITS DESIGNATING
142   %             THE YEAR, THE NEXT TWO DIGITS DESIGNATING THE
143   %             MONTH, AND THE NEXT PAIR OF DIGITS DESIGNATING
144   %             THE DAY OF RECORD CHANGE, WITH THE FINAL PAIR
145   %             OF DIGITS DENOTING THE DATA MANAGER RESPONSIBLE
146   %             FOR THE ADDITION OR THE DELETION OF THOSE
147   %             RECORDS.
148         DIMENSION      ALFA(5),       QQ(5)
149         DIMENSION      ITYPE(400),    KWORD(400)
150         DIMENSION      DROP(4),       IDATA(400)
151         DIMENSION      K1(50),        K2(50)
152         DOUBLE PRECISION    FILENM,   VLIST(400)
```

```
      COMMON  NAMED(800),ITYPE,PAD(401),VLIST,NELEM,FILENM
      COMMON /LOG/ MONTH,IDAY,IYEAR,ILOG,MAXREC,KWORD,KDIC,IRITE
      DATA QQ/'EQUA','TE 2','1 '/, IBLANK/' '/
      DATA NO,IYES/'NO','YES'/
      DATA DROP/'DROP',3*' '/,QUIT/'QUIT'/
      DATA K1/4,6,7*1,4,6,1,4,6,1,4,5,1,4,7,1,4,5,1,3,1,1/
      DATA K2/4,6,7*1,4,6,4,6,4,5,4,7,3,5,1,3,25*1/
      READ(5,1020,PROMPT='PLEASE PRESS CARRIAGE RETURN. ') IRITE
 1020 FORMAT(I2)
      IF(IRITE .EQ. 23) GO TO 8
      WRITE(6,1000)
 1000 FORMAT(/' THIS IS "UPDATE."'/' IT IS DESIGNED TO PERMIT THE'%
     ' USER TO ADD RECORDS TO OR DELETE RECORDS FROM THE'/       %
     ' MASTER FILE, OR TO CHANGE RECORDS OR PORTIONS OF RECORDS' %
     ' IN A SUBFILE OF THE'/' MASTER FILE AND TO POST THESE'     %
     ' CHANGES ONTO THE MASTER FILE.'//' THE FIVE PROCEDURES'    %
     ' USED TO UPDATE THE MASTER FILE ARE:'//' 1.  THE ADDITION' %
     ' OF NEW RECORDS ALREADY WRITTEN INTO A TEMPORARY FILE.'    %
     '  UNLIKE'/'     THE OTHER UPDATE PROCEDURES WHICH OPERATE' %
     ' ON RECORDS IN THE MASTER FILE'/'     FORMAT, THIS'        %
     ' PROCEDURE OPERATES ON RAW DATA RECORDS AND CONVERTS'      %
     ' THESE'/'     RECORDS TO THE MASTER FILE FORMAT FOR'       %
     ' INSERTION INTO THE MASTER FILE.'//' 2.  THE DELETION, BY' %
     ' KEY, OF RECORDS ALREADY EXISTING IN THE MASTER FILE.'/    )
      WRITE(6,1003)
 1003 FORMAT(     %
     ' 3.  THE SEQUENTIAL REVISION OF RECORDS FROM A SELECTED'   %
     ' SUBFILE (I.E. SELECTED'/'     THROUGH A LOGICAL SEARCH)'  %
     ' FROM THE MASTER FILE.  THIS SUBFILE MAY THEN BE'/'    '   %
     ' POSTED ONTO THE MASTER FILE AFTER THE DESIRED REVISIONS'  %
     ' ARE COMPLETED.'//' 4.  THE SELECTION, BY KEY, OF'         %
     ' ANY RECORD BELONGING TO THE SUBFILE FOR REVISION'/'    '  %
     ' OF ANY SELECTED DATA ELEMENT.  THE DATA MANAGER HAS THE'  %
     ' OPTION OF POSTING'/'     THE SELECTED RECORD ONTO THE'    %
     ' MASTER FILE OR LEAVING IT, AS REVISED, IN THE'/'    '     %
     ' SUBFILE FOR FURTHER REVISION.'//' 5.  THE BATCH REVISION' %
     ' OF A GIVEN DATA ELEMENT WHICH WILL BE THE SAME VALUE FOR' %
     '     ALL RECORDS IN THE SELECTED SUBFILE.'//' WHEN'        %
     ' REVISION HAS BEEN COMPLETED ON ANY SELECTED SUBFILE, THE' %
     ' USER MAY THEN'/' ELECT TO POST THE REVISED SUBFILE ONTO'  %
     ' THE MASTER FILE, OR SAVE THE SUBFILE FOR'/' REVIEW AND'   %
     ' POSSIBLE FURTHER REVISION BY ANY OF THE 2-5 UPDATE'       %
     ' PROCEDURES.'/' RECORDS DELETED FROM THE MASTER FILE ARE'  %
     ' SAVED IN A SPECIAL "SAVE DELETION"'/' FILE FOR FUTURE'    %
     ' RECOVERY IF THERE SHOULD ARISE A NEED TO RECONSTITUTE'    %
     ' THESE'/' RECORDS.'//)
 8    KDIC  =  0
      IF(FILENM .EQ. 'WCOAL') KDIC = 1
      IF(FILENM .EQ. 'USALYT') KDIC = 2
      IF(FILENM .EQ. 'ECOAL') KDIC = 3
      DO 9 I=1,NELEM
      IF(KDIC .EQ. 1)  KWORD(I) = K1(I)
      IF(KDIC .EQ. 3)  KWORD(I) = K1(I)
 9    IF(KDIC .EQ. 2)  KWORD(I) = K2(I)
```

```
      READ(9,KEY=KDIC) MAXREC
      CALL MMDDYY(MONTH,IDAY,IYEAR)
15    CONTINUE
      READ(5,1010,ERR=15,PROMPT='DATA MANAGER ID NO. (1-99): ') IDM
1010  FORMAT(I2)
      ILOG  =  1000000*IYEAR + 10000*MONTH + 100*IDAY + IDM
100   CONTINUE
      READ(5,*,PROMPT='UPDATE PROCEDURE (1-5): ') IREV
      IF(IREV .EQ. 0) RETURN
      IF(IREV .EQ. 1) GO TO 16
      DO 10 I=1,2
10    QQ(I+3)  =  IBLANK
      READ(5,1001,PROMPT='DATA FILE TO BE REVISED: ')%
      QQ(4),QQ(5)
      IF(QQ(4) .EQ. QUIT) RETURN
1001  FORMAT(2A4)
      ENCODE(ALFA,1002,ERR=998) QQ
1002  FORMAT(2A4,A2,2A4)
      CALL OBEY(ALFA,5)
      REWIND 21
      CALL UPDATE(21,1)
16    CALL OBEY('EQUATE 9 USCDIC',4)
      IF(KDIC .EQ. 1) CALL OBEY('EQUATE 20 WCOAL',4)
      IF(KDIC .EQ. 2) CALL OBEY('EQUATE 20 USALYT',4)
      IF(KDIC .EQ. 3) CALL OBEY('EQUATE 20 ECOAL',4)
      REWIND 20
      CALL UPDATE(20,1)
      IF(IREV .EQ. 1) CALL ADDREC($200,$100,NREC,QQ(4),QQ(5))
      IF(IREV .EQ. 2) CALL DELREC($100)
      IF(IREV .EQ. 3) CALL SEQREV(NREC)
      IF(IREV .EQ. 4) CALL KEYREV($100)
      IF(IREV .EQ. 5) CALL BATCH($200,NREC)
200   REWIND 21
      WRITE(6,1004) QQ(4),QQ(5)
1004  FORMAT(/' FILE  "',A4,A2,'"  HAS BEEN REVISED.'/%
      ' DO YOU WISH TO WRITE THIS FILE ONTO THE MASTER%
 FILE?')
18    READ(5,1005,PROMPT='"YES" OR "NO": ') ITEST
1005  FORMAT(A4)
      IF((ITEST .NE. IYES) .AND. (ITEST .NE. NO)) GO TO 18
      IF(ITEST .EQ. NO) GO TO 100
%
% 'IDATA' IS USED AS A DUMMY VARIABLE FOR READ AND WRITE
% OPERATIONS.
      REWIND 21
      WRITE(6,1007)
1007  FORMAT(/' DO YOU WANT TO POST THE ENTIRE SUBFILE?')
19    READ(5,1005,PROMPT='"YES" OR "NO": ') ITEST
      IF((ITEST .NE. IYES) .AND. (ITEST .NE. NO)) GO TO 19
      IF(ITEST .EQ. IYES) GO TO 21
      DO 20 I=1,NREC
      READ(21) (IDATA(K),K=1,NELEM)
      CALL GETKEY(21,NKEY)
      IF(NKEY .GT. MAXREC) MAXREC = NKEY
```

```
20    WRITE(20,KEY=NKEY) (IDATA(K),K=1,NELEM)
      GO TO 22
21    READ(21,END=22) (IDATA(K),K=1,NELEM)
      CALL GETKEY(21,NKEY)
      IF(NKEY .GT. MAXREC) MAXREC = NKEY
      WRITE(20,KEY=NKEY) (IDATA(K),K=1,NELEM)
      GO TO 21
22    REWIND 9
      CALL UPDATE(9,1)
      WRITE(9,KEY=KDIC) MAXREC
      REWIND 9
      WRITE(6,1009)
      REWIND 20
1009  FORMAT(/' DO YOU WISH TO SAVE THIS FILE?')
25    READ(5,1005,PROMPT='"YES" OR "NO": ') ITEST
      IF((ITEST .NE. IYES) .AND. (ITEST .NE. NO)) GO TO 25
      DROP(3)  =  QQ(4)
      DROP(4)  =  QQ(5)
      REWIND 21
      IF(ITEST .EQ. NO) CALL OBEY(DROP,4)
      GO TO 500
998   WRITE(6,1006)
1006  FORMAT(/' NAME OF FILE HAS BEEN IMPROPERLY ENTERED.')
      GO TO 100
500   WRITE(6,1008)
1008  FORMAT(/' UPDATE OPERATIONS HAVE BEEN COMPLETED.'/%
     ' IF YOU WISH TO CONTINUE WITH THE UPDATE PROCEDURE,'      %
     ' ENTER THE NUMBER (1-5).'/' ENTERING A "0" FOR THE'      %
     ' PROCEDURE PROMPT OR "QUIT" FOR THE FILE PROMPT WILL'/   %
     ' RETURN CONTROL TO THE SEARCH AND RETRIEVAL PORTION OF'  %
     ' "PACER."'/)
      GO TO 100
      END
```

SUBROUTINE ADDREC

Subroutine ADDREC is designed to read an unkeyed raw-data file, search the dictionaries for matching alphanumeric entries, prompt the user to add the nonmatching terms to the dictionary, convert the alphanumeric data values to master-file dictionary pointers, and write onto a second file a keyed, PACER-compatible record which is suitable for posting onto the master file. A listing of subroutine ADDREC appears at the end of the discussion. The subroutine listing contains a brief description (lines 102–107) followed by a definition list of the major variables (lines 109–167). The unlabeled COMMON and the LOG COMMON (lines 175–176) are the same as in subroutine REVISE. The QQIN array is used to equivalence the raw-data file to file 31 whereas the QQOUT array is used to equivalence file 21 to the translated subfile that will then be posted onto the master file. The alphanumeric data strings, IBLANK, QBLANK, YES, XNO, and QUIT, are used for comparison of prompted user responses at various decision points throughout the logic. The NODATA array is used for entering the alphanumeric string NO DATA ENTERED into those alphanumeric data fields for which no data value exists.

If the value of IRITE was entered as any value other than 23 (line 182), the instruction message will be printed (lines 183–194). Then, the user is prompted to specify the name of his raw-data file (lines 195–196). If the characters QUIT were entered, the record addition procedure will be terminated and after the message (lines 307–312) is printed, file 31 is rewound and control is returned to REVISE (lines 313–314). The user is then prompted to specify the name of the translated file (lines 198–199). This file is transferred back to subroutine REVISE and becomes the subfile, which is then posted onto the master file. The file name is stored in variables QQ1 and QQ2 (lines 201–202) for this purpose. Again, if the characters, QUIT, were entered for the file name (line 200), the record addition procedure is terminated.

The QQIN and QQOUT arrays are then encoded into the ALFIN and ALFOUT arrays (lines 203–205) for use by the OBEY subroutine to equivalence the file names to the appropriate unit numbers. The key register, KEYR, is then set equal to the maximum record key value, MAXREC, in the master file. The key register is then incremented by 1 for each translated record which is posted onto the master file. Thus, all new record additions will be added to the end of the master file. The total number of computer words per record, KWT, is determined by summing over all the KWORD values for the given number of data elements, NELEM, belonging to that master file (lines 209–211). This value of KWT is transmitted to the formatted output routines (lines 299–304) to provide the length of the total raw-data string.

The user response to the prompting statements (lines 212–215) establishes the alphanumeric value for REVIZ which is tested for a YES value before each individual record is displayed. A test is made to ensure that the only alphanumeric value entered is either a YES or a NO (line 216); otherwise, the user is again prompted for a YES or NO response.

The counter for the number of records reviewed, NREC, is then initialized to zero (line 217), as is the counter for number of computer words required for the first through the I-th data elements, NWORDS (line 218). Program control returns to statement 100 each time a new record is to be read from the raw-data file. The reading of each record from the raw-data file is accomplished by a specially formatted subroutine selected by testing the value of KDIC (lines 219–221). Following the successful reading of the next record from the raw-data file (that is, if the EOF mark has not been encountered by the read operation), the value of NREC is incremented by 1 (line 222).

The logic to translate the raw-data record into a record compatible with a PACER master file is contained in the DO loop extending to statement 99 (lines 223–295). The DO loop index ranges from 1 to the number of data elements, NELEM. The number of computer words required to store the data string for the first "I" data elements is obtained by incrementing NWORDS by the number of words, KWORD, reserved for the data string of the I-th data element (line 224). Next the data type value, ITYPE, is tested (line 225) to determine whether the data element is an integer number, real number, or an alphanumeric string. If the data is integer, a transfer is made to statement 10; if it is real, a transfer is made to statement 20; and if it is alphanumeric, a transfer is made to statement 30.

At statement 10, the integer value of the raw data is simply transferred from the raw-data-array element, NSTRNG, to the I-th position in the master-record array, IMAGE (line 229). In the present application, the LOG value is substituted into the ID position of the master-record array, if no value has previously been entered (lines 230–235). Then,

transfer is made (line 236) to the end of the DO loop, where the value of I is incremented by 1, and the procedure is repeated for the next data element of the record.

At statement 20, the real value of the data is transferred from the data-array element, STRING, to the I-th position in the master-record array, QIMAGE (line 240), program control then jumping (line 241) to the end of the DO loop. It should be noted at this point that the arrays, IMAGE and QIMAGE, are equivalenced, as are the arrays, NSTRNG and STRING, although in each case one array is typed as integer and the other array is typed as real. The equivalencing of these arrays permits data to be read from or written to a data file through the use of only one of the equivalenced arrays.

Beginning at statement 30, the alphanumeric data string is matched to a dictionary entry, and an integer value, which indexes or "points" to that diction element, is placed in the I-th data position of the IMAGE array. The number of computer words required for the I-th data element is stored in KEND (line 250), the number of computer words required for the preceding I-th data elements is stored in KBEGIN (line 251), and the value of KP1 is defined to be one greater than KBEGIN (line 252). Special logic has been inserted (lines 252–254) to replace blank alphanumeric data in elements 5 and 7 with a dash. Then, all seven words of the NAME array are blanked (lines 255–256), and the first LT nonblank values of the NSTRING array are counted and substituted into the NAME array (lines 257–260). If the number of nonblank words, LT, is greater than zero, the logic jumps to statement 54 (line 261); otherwise, it inserts the alphanumeric data, NO DATA ENTERED, into the NAME array and also into the NSTRNG array (lines 262–266).

The ITYPE value for alphanumeric data is the dictionary number of the I-th data element plus 10. Therefore, the dictionary number, NDIC, can be found by subtracting 10 from ITYPE. This value, the alphanumeric string in array, NAME, and the length of that string, LT, are each placed into the calling sequence of subroutine MATCH (line 268) which searches for a matching dictionary entry and the record key, MKEY, for that entry. If a new entry made from the terminal contains only blank data, control will be returned to statement 995. If a dictionary match is found, the pointer to this entry is translated into the I-th element of the IMAGE array (line 269), and control is passed (line 270) to the end of the DO loop to recycle and process the next data element.

If there is no dictionary match, control is transferred to statement 994, where a message is displayed on the terminal asking the user if he wants the alphanumeric string entered into the dictionary. Checks are made to ensure that the response is either YES or NO (lines 271–278). If the response is YES, the number of entries, NUM (returned from subroutine MATCH), contained in that dictionary category is increased by 1 (line 279). Then subroutine ADDICT is called (line 280) to write the entry into the dictionary and to update the value of NUM stored at the beginning of that dictionary category.

The value of NUM is also stored in the I-th position of IMAGE as the new dictionary pointer (line 281), and control is again transferred to the end of the DO loop (line 282).

If the user does not wish to have the nonmatching alphanumeric data entered into the dictionary, he is then prompted to re-enter the alphanumeric string (lines 283–288). The alphanumeric string is also transferred into NSTRING (lines 289–290), and control is transferred (line 291) to statement 50, where the matching procedure is executed once more.

If, in subroutine MATCH, it is determined that the data string is blank, a message is written (lines 292 and 293), and control is transferred (line 294) to statement 60 for re-entry of the data (line 283).

Once the DO loop has been completed, the input value of the record key, KEYP, is tested to see if it has been entered (line 296). If it has not, KEYR, which was initialized as the value of MAXREC, is incremented by 1. Next, KEYA is set equal to KEYR (line 297). If KEYP is greater than zero, KEYA is instead set to the value of KEYP (line 298). This is to ensure that, if the records to be added were once a part of the master file and the record key is known, then they would be written into the same keyed location.

If a display of each record has been requested, these records will be output through a specially formatted subroutine which is dependent upon the data structure (lines 299–304).

Finally, the translated record is written onto unit 21 (line 305), and control is transferred (line 306) to statement 100 to continue with the processing of the next record in the raw-data file.

When all records have been processed, a message is displayed stating that the file is ready for posting (lines 307–312). The raw-data file is rewound (line 313), and control is returned to REVISE (line 314).

The ADDREC logic that is data-base dependent appears in: lines 219–222, where a particular input

routine is selected based upon a specific index in the master file; lines 230–235, where the log number is substituted for blank data of the ID number in the given master-file record; lines 253–254, where dashes are substituted for blank entry of the directions for township and range in all master files; and lines 299–304, where the special output display subroutines are dependent upon the data structure of each of the master files.

The INFONET-dependent OBEY subroutine is used twice in ADDREC. The first time (line 206) it is used to equivalence the raw-data file name to unit 31, and the second time (line 207) to equivalence the translated output file to unit 21.

```
100   SUBROUTINE ADDREC(*,*,NREC,QQ1,QQ2)
101   %
102   %  SUBROUTINE "ADDREC" IS DESIGNED TO READ AN UNKEYED, DATA
103   %  FILE, SEARCH THE DICTIONARIES TO ENSURE CORRECT ALPHANUMERIC
104   %  INPUT, CHANGE THE ALPHANUMERIC DATA VALUES TO DICTIONARY
105   %  POINTERS AS USED IN "PACER," AND WRITE A KEYED, "PACER"
106   %  COMPATIBLE RECORD SUITABLE FOR POSTING ONTO THE MASTER
107   %  FILE.
108   %
109   %     NELEM  = NUMBER OF DATA ELEMENTS IN A MASTER RECORD.
110   %     VLIST  = DATA ELEMENT NAMES AS THEY APPEAR IN THE
111   %              DEFINITION AND MASK FILES.
112   %     ITYPE  = TYPE DESIGNATION OF THE DATA ELEMENTS AS
113   %              LISTED IN THE MASK FILE.
114   %              ITYPE = 1...INTEGER TYPE DATA
115   %              ITYPE = 2...REAL TYPE DATA
116   %              ITYPE > 10...DATA CONSISTS OF A DICTIONARY
117   %                POINTER TO REFERENCE A LOOKUP IN THE
118   %                (ITYPE - 10)TH DICTIONARY.
119   %     NREC   = NUMBER OF RECORDS REVIEWED IN THE SELECTED SUBFILE.
120   %     KDIC   = INDEX TO IDENTIFY THE MASTER FILE BEING USED AND TO
121   %              LOCATE THE PROPER MAXREC VALUE FROM THE DICTIONARY
122   %              FILE.
123   %     IMAGE  = INTEGER REPRESENTATION OF MASTER DATA
124   %              ELEMENTS, INCLUDING INTEGER DATA AND
125   %              DICTIONARY POINTERS.
126   %     QIMAGE = REAL REPRESENTATION OF MASTER DATA ELEMENTS
127   %              THROUGH EQUIVALENCE WITH 'IMAGE.'
128   %     NSTRNG = INTEGER REPRESENTATION OF THE DATA STRING FOR
129   %              ALL DATA ELEMENTS (INTEGER & ALPHANUMERIC)
130   %              BELONGING TO A MASTER RECORD.
131   %     STRING = REAL REPRESENTATION (THROUGH EQUIVALENCE
132   %              STATEMENT) OF THE DATA STRING FOR ALL REAL-
133   %              VALUED DATA ELEMENTS BELONGING TO A
134   %              MASTER RECORD.
135   %     NAME   = ARRAY (UP TO 7 COMPUTER WORDS) OF NEW ALPHA-
136   %              NUMERIC DATA FOR COMPARISON WITH EXISTING
137   %              DICTIONARY ENTRIES TO  1) DETERMINE DICT-
138   %              IONARY POINTER OR  2) TO DETERMINE THAT A
139   %              DICTIONARY ENTRY DOES NOT EXIST FOR THE
140   %              GIVEN DICTIONARY CATEGORY.
141   %     LT     = NUMBER OF WORDS (LESS THAN OR EQUAL TO 7)
142   %              NEEDED TO STORE THE ALPHANUMERIC ENTRY IN
143   %              "NAME."
144   %     KWORD  = AN ARRAY SPECIFYING THE MAXIMUM NUMBER OF
145   %              COMPUTER WORDS RESERVED FOR EACH RAW DATA
146   %              ELEMENT.
147   %     NWORDS = THE TOTAL MAXIMUM NUMBER OF COMPUTER WORDS
148   %              NEEDED TO STORE RAW DATA FOR THE 1ST THROUGH
149   %              THE I-TH DATA ELEMENTS.
150   %     KBEGIN = THE TOTAL MAXIMUM NUMBER OF WORDS NEEDED TO
151   %              STORE RAW DATA FOR THE 1ST THROUGH THE
152   %              (I-1)TH DATA ELEMENTS.
```

```
C     KEND   = THE MAXIMUM NUMBER OF WORDS NEEDED TO STORE
C              THE I-TH DATA ELEMENT.  THIS UNSUBSCRIPTED
C              VARIABLE IS EQUIVALENT TO KWORD(I).
C     NDIC   = DICTIONARY CATEGORY REFERENCE NUMBER.
C     NUM    = TOTAL NUMBER OF DICTIONARY ENTRIES IN THE
C              "NDIC" DICTIONARY CATEGORY.
C     MAXREC = MAXIMUM RECORD KEY IN THE MASTER FILE.
C     KEYR   = KEY NUMBER FOR RECORDS TO BE ADDED FOLLOWING
C              THE LAST RECORD ("MAXREC") IN THE MASTER FILE.
C     KWT    = TOTAL OF ALL "NELEM" "KWORD"S.
C     ALFIN  = DUMMY ARRAY TO DESIGNATE INPUT FILE NAME.
C     ALFOUT = DUMMY ARRAY TO DESIGNATE NAME OF TRANSLATED
C              FILE TO BE POSTED TO THE MASTER FILE.
C     QQIN   = DUMMY ARRAY TO BE ENCODED INTO "ALFIN."
C     QQOUT  = DUMMY ARRAY TO BE ENCODED INTO "ALFOUT."
C
      DIMENSION      IMAGE(400),     NSTRNG(400),  %
      ITYPE(400),    QIMAGE(400),    STRING(400),  %
      KWORD(400),    NAME(7),        VLIST(400),   %
      ALFIN(5),      ALFOUT(5),      QQIN(5),      %
      QQOUT(5),      NODATA(4)
      DOUBLE PRECISION               VLIST
      COMMON NAMED(800),ITYPE,PAD(401),VLIST,NELEM
      COMMON /LOG/ MONTH,IDAY,IYEAR,ILOG,MAXREC,KWORD,KDIC,IRITE
      EQUIVALENCE (IMAGE,QIMAGE),(NSTRNG,STRING)
      DATA IBLANK,QBLANK,YES,XNO/' ',' ','YES','NO'/
      DATA QQIN/'EQUA','TE ','31 '/,QQOUT/'EQUA','TE ','21 '/
      DATA QUIT/'QUIT'/
      DATA  NODATA/'NO D','ATA ','ENTE','RED '/
      IF(IRITE .EQ. 23) GO TO 1
      WRITE(6,1009)
1009  FORMAT(/' THE ADD RECORD PROCEDURE IS DESIGNED TO READ A'   %
     ' RAW DATA INPUT FILE AND CONVERT'/' THE RECORDS TO THE'     %
     ' RECORD STRUCTURE THAT IS COMPATIBLE WITH THE MASTER FILE'/%
     ' OF "PACER," CHECKING FOR CORRECT DICTIONARY ENTRIES, AND' %
     ' PROMPTING THE DATA'/' MANAGER TO REQUEST ADDITION OF THE' %
     ' NONMATCHING DICTIONARY ENTRIES TO THE'/' DICTIONARY LIST' %
     ' OR TO CORRECT THE INPUT ENTRY SO THAT IT MATCHES A VALUE'/%
     ' ALREADY IN THE DICTIONARY LIST.'//' IF THE INPUT FILE DOES'%
     ' NOT CONTAIN RAW DATA, THE DATA MANAGER CAN EXIT THIS'/     %
     ' REVISION PROCEDURE TO SELECT A DIFFERENT PROCEDURE (3-5)' %
     ' BY ENTERING "QUIT,"'/' WHEN PROMPTED FOR A FILE NAME.'// )
1     READ(5,1002,PROMPT='NAME OF RAW DATA INPUT%
 FILE: ') QQIN(4),QQIN(5)
      IF(QQIN(4) .EQ. QUIT) GO TO 999
      READ(5,1002,PROMPT='NAME OF TRANSLATED FILE%
 FOR POSTING TO MASTER FILE: ') QQOUT(4),QQOUT(5)
      IF(QQOUT(4) .EQ. QUIT) GO TO 999
      QQ1  =  QQOUT(4)
      QQ2  =  QQOUT(5)
1003  FORMAT(A4,2A3,A4,A2)
      ENCODE(ALFIN,1003,ERR=998) QQIN
      ENCODE(ALFOUT,1003,ERR=998) QQOUT
      CALL OBEY(ALFIN,4)
```

```
207         CALL OBEY(ALFOUT,4)
208         KEYR  =  MAXREC
209         KWT  =  0
210         DO  2  I=1,NELEM
211   2     KWT  =  KWT + KWORD(I)
212         WRITE(6,1011)
213   1011  FORMAT(/' DO YOU WISH TO DISPLAY EACH RECORD IN THE '%
214         'INPUT FILE ON THE TERMINAL?')
215   5     READ(5,1001,PROMPT='"YES" OR "NO": ') REVIZ
216         IF((REVIZ .NE. YES) .AND. (REVIZ .NE. XNO)) GO TO 5
217         NREC  =  0
218   100   NWORDS  =  0
219         IF(KDIC .EQ. 1) CALL REDUSC(NSTRNG,STRING,KEYP,$999)
220         IF(KDIC .EQ. 2) CALL REDUSA(NSTRNG,STRING,KEYP,$999)
221         IF(KDIC .EQ. 3) CALL REDUSC(NSTRNG,STRING,KEYP,$999)
222         NREC  =  NREC + 1
223         DO 99 I=1,NELEM
224         NWORDS  =  NWORDS + KWORD(I)
225         IF(ITYPE(I) - 2) 10,20,30
226   %
227   %  LOAD INTEGER DATA INTO THE "PACER" RECORD FORMAT.
228   %
229   10    IMAGE(I)  =  NSTRNG(NWORDS)
230         IF (KDIC .NE. 1) GO TO 15
231         IF(I .EQ. 26 .AND. IMAGE(26) .EQ. 0) IMAGE(26) = ILOG
232   15    IF(KDIC .NE. 2) GO TO 16
233         IF(I .EQ. 23 .AND. IMAGE(23) .EQ. 0) IMAGE(23) = ILOG
234   16    IF(KDIC .NE. 3) GO TO 99
235         IF(I .EQ. 26 .AND. IMAGE(26) .EQ. 0) IMAGE(26) = ILOG
236         GO TO 99
237   %
238   %  LOAD REAL DATA INTO THE "PACER" RECORD FORMAT.
239   %
240   20    QIMAGE(I)  =  STRING(NWORDS)
241         GO TO 99
242   %
243   %  CHECK ALPHANUMERIC ENTRIES FOR DICTIONARY CONSISTENCY.
244   %  IF ENTRY DOES NOT APPEAR IN THE DICTIONARY...CHECK WITH
245   %  THE DATA MANAGER TO DETERMINE IF IT IS INTENDED THAT
246   %  THE ENTRY BE ADDED TO THE DICTIONARY.  THEN, COMPUTE
247   %  THE DICTIONARY POINTERS AND LOAD THE DATA INTO THE
248   %  "PACER" RECORD FORMAT.
249   %
250   30    KEND  =  KWORD(I)
251         KBEGIN  =  NWORDS - KWORD(I)
252         KP1  =  KBEGIN + 1
253         IF(I .EQ. 5 .AND. NSTRNG(KP1) .EQ. IBLANK)  NSTRNG(KP1) = '-'
254         IF(I .EQ. 7 .AND. NSTRNG(KP1) .EQ. IBLANK)  NSTRNG(KP1) = '-'
255   40    DO 41 K=1,7
256   41    NAME(K)  =  IBLANK
257   50    LT  =  0
258         DO 51 K=1,KEND
259         NAME(K)  =  NSTRNG(KBEGIN + K)
260   51    IF(NAME(K) .NE. IBLANK) LT = LT + 1
```

```
      IF(LT .NE. 0) GO TO 54
      DO 52 K=1,4
52    NAME(K)  =  NODATA(K)
      LT  =  4
      DO 53 K=1,KEND
53    NSTRNG(KBEGIN+K)  =  NAME(K)
54    NDIC  =  ITYPE(I) - 10
      CALL MATCH(NAME,NDIC,LT,MKFY,NUM,$994,$995)
      IMAGE(I)  =  MKEY - NDIC*10000
      GO TO 99
994   WRITE(6,2001) VLIST(I),NAME
2001  FORMAT(/' THERE IS NO DICTIONARY MATCH FOR DATA NAME: '%
     ,A7/' ALPHANUMERIC DATA: ',7A4/%
     ' DO YOU WISH TO ENTER THIS DATA IN THE DICTIONARY?')
55    READ(5,1001,PROMPT='"YES" OR "NO": ') TEST
1001  FORMAT(A4)
      IF((TEST .NE. YES) .AND. (TEST .NE. XNO)) GO TO 55
      IF(TEST .NE. YES) GO TO 60
      NUM  = NUM + 1
      CALL ADDICT(NAME,LT,NDIC,NUM)
      IMAGE(I)  =  NUM
      GO TO 99
60    WRITE(6,2002) VLIST(I)
2002  FORMAT(/' ENTER DATA VALUE FOR DATA NAME "',A7,'"')
      DO 61 L=1,7
61    NAME(L)  =  IBLANK
      READ(5,1002,PROMPT='DATA VALUE: ') NAME
1002  FORMAT(7A4)
      DO 62 K=1,KEND
62    NSTRNG(KBEGIN+K)  =  NAME(K)
      GO TO 50
995   WRITE(6,2003)
2003  FORMAT(/' BLANK DICTIONARY ENTRIES ARE NOT LEGITIMATE.')
      GO TO 60
99    CONTINUE
      IF(KEYP .LT. 1) KEYR  =  KEYR + 1
      KEYA  =  KEYR
      IF(KEYP .GT. 0) KEYA = KEYP
      IF(REVIZ .EQ. 'YES' .AND. KDIC .EQ. 1)      %
```

```
300           CALL OUTUSC(NSTRNG,STRING,KEYA,KWT)
301         IF(REVIZ .EQ. 'YES' .AND. KDIC .EQ. 2)     %
302           CALL OUTUSA(NSTRNG,STRING,KEYA,KWT)
303         IF(REVIZ .EQ. 'YES' .AND. KDIC .EQ. 3)     %
304           CALL OUTUSC(NSTRNG,STRING,KEYA,KWT)
305         WRITE(21,KEY=KEYA) (IMAGE(I),I=1,NELEM)
306         GO TO 100
307   999   WRITE(6,2005)
308   2005  FORMAT(/' THE INPUT FILE IS NOW READY FOR POSTING ONTO'/%
309         ' THE MASTER FILE.  IF YOU WISH TO MAKE FURTHER CHANGES,'/%
310         ' TO THIS FILE BEFORE IT IS POSTED, SELECT THE SEQUENTIAL,'/%
311         ' BATCH, OR KEYED REVISION PROCEDURE AND SPECIFY THE NAME'/%
312         ' OF THIS FILE.')
313         REWIND 31
314         RETURN 1
315   998   WRITE(6,1010)
316   1010  FORMAT(' NAME OF FILE HAS BEEN IMPROPERLY ENTERED.')
317         GO TO 1
318         END
```

SUBROUTINE DELREC

This subroutine is used for the deletion of the contents of a record contained in the master file. A listing of subroutine DELREC appears at the end of this discussion. The corresponding record in the reference subfile is deleted in its entirety. The position of the deleted record, however, is maintained in the master file. A deletion message is inserted in 3 of the alphanumeric data fields, and the ID number is replaced by a coded log number, specifying the date and the identification number of the data manager responsible for the deletion.

The COMMON statements contain the same variable names as are contained in the COMMON statements of subroutine REVISE. The DATA statement (line 170) initializes the variable, IBLANK, to be used for the testing of input. The message providing instructions to the user (lines 172–188) is suppressed if IRITE has been previously given the value of 23 (line 171).

The master-file index, KDIC, is tested (lines 108–194) to determine which master file will be equivalenced to file 30 and which "save" file will be equivalenced to file 33. Then, both file 30 and file 33 are rewound (lines 195–196) to prepare them to be opened in the update mode by subroutine UPDATE (lines 197–198).

The data manager is then prompted to provide the key number for the record to be deleted (lines 199–200). Key number, KEYR, is tested (line 201) for a negative value which concludes the deletion procedure by transferring execution to statement 999. If the value of KEYR is positive, the record with that key number is read from the master file (line 202), decoded into the raw data form, and displayed on the output device.

The counter for the number of computer words, NWORDS, is set to zero (line 203), and the DO loop (lines 204–224) contains the logic for translating the master record to the raw-data form for display. The DO loop index ranges from 1 to the number of data elements, NELEM, and the ITYPE array is tested to determine the type of each data element.

If the data-element type is an integer, control passes to statement 110 (line 206) where the index, in terms of number of computer words, NWORDS, is computed for the position of that element. Then, the integer data is transferred from the I-th master-file element of IMAGE into the NWORDS-th computer word of NSTRNG (line 207) with control passing to statement 200, the end of the DO loop.

If the data-element type is a real number, control passes from the ITYPE test (line 205) to statement 120 (line 209) where the NWORDS index is then computed for that element. Then, the real data are transferred from the I-th master file element of QIMAGE into the NWORDS position of STRING (line 210). Control then passes to the end of the DO loop for recycling to the next data-element position.

If the data-element type is alphanumeric, control passes from the test of ITYPE (line 205) to statement number 130, where the dictionary pointer value is tested for nonblank value (line 212). If it is blank, the word length of the alphanumeric string, LT, is set to zero (line 213), and the logic then jumps to statement 145 (line 219). If the string is nonblank, control passes to statement 135 (line 215), where subroutine SCANDC is called to return the value for the alphanumeric string from the dictionary entry referenced by the integer value of IMAGE. The dictionary category number is equal to the value of ITYPE minus 10.

The LT nonblank words of the alphanumeric string, NAME, which was returned from SCANDC, are then transferred into the appropriate positions of the NSTRNG array (lines 216 and 217). If some of the words at the end of the string are blank (that is, if LT is less than the KWORD value for that data element, line 218), then blanks are substituted into the remaining NSTRING positions reserved for that data element (lines 219–222). The computer word counter, NWORDS, is then incremented by the number of computer words reserved for the next data element (line 223), and the end of the DO loop is reached (line 224).

After the DO loop activities have been concluded for all NELEM data elements, the deleted record is displayed in the output format according to the format subroutine determined by the value of the master-file index, KDIC (lines 225–227). The deleted record is also written onto the "save" file on file 33.

All data elements belonging to the master-file record are then blanked out (lines 229–230), and a special deletion message is superimposed onto the master-file record (lines 231–240). Parts of the deletion message are stored in several of the different data-element dictionaries. These data-element positions vary according to the master-file record structure. Therefore, the master-file index, KDIC, must be tested (lines 232–236) to determine the data-element positions for storage of the deletion message.

The blanked record containing the deletion information is then written over its original position on the master file (line 241), and it is deleted in its

entirety from the subfile on file unit 21 (line 242) by means of the INFONET-dependent subroutine DELETE. This deletion from the subfile prevents the recreation of the record should the total subfile be posted onto the master file at a later time.

Control is then returned (line 243) to statement 100 for a new key number. If a negative value is entered to conclude the deletion procedure, a message is printed (lines 244–246), files 30 and 33 are rewound (lines 247–248) to remove them from the update mode, and control is returned to subroutine REVISE by the RETURN 1 statement (line 249). If, during the dictionary lookup (subroutine SCANDC), a blank alphanumeric string is encountered, a message is printed (lines 250–254), the master file is closed by a REWIND (line 255), and control is returned to subroutine REVISE (line 256).

As in the other subroutines, INFONET-dependent subroutines are used to place data files in the update mode by first calling subroutine OBEY (lines 189–194) to dynamically equivalence the master file and the subfile to files 30 and 33, respectively, and then, by calling subroutine UPDATE (lines 197–198), to place them in the update mode. The deletion of keyed records from file 21 is accomplished through the use of subroutine DELETE (line 242). Also, certain of the subroutine logic is data dependent and must be modified to accommodate the addition of new master files. This logic occurs in the testing of KDIC for executing the subroutine OBEY calls (lines 189–194), the testing of KDIC for calling the output-display routines (lines 225–227), and the logic for inserting the message onto the master record (lines 229–240) with the associated KDIC tests for the different data positions.

```
100   SUBROUTINE DELREC
101   %
102   %  THIS IS SUBROUTINE "DELREC" (DELETE RECORD).
103   %  IT IS DESIGNED TO DELETE INFORMATION FROM THE RECORDS OF
104   %  THE MASTER FILE, LEAVING A CODED IDENTIFICATION NUMBER
105   %  TO SHOW THE DATE OF DELETION AND THE IDENTIFICATION OF
106   %  THE DATA MANAGER RESPONSIBLE FOR EXECUTING THE DELETION.
107   %  A DELETION MESSAGE IS ALSO SUPERIMPOSED OVER SEVERAL OF
108   %  THE DATA FIELDS WITH THE REMAINING DATA FIELDS LEFT BLANK
109   %  FOR STORAGE IN THE KEYED MASTER FILE.
110   %
111   %  THE RECORD DELETED FROM THE MASTER FILE IS WRITTEN FOR
112   %  PRESERVATION ONTO A SAVE DELETION ("ESAVE," "WSAVE," OR
113   %  "SAVUSA" FILE AS WELL AS PRINTED OUT IN HARD COPY ON THE
114   %  TERMINAL OUTPUT DEVISE.
115   %
116   %    NELEM  = NUMBER OF DATA ELEMENTS IN A MASTER RECORD.
117   %    VLIST  = DATA ELEMENT NAMES AS THEY APPEAR IN THE
118   %               DEFINITION AND MASK FILES.
119   %    ITYPE  = TYPE DESIGNATION OF THE DATA ELEMENTS AS
120   %               LISTED IN THE MASK FILE.
121   %               ITYPE = 1...INTEGER TYPE DATA
122   %               ITYPE = 2...REAL TYPE DATA
123   %               ITYPE > 10...DATA CONSISTS OF A DICTIONARY
124   %                 POINTER TO REFERENCE A LOOKUP IN THE
125   %                 (ITYPE - 10)TH DICTIONARY.
126   %    NREC   = NUMBER OF RECORDS REVIEWED IN THE SELECTED SUBFILE.
127   %    KDIC   = INDEX TO IDENTIFY THE MASTER FILE BEING USED AND TO
128   %               LOCATE THE PROPER MAXREC VALUE FROM THE DICTIONARY
129   %               FILE.
130   %    IMAGE  = INTEGER REPRESENTATION OF MASTER DATA
131   %               ELEMENTS, INCLUDING INTEGER DATA AND
132   %               DICTIONARY POINTERS.
133   %    QIMAGE = REAL REPRESENTATION OF MASTER DATA ELEMENTS
134   %               THROUGH EQUIVALENCE WITH 'IMAGE.'
135   %    NSTRNG = INTEGER REPRESENTATION OF THE DATA STRING FOR
136   %               ALL DATA ELEMENTS (INTEGER & ALPHANUMERIC)
137   %               BELONGING TO A MASTER RECORD.
138   %    STRING = REAL REPRESENTATION (THROUGH EQUIVALENCE
139   %               STATEMENT) OF THE DATA STRING FOR ALL REAL-
140   %               VALUED DATA ELEMENTS BELONGING TO A
141   %               MASTER RECORD.
142   %    NAME   = ARRAY (UP TO 7 COMPUTER WORDS) OF NEW ALPHA-
143   %               NUMERIC DATA FOR COMPARISON WITH EXISTING
144   %               DICTIONARY ENTRIES TO  1) DETERMINE DICT-
145   %               IONARY POINTER OR  2) TO DETERMINE THAT A
146   %               DICTIONARY ENTRY DOES NOT EXIST FOR THE
147   %               GIVEN DICTIONARY CATEGORY.
148   %    LT     = NUMBER OF WORDS (LESS THAN OR EQUAL TO 7)
149   %               NEEDED TO STORE THE ALPHANUMERIC ENTRY IN
150   %               "NAME."
151   %    KWORD  = AN ARRAY SPECIFYING THE MAXIMUM NUMBER OF
152   %               COMPUTER WORDS RESERVED FOR EACH RAW DATA
```

```
153  %              ELEMENT.
154  %     NWORDS = THE TOTAL MAXIMUM NUMBER OF COMPUTER WORDS
155  %              NEEDED TO STORE RAW DATA FOR THE 1ST THROUGH
156  %              THE I-TH DATA ELEMENTS.
157        COMMON          NAMED(800),     ITYPE,          %
158                        PAD(401),       VLIST,          %
159                        NELEM
160        COMMON /LOG/    MONTH,          IDAY,           %
161                        IYEAR,          ILOG,           %
162                        MAXREC,         KWORD,          %
163                        KDIC,           IRITE
164        DOUBLE PRECISION                VLIST(400)
165        DIMENSION       ITYPE(400),     KWORD(400),     %
166                        IMAGE(400),     QIMAGE(400),    %
167                        NSTRNG(400),    STRING(400),    %
168                        NAME(7)
169        EQUIVALENCE    (IMAGE,QIMAGE),(NSTRNG,STRING)
170        DATA IBLANK/' '/
171        IF(IRITE .EQ. 23) GO TO 50
172        WRITE(6,2001)
173   2001 FORMAT(/' THIS IS THE MASTER FILE RECORD DELETION PROCDURE.' %
174        '  WHEN PROMPTED, THE DATA'/' MANAGER WILL SPECIFY THE KEY'  %
175        ' NUMBER OF THE RECORD TO BE DELETED.  THE DELETED'/         %
176        ' RECORD IS WRITTEN FOR PRESERVATION, ONTO THE "ESAVE,"'     %
177        ' "WSAVE," OR "SAVUSA"'/' FILES.  THEN, THE DATA ELEMENTS'   %
178        ' IN THE MASTER FILE WILL BE BLANKED, AND A NEW'/            %
179        ' IDENTIFICATION NUMBER WILL BE WRITTEN ONTO THE MASTER FILE.'%
180        '  THIS NUMBER WILL'/' BE ENCODED TO CONTAIN THE DATE OF'    %
181        ' DELETION AND THE IDENTIFICATION NUMBER OF'/' THE DATA'     %
182        ' MANAGER RESPONSIBLE FOR EXECUTING THE DELETION.  IN'       %
183        ' ADDITION, A'/' DELETION MESSAGE WILL BE SUPERIMPOSED OVER' %
184        ' SEVERAL OF THE DATA FIELDS TO NOTE'/' TO THE USER THAT'    %
185        ' THAT KEY NUMBER NO LONGER HAS A VALID RECORD IN THE MASTER'/%
186        ' FILE.'//' WHEN THE DATA MANAGER HAS CONCLUDED THE DELETION'%
187        ' PROCEDURE, ENTRY OF A "-1" AT'/' THE PROMPT FOR "KEY" WILL'%
188        ' END THE PROCESS.'  )
189   50   IF(KDIC .EQ. 1) CALL OBEY('EQUATE 30 WCOAL',4)
190        IF(KDIC .EQ. 1) CALL OBEY('EQUATE 33 WSAVE',4)
191        IF(KDIC .EQ. 2) CALL OBEY('EQUATE 30 USALYT',4)
192        IF(KDIC .EQ. 2) CALL OBEY('EQUATE 33 SAVUSA',4)
193        IF(KDIC .EQ. 3) CALL OBEY('EQUATE 30 ECOAL',4)
194        IF(KDIC .EQ. 3) CALL OBEY('EQUATE 33 ESAVE',4)
195        REWIND 30
196        REWIND 33
197        CALL UPDATE(30,1)
198        CALL UPDATE(33,1)
199   100  READ(5,*,PROMPT='KEY NUMBER OF RECORD TO BE%
200    DELETED: ') KEYR
201        IF(KEYR .LT. 0) GO TO 999
202        READ(30,KEY=KEYR) (IMAGE(I),I=1,NELEM)
203        NWORDS  =  0
204        DO 200 I=1,NELEM
205        IF(ITYPE(I) - 2) 110,120,130
206   110  NWORDS  =  NWORDS + KWORD(I)
```

```
      NSTRNG(NWORDS)  =  IMAGE(I)
      GO TO 200
120   NWORDS  =  NWORDS + KWORD(I)
      STRING(NWORDS)  =  QIMAGE(I)
      GO TO 200
130   IF(IMAGE(I) .NE. IBLANK) GO TO 135
      LT  =  0
      GO TO 145
135   CALL SCANDC(ITYPE(I)-10,IMAGE(I),NAME,LT,$996)
      DO 140 J=1,LT
140   NSTRNG(J+NWORDS)  =  NAME(J)
      IF(LT .GE. KWORD(I)) GO TO 155
145   LAST  =  KWORD(I)
      LTP1  =  LT + 1
      DO 150 J=LTP1,LAST
150   NSTRNG(J+NWORDS)  =  IBLANK
155   NWORDS  =  KWORD(I) + NWORDS
200   CONTINUE
      IF(KDIC .EQ. 1) CALL OUTUSC(NSTRNG,STRING,KEYR,NWORDS)
      IF(KDIC .EQ. 2) CALL OUTUSA(NSTRNG,STRING,KEYR,NWORDS)
      IF(KDIC .EQ. 3) CALL OUTUSC(NSTRNG,STRING,KEYR,NWORDS)
      WRITE(33,KEY=KEYR) (IMAGE(I),I=1,NELEM)
      DO 300 I=1,NELEM
300   IMAGE(I)  =  IBLANK
      IMAGE(11)  =  2
      IF(KDIC .NE. 1 .OR. KDIC .NE. 3) GO TO 305
      IMAGE(14)  =  2
      IMAGE(17)  =  523
      IMAGE(26)  =  ILOG
305   IF(KDIC .NE. 2) GO TO 310
      IMAGE(13)  =  2
      IMAGE(15)  =  523
      IMAGE(23)  =  ILOG
310   CONTINUE
      WRITE(30,KEY=KEYR) (IMAGE(I),I=1,NELEM)
      CALL DELETE(21,KEYR,1)
      GO TO 100
999   WRITE(6,1002)
1002  FORMAT(/' YOU HAVE ENTERED A NEGATIVE KEY VALUE TO TERMINATE'%
      ' THE DELETION PROCEDURE.')
      REWIND 30
      REWIND 33
      RETURN 1
996   WRITE(6,1010) KEYR
1010  FORMAT(/' ALPHANUMERIC DATA CONTAINS BLANK ENTRIES FOR'      %
      ' RECORD KEY NUMBER: ',I8/ ' RECORD HAS EITHER ALREADY BEEN' %
      ' DELETED OR BEEN INCORRECTLY ENTERED'/' SELECT PROCEDURE'   %
      ' FOUR (KEY REVIEW) TO REVIEW AND CORRECT THIS PROBLEM.'     )
      REWIND 30
      RETURN 1
      END
```

SUBROUTINE SEQREV

Subroutine SEQREV is designed to perform a sequential review of the records contained in a subfile of the master file. A listing of subroutine SEQREV appears at the end of this discussion. After each record is read in sequence, subroutine MODIFY is called to perform the revision of the data elements in the subfile and to call for display of the record. The option of posting the subfile to the master file cannot be executed until program control is returned to REVISE.

The COMMON statements (lines 122–123) contain the variables as listed in the discussion of REVISE. Because the first 405 words in the COMMON statement labeled LOG are not used in SEQREV, they have simply been lumped into an array called IDUMMY.

If the IRITE variable transmitted through COMMON has a value of 23, instructions for use of the sequential revision procedure are not displayed (lines 125–139). Then, the number of records reviewed, NREC, is initialized to zero (line 140), and the sequential reading of the subfile begins (line 141). If an end of file is encountered during the read, all of the records in the subfile have been read, and control proceeds to statement 999 (line 144), which returns program control to REVISE. If a record has been read from the subfile, NREC is incremented by 1, and subroutine MODIFY is called. If the characters, QUIT, have not been entered for the prompt, NAME OF DATA ELEMENT TO BE CHANGED, program control is returned from MODIFY to statement 500 of subroutine SEQREV for the next sequential read. If QUIT has been entered, MODIFY makes a standard return to SEQREV, which, in turn, executes a standard return (line 144) to REVISE.

```
      SUBROUTINE SEQREV(NREC)
      %
      %  THIS IS SUBROUTINE "SEQREV" (SEQUENTIAL REVISION).
      %  IT IS DESIGNED TO READ AND DISPLAY RECORDS, IN
      %  SEQUENCE, FROM A SUBFILE OF THE MASTER FILE FOR THE
      %  REVIEW AND POSSIBLE REVISION OF EACH RECORD.  WHEN
      %  THE ENTIRE SUBFILE HAS BEEN EXAMINED AND THE NECESSARY
      %  REVISIONS COMPLETED, CONTROL WILL BE RETURNED TO
      %  SUBROUTINE "REVISE" WHERE THE DATA MANAGER MAY EXERCISE
      %  THE OPTION OF POSTING THE REVISED FILE ONTO THE MASTER
      %  FILE.
      %
      %    NREC   = NUMBER OF RECORDS REVIEWED IN THE SELECTED SUBFILE.
      %    IMAGE  = INTEGER REPRESENTATION OF MASTER DATA
      %             ELEMENTS, INCLUDING INTEGER DATA AND
      %             DICTIONARY POINTERS.
      %    QIMAGE = REAL REPRESENTATION OF MASTER DATA ELEMENTS
      %             THROUGH EQUIVALENCE WITH 'IMAGE.'
            DIMENSION       VLIST(400),     ITYPE(400),     %
                            IMAGE(400),     QIMAGE(400)
            EQUIVALENCE     (IMAGE,QIMAGE)
            DOUBLE PRECISION                VLIST
            COMMON     NAMED(800),ITYPE,PAD(401),VLIST,NELEM
            COMMON /LOG/ IDUMMY(405),KDIC,IRITE
            IF(IRITE .EQ. 23) GO TO 499
            WRITE(6,1001)
      1001  FORMAT(//' RECORDS FROM THE DESIGNATED SUBFILE WILL BE'     %
           ' PRESENTED SEQUENTIALLY FOR REVIEW'/' AND UPDATE.  AFTER'  %
           ' ALL RECORDS HAVE BEEN EXAMINED BY THE REVIEWER, HE MAY'   %
           ' THEN'/' ELECT TO POST THE RECORDS IN THIS SUBFILE ONTO THE'%
           ' MASTER FILE.'//' WHENEVER YOU WISH TO LEAVE A SELECTED'   %
           ' DATA ELEMENT UNCHANGED, ENTER AN'/' ASTERISK,  *,'        %
           ' FOLLOWED BY A CARRIAGE RETURN.'//' IF YOU WISH TO PROCEDE'%
           ' TO THE NEXT RECORD IN THE FILE, ENTER THE CHARACTERS'/    %
           ' "NEXT" FOLLOWING THE PROMPT: "NAME OF DATA ELEMENT TO BE' %
           ' CHANGED."  THE "NEXT"'/' COMMAND WILL LEAVE THAT RECORD'  %
           ' IN ITS ORIGINAL, UNREVISED STATE AND THE NEXT'/' RECORD'  %
           ' WILL BE DISPLAYED, IN SEQUENCE, FROM THE SUBFILE.  IF AT'%
           ' ANY TIME'/' YOU DO NOT WISH TO REVIEW THE REMAINDER OF'   %
           ' THE FILE, ENTER "QUIT."'  )
      499   NREC  =  0
      500   READ(21,END=999) (IMAGE(I),I=1,NELEM)
            NREC  =  NREC + 1
            CALL MODIFY($500,IMAGE,QIMAGE)
      999   RETURN
            END
```

SUBROUTINE KEYREV

Like SEQREV, most of the record modification is accomplished by a call to subroutine MODIFY. A listing of subroutine KEYREV appears at the end of this discussion. The record selection, however, is accomplished by specifying the key value for a record from the subfile instead of reading the records sequentially.

The variables in the COMMON statements (lines 120–121) are identical with those in the SEQREV discussion. The DATA statement (line 122) initializes IYES and NO for testing against user entries. If the value of IRITE is other than 23 (line 123), the instruction message for the keyed revision procedure will be displayed (lines 124–140).

The number of records reviewed, NREC, is set to zero (line 141), and the user is prompted to enter the value for the record key, IKEY (line 142). Then, IKEY is tested for a negative value (line 143), which will lead to termination of the keyed revision procedure by a transfer to statement 999 for the display of a termination message giving the total number of records reviewed (lines 147–150).

If the IKEY value is positive, NREC is incremented by one (line 144), and the record with that key value is read from the subfile on unit 21 (line 145). Subroutine MODIFY is called (line 146) for the user interactive modification of each record referenced from the subfile. If the QUIT command has not been entered while under the control of MODIFY, and if modification of the record has been completed, control will be returned to statement 100, where the user is prompted for a YES or NO answer to post the record to the master file (lines 153–156). If the response is other than a YES or a NO, a test for these values (line 157) will recycle to statement 10 until the appropriate input is entered. If the response is YES, the record is posted onto the master file on unit 20 (line 158).

A prompt is made for intent to review more records (lines 159–161). Again, a YES or NO response is required. If the test does not match a YES or a NO value (line 162), control is returned to statement 20 for re-entry of the proper value. A NO value will terminate the keyed revision procedure and return to REVISE (line 163). If the value is YES, control (line 164) is recycled to statement 500 for the entry of the key of the next record to be reviewed.

No INFONET-dependent library software is used in this subroutine, nor is there any logic that is dependent on the structure of the master file.

```
      SUBROUTINE KEYREV(*)
      %
      %  THIS IS SUBROUTINE "KEYREV" (KEYED REVISION).
      %  IT IS DESIGNED TO READ AND DISPLAY RECORDS, AS SPECIFIED
      %  BY RECORD KEY, FROM A SUBFILE OF THE MASTER FILE FOR THE
      %  REVIEW AND POSSIBLE REVISION OF EACH RECORD.  WHEN A
      %  RECORD HAS BEEN REVIEWED, THE DATA MANAGER IS GIVEN
      %  THE OPTION OF POSTING THE RECORD ONTO THE MASTER
      %  FILE OR LEAVING IT IN THE SUBFILE FOR FURTHER REVIEW.
      %
      %    NREC   = NUMBER OF RECORDS REVIEWED IN THE SELECTED SUBFILE.
      %    IMAGE  = INTEGER REPRESENTATION OF MASTER DATA
      %             ELEMENTS, INCLUDING INTEGER DATA AND
      %             DICTIONARY POINTERS.
      %    QIMAGE = REAL REPRESENTATION OF MASTER DATA ELEMENTS
      %             THROUGH EQUIVALENCE WITH 'IMAGE.'
            DIMENSION      VLIST(400),     ITYPE(400),     %
                           IMAGE(400),     QIMAGE(400)
            EQUIVALENCE   (IMAGE,QIMAGE)
            DOUBLE PRECISION               VLIST
            COMMON    NAMED(800),ITYPE,PAD(401),VLIST,NELEM
            COMMON /LOG/ IDUMMY(405),KDIC,IRITE
            DATA  IYES,NO/'YES','NO'/
            IF(IRITE .EQ. 23) GO TO 499
            WRITE(6,1001)
      1001  FORMAT(/' RECORDS FROM THE DESIGNATED SUBFILE WILL BE'      %
           ' PRESENTED, AS SPECIFIED BY KEY'/' NUMBER, FOR REVIEW AND' %
           ' UPDATE.  AFTER THE REVIEWER HAS EXAMINED THE CONTENTS'/   %
           ' OF THE RECORD OF INTEREST, HE MAY ELECT TO POST THAT'     %
           ' RECORD ONTO THE MASTER'/' FILE.'//' REGARDLESS OF WHETHER'%
           ' THE RECORD IS POSTED TO THE MASTER FILE, IT WILL REMAIN,'/%
           ' AS REVISED, IN THE SUBFILE.'//' TO ACCESS THE DESIRED'    %
           ' RECORD, RESPOND TO THE PROMPT "KEY" BY ENTERING THE'/     %
           ' RECORD''S KEY NUMBER. ...  ENTERING A "-1" WILL CONCLUDE' %
           ' THE KEYED ACCESS'/' PROCEDURE.'//' IF YOU WISH TO GO ON'  %
           ' TO ANOTHER RECORD IN THE FILE, ENTER THE CHARACTERS'      %
           ' "NEXT"'/' FOLLOWING THE PROMPT: "NAME OF DATA ELEMENT TO' %
           ' BE CHANGED."  THE "NEXT" COMMAND'/' WILL LEAVE THAT'      %
           ' RECORD IN ITS ORIGINAL, UNREVISED STATE AND PROMPT FOR'   %
           ' THE NEXT'/' RECORD KEY.'//' WHENEVER YOU WISH TO LEAVE A' %
           ' SELECTED DATA ELEMENT UNCHANGED, ENTER'/' AN ASTERISK, *.')
      499   NREC  =  0
      500   READ(5,*,PROMPT='KEY: ') IKEY
            IF(IKEY .LT. 0) GO TO 999
            NREC  =  NREC + 1
            READ(21,KEY=IKEY) (IMAGE(I),I=1,NELEM)
            CALL MODIFY($800,IMAGE,QIMAGE)
      999   WRITE(6,1002) NREC
      1002  FORMAT(/' THE KEYED ACCESS REVISION PROCEDURE HAS BEEN'    %
           ' TERMINATED.'/' A TOTAL OF',I4,' RECORDS HAVE BEEN'        %
           ' REVIEWED.')
            REWIND 21
      998   RETURN 1
```

```
800   WRITE(6,2001)
2001  FORMAT(/' DO YOU WISH TO POST THIS RECORD TO THE MASTER'    %
      ' FILE?')
10    READ(5,1003,PROMPT='"YES" OR "NO": ') ITEST
      IF((ITEST .NE. IYES) .AND. (ITEST .NE. NO)) GO TO 10
      IF(ITEST .EQ. IYES) WRITE(20,KEY=IKEY) (IMAGE(I),I=1,NELEM)
      WRITE(6,2002)
2002  FORMAT(/' DO YOU WANT TO REVIEW ANY MORE RECORDS?')
20    READ(5,1003,PROMPT='"YES" OR "NO": ') ITEST
      IF((ITEST .NE. IYES) .AND. (ITEST .NE. NO)) GO TO 20
      IF(ITEST .EQ. NO) RETURN 1
      GO TO 500
1003  FORMAT (A4)
      END
```

SUBROUTINE BATCH

This subroutine is used to change *all* subfile records to contain a specified data value for a particular data element. A listing of subroutine BATCH appears at the end of this discussion. After the batch revisions have been entered onto the specified subfile, control is returned to subroutine REVISE, where the option is presented for posting the subfile onto the master file.

The COMMON statements (lines 163–164) contain the same variables as already described in the discussion of subroutine REVISE. The EQUIVALENCE statement (line 165) is used to permit interchangeability of the integer and the real master-record data arrays, IMAGE and QIMAGE; the integer- and real-data string arrays, NSTRNG and STRING; and the integer and real representations of the asterisk value, ISTAR and STAR, for signaling that the input data value should be left unchanged. The DATA statements (lines 166–167) establish the alphanumeric data values for those quantities. These values are used for the logical testing of user-entered input, or for the blanking of data.

As in all other of the updating subroutines, if the value 23 has been previously entered for IRITE, the test for this value (line 168) will cause the instruction message (lines 169–176) to be skipped.

The user is then prompted for the name of the data element to be changed (lines 177–179). If the value QUIT is entered, the test (line 180) causes the batch revision procedure to be terminated with a transfer to statement 999 for a termination message (lines 277–279) and a normal return to subroutine REVISE. If the value, QUIT, has not been entered, the name of the data element is checked against the list of data-element names to ensure that a legitimate name has been entered (lines 181–184). The user is prompted again (lines 185–187) if the name has been improperly entered. This check for the data-element name also references the value of ITEM, the position of that data element in the master-file record. The value of ITYPE for the ITEM-th position is checked for the data type assigned to that particular data element (line 188).

If the value of ITYPE is equal to 1, control is transferred to statement 110 (line 193), where the user is prompted for entry of the integer data value. If the user enters an asterisk, the test (line 194) is satisfied, and control moves to statement 997, where a message is displayed telling the user that the data item is not changed (lines 273–275). Otherwise, the input value is transferred to the appropriate position of the IMAGE array (line 195), and control is transferred (line 196) to statement 155 for the posting of this data change onto all the records in the subfile.

If the value of ITYPE is equal to 2, a real-data type, control is transferred to statement 120 (line 201), where the user is prompted for entry of the decimal input. Again, if the user enters an asterisk, the test (line 202) is satisfied, and control is transferred to statement 997. If data other than an asterisk is entered, it is placed in the appropriate position of the QIMAGE array (line 203), and then control is transferred (line 204) to statement 155 for the posting of this real-data change onto all the records in the subfile.

Finally, if the ITYPE value for the ITEM-th data position is greater than 2, the corresponding data consists of an alphanumeric dictionary entry, and control is transferred to statement 130 (line 211). Here the logic is structured (lines 211–215) to find the first position, KSUM, and the last position, KSUMP1, of the raw-data array, NSTRING, for storage of the new alphanumeric data-element entry. The user is then prompted for entry of the data value (lines 216–217). If an asterisk has been entered (line 218), control again passes to statement 997 for a commentary message before recycling to statement 60 (line 178) for selection of another data element to be changed.

The alphanumeric data value in NSTRING is then transferred into the seven-word array, NAME (lines 219–224), for comparison with the list of dictionary entries. The number of the dictionary category, NDIC, is determined by subtracting 10 from the ITYPE value for the ITEM-th data element (line 225). Then, subroutine MATCH is called (line 226) to search for a matching alphanumeric entry in that dictionary category. If a match is found, the pointer is computed from NDIC and the calling sequence variable, MKEY, and is stored in the ITEM-th position of the IMAGE array (line 227).

If there is no dictionary match, control is passed through the calling sequence to statement 995, where a message is written (lines 259–262) stating that there is no dictionary match for the given element name and the data-element value. A YES or NO response (line 263) permits the user to add the data-element value to the dictionary. The response is tested (line 264) for a YES or NO entry. If any other data were entered, the prompt is displayed once again. A NO response indicates that the data value is unsatisfactory for dictionary entry (line

265), and control is transferred to statement 60 (line 178), prompting again for the name of the data element to be changed.

If a YES response is given, the number of dictionary entries, NUM, is incremented by 1 (line 266). The value of NUM has been previously retrieved through the calling sequence of subroutine MATCH. NUM appears in the calling sequence of subroutine ADDICT (line 267) for computation of the key value pointing to the new dictionary entry. The value of NUM is then incremented by 1 and stored at the beginning of that dictionary category. It is also stored in the ITEM-th position of the IMAGE array (line 268) for posting onto the subfile, and control is transferred (line 269) to statement 155.

After the numeric data have been entered, or after the proper dictionary entry pointer has been determined for the alphanumeric data, the subfile is ready for revision. The counter for number of records is set to zero (line 236). The subfile is rewound (line 237), placed in the update mode (line 238), and read from the beginning, one record at a time (line 239), with the dummy data array, IDATA. The changed data-element value in the IMAGE array is placed into the same position of the IDATA array (line 240). The number of records, NREC, is incremented by 1 (line 241). The INFONET subroutine, GETKEY, is used (line 242) to obtain the key number associated with the last record read from the subfiles. The changed record is overwritten onto the subfile (line 243) in the same keyed position from which it was read. Control then is transferred (line 244) to statement 160 to read the next record in the file.

When the last record has been read from the subfile, the end of file mark is encountered by the next attempt to read that subfile (line 239), and control is transferred to statement 170, where a message is displayed asking the user if there are any more data elements to be changed. If there are more data elements to be changed, the program cycles back through the same logic. If, however, QUIT is entered for the name of the data element to be changed, control is transferred (line 180) to statement 999, where a termination message is written (lines 277–279), and a standard return (line 280) is made to subroutine REVISE.

Subroutine BATCH does not require any special modification whenever the structure of the master file changes. Two statements in the program, however, are dependent on the INFONET computer system. These are the call to the UPDATE routine (line 238) and the call to the GETKEY routine (line 242). The UPDATE routine permits file 21 to be overwritten without destroying all other data records residing on that file. The GETKEY routine accesses the key value for the last record to be read. This value is required in order to rewrite the record back onto the subfile.

```
100   SUBROUTINE BATCH(*,NREC)
101   %
102   %  SUBROUTINE "BATCH" IS DESIGNED FOR MAKING BATCH CHANGES
103   %  TO GIVEN PARAMETERS OF A SUBFILE SELECTED FROM THE MASTER
104   %  FILE.  ALPHANUMERIC ENTRIES ARE CHECKED WITH THE DICTION-
105   %  ARIES FOR ACCURACY AND, IF THERE IS NO COMPARABLE ENTRY
106   %  IN THE DICTIONARY, THE DATA MANAGER IS QUERIED WITH
107   %  REGARD TO THE VALIDITY OF THAT ALPHANUMERIC ENTRY
108   %  BEFORE ADDING IT TO THE DICTIONARY LIST.
109   %
110   %  AFTER ALL BATCH CHANGES ARE MADE TO THOSE PARAMETERS,
111   %  CONTROL IS RETURNED TO SUBROUTINE "REVISE" FOR FURTHER
112   %  REVISION STEPS OR FOR POSTING THE CHANGED SUBFILE ONTO
113   %  THE MASTER FILE.
114   %
115   %     NELEM  = NUMBER OF DATA ELEMENTS IN A MASTER RECORD.
116   %     VLIST  = DATA ELEMENTS NAMES AS THEY APPEAR IN THE
117   %              DEFINITION AND MASK FILES.
118   %     NREC   = NUMBER OF RECORDS REVIEWED IN THE SELECTED SUBFILE.
119   %     KDIC   = INDEX TO IDENTIFY THE MASTER FILE BEING USED AND TO
120   %              LOCATE THE PROPER MAXREC VALUE FROM THE DICTIONARY
121   %              FILE.
122   %     ITYPE  = TYPE DESIGNATION OF THE DATA ELEMENTS AS
123   %              LISTED IN THE MASK FILE.
124   %              ITYPE = 1...INTEGER TYPE DATA
125   %              ITYPE = 2...REAL TYPE DATA
126   %              ITYPE > 10...DATA CONSISTS OF A DICTIONARY
127   %                POINTER TO REFERENCE A LOOKUP IN THE
128   %                (ITYPE - 10)TH DICTIONARY.
129   %     IMAGE  = INTEGER REPRESENTATION OF MASTER DATA
130   %              ELEMENTS, INCLUDING INTEGER DATA AND
131   %              DICTIONARY POINTERS.
132   %     QIMAGE = REAL REPRESENTATION OF MASTER DATA ELEMENTS
133   %              THROUGH EQUIVALENCE WITH 'IMAGE.'
134   %     NSTRNG = INTEGER REPRESENTATION OF THE DATA STRING FOR
135   %              ALL DATA ELEMENTS (INTEGER & ALPHANUMERIC)
136   %              BELONGING TO A MASTER RECORD.
137   %     STRING = REAL REPRESENTATION (THROUGH EQUIVALENCE
138   %              STATEMENT) OF THE DATA STRING FOR ALL REAL-
139   %              VALUED DATA ELEMENTS BELONGING TO A
140   %              MASTER RECORD.
141   %     ITEM   = DATA ELEMENT POSITION INDEX FOR EACH RECORD.
142   %     NAME   = ARRAY (UP TO 7 COMPUTER WORDS) OF NEW ALPHA-
143   %              NUMERIC DATA FOR COMPARISON WITH EXISTING
144   %              DICTIONARY ENTRIES TO  1) DETERMINE DICT-
145   %              IONARY POINTER OR  2) TO DETERMINE THAT A
146   %              DICTIONARY ENTRY DOES NOT EXIST FOR THE
147   %              GIVEN DICTIONARY CATEGORY.
148   %     LT     = NUMBER OF WORDS (LESS THAN OR EQUAL TO 7)
149   %              NEEDED TO STORE THE ALPHANUMERIC ENTRY IN
150   %              "NAME."
151   %     KWORD  = AN ARRAY SPECIFYING THE MAXIMUM NUMBER OF
152   %              COMPUTER WORDS RESERVED FOR EACH RAW DATA
```

```
153  %               ELEMENT.
154  %     NWORDS = THE TOTAL MAXIMUM NUMBER OF COMPUTER WORDS
155  %              NEEDED TO STORE RAW DATA FOR THE 1ST THROUGH
156  %              THE I-TH DATA ELEMENTS.
157        DIMENSION       IMAGE(400),     NSTRNG(400),  %
158        ITYPE(400),     QIMAGE(400),    STRING(400),  %
159        VLIST(400),     KWORD(400),     NAME(7)
160        DIMENSION       IDATA(400)
161        DOUBLE PRECISION                VLABEL,       %
162        VLIST,          QNEXT,          QUIT
163        COMMON  NAMED(800),ITYPE,PAD(401),VLIST,NELEM
164        COMMON /LOG/ MONTH,IDAY,IYEAR,ILOG,MAXREC,KWORD,KDIC,IRITE
165        EQUIVALENCE (ISTAR,STAR),(IMAGE,QIMAGE),(NSTRNG,STRING)
166        DATA IBLANK,QBLANK,YES,XNO,STAR/' ',' ','YES','NO','*'/
167        DATA QNEXT,QUIT/'NEXT','QUIT'/
168        IF(IRITE .EQ. 23) GO TO 60
169        WRITE(6,1001)
170  1001  FORMAT(/' THIS IS THE BATCH UPDATE PROCEDURE.'/' GIVEN A'    %
171        ' SPECIFIED DATA ELEMENT NAME AND A SPECIFIED DATA ELEMENT' %
172        ' VALUE, THIS'/' PROCEDURE CHANGES ALL RECORDS IN THE GIVEN'%
173        ' SUBFILE TO THE DATA VALUE SPECIFIED'/' FOR THAT DATA'     %
174        ' ELEMENT NAME.'//' THE BATCH EDIT/REVISION PROCEDURE CAN'  %
175        ' BE TERMINATED BY ENTERING "QUIT" WHEN'/' A PROMPT FOR THE'%
176        ' NAME OF THE DATA ELEMENT IS ENCOUNTERED.'                 )
177  1002  FORMAT(A8)
178  60    READ(5,1002,PROMPT='NAME OF DATA ELEMENT TO%
179   BE CHANGED: ') VLABEL
180        IF(VLABEL .EQ. QUIT) GO TO 999
181        ITEM  =  0
182        DO 70 I=1,NELEM
183        ITEM  =  ITEM + 1
184  70    IF(VLABEL .EQ. VLIST(I)) GO TO 80
185        WRITE(6,1003)
186  1003  FORMAT(/' ITEM NOT IN LIST OF DATA ELEMENTS, REENTER')
187        GO TO 60
188  80    IF(ITYPE(ITEM)-2) 110,120,130
189  %
190  % THE FOLLOWING STATEMENTS PERMIT REPLACEMENT OF THE INTEGER
191  % DATA CONTAINED IN THE 'IMAGE' ARRAY.
192  %
193  110   READ(5,*,PROMPT='ENTER VALUE: ') INPT
194        IF(INPT .EQ. ISTAR) GO TO 997
195        IMAGE(ITEM)  =  INPT
196        GO TO 155
197  %
198  % THE FOLLOWING STATEMENTS PERMIT REPLACEMENT OF THE REAL
199  % DATA CONTAINED IN THE 'IMAGE' ARRAY.
200  %
201  120   READ(5,*,PROMPT='ENTER DECIMAL VALUE: ') QPUT
202        IF(QPUT .EQ. STAR) GO TO 997
203        QIMAGE(ITEM)  =  QPUT
204        GO TO 155
205  %
206  % THE FOLLOWING STATEMENTS PERMIT REPLACEMENT OF ALPHA-
```

```
207  % NUMERIC DATA, GENERATE NEW POINTERS, AND UPDATE THE
208  % DICTIONARY IF THE SPECIFIED DATA VALUE DOES NOT
209  % ALREADY RESIDE THERE.
210  %
211  130  KSUM  =  1
212       ITEM1  =  ITEM - 1
213       DO 140 I=1,ITEM1
214  140  KSUM  =  KSUM + KWORD(I)
215       KSUMP1  =  KSUM + KWORD(ITEM) - 1
216  1006 FORMAT(7A4)
217       READ(5,1006,PROMPT='ENTER VALUE: ')  (NSTRNG(I),I=KSUM,KSUMP1)
218       IF(NSTRNG(KSUM) .EQ. ISTAR) GO TO 997
219       LAST  =  KWORD(ITEM)
220       KSUMM1  =  KSUM - 1
221       DO 145 I=1,7
222  145  NAME(I)  =  IBLANK
223       DO 150 I= 1,LAST
224  150  NAME(I)  =  NSTRNG(I+KSUMM1)
225       NDIC  =  ITYPE(ITEM) - 10
226       CALL MATCH (NAME,NDIC,LT,MKEY,NUM,$995,$996)
227       IMAGE(ITEM)  =  MKEY - NDIC*10000
228  %
229  %  THE FOLLOWING STATEMENTS READ, SEQUENTIALLY, ALL OF THE
230  %  RECORDS IN THE SUBFILE AND APPLY THE CHANGE IN DATA
231  %  VALUE FOR THE SPECIFIED DATA PARAMETER TO EACH RECORD.
232  %  AND THEN REWRITE THE RECORD BACK ONTO THE SUBFILE.
233  %
234  %  "IDATA" IS USED AS A DUMMY I/O PARAMETER.
235  %
236  155  NREC  =  0
237       REWIND 21
238       CALL UPDATE(21,1)
239  160  READ(21,END=170) (IDATA(I),I=1,NELEM)
240       IDATA(ITEM)  =  IMAGE(ITEM)
241       NREC  =  NREC + 1
242       CALL GETKEY(21,IKEY)
243       WRITE(21,KEY=IKEY) (IDATA(I), I=1,NELEM)
244       GO TO 160
245  170  WRITE(6,1007)
246  1007 FORMAT(/' DO YOU WISH TO CHANGE ANY MORE DATA ELEMENTS' %
247       ' BELONGING TO THIS FILE?  ENTER ')
248  180  READ(5,1008,PROMPT='"YES" OR "NO": ')TEST
249  1008 FORMAT(A4)
250       IF((TEST .NE. YES) .AND. (TEST .NE. XNO)) GO TO 180
251       IF(TEST .EQ. YES) GO TO 60
252       RETURN 1
253  994  WRITE(6,1010)
254  1010 FORMAT(/' DICTIONARY KEY EXCEEDS PERMISSABLE CAPACITY%
255   FOR THE GIVEN '/' DATA ELEMENT NAME.'/%
256   ' EXECUTION TERMINATED!'/%
257       ' CONTACT PROGRAMMER.')
258       STOP
259  995  WRITE(6,2001) VLABEL,NAME
260  2001 FORMAT(/' THERE IS NO DICTIONARY MATCH FOR DATA NAME: '%
```

```
261         ,A7/' ALPHANUMERIC DATA: ',7A4/%
262         ' DO YOU WISH TO ENTER THIS DATA IN THE DICTIONARY?')
263   190   READ(5,1008,PROMPT='"YES OR "NO": ') TEST
264         IF((TEST .NE. YES) .AND. (TEST .NE. XNO)) GO TO 190
265         IF(TEST .EQ. XNO) GO TO 60
266         NUM  =  NUM+1
267         CALL ADDICT(NAME,LT,NDIC,NUM,$994)
268         IMAGE(ITEM)  =  NUM
269         GO TO 155
270   996   WRITE(6,1009)
271   1009  FORMAT(/' BLANK DICTIONARY ENTRIES ARE NOT LEGITIMATE.')
272         GO TO 170
273   997   WRITE(6,1011) VLABEL
274   1011  FORMAT(/' YOU HAVE ENTERED AN ASTERISK,  *,  TO LEAVE',%
275         1X,A7,' UNCHANGED.'/)
276         GO TO 60
277   999   WRITE(6,1012)
278   1012  FORMAT(/' YOU HAVE ENTERED "QUIT" TO TERMINATE THE BATCH'   %
279         ' EDIT/REVISION PROCEDURE.'  )
280         RETURN
281         END
```

SUBROUTINE MODIFY

This subroutine is used to change the data values on a single record. A listing of subroutine MODIFY appears at the end of this discussion. It is called from subroutine SEQREV or from subroutine KEYREV. In either case, the record is accessed from the subfile by reading each record in sequence, as done by SEQREV, or by referencing the record by its key value, as performed in KEYREV. The user may modify as many data values as desired on a selected record. When modification of that record is concluded, control is returned to the calling subroutine for selection of the next record to be modified.

The major variables are identified and defined in the comments at the beginning of the listing. Those variables listed in the COMMON statements (lines 154–155) are identical with the variables discussed in subroutine REVISE. Again, the integer and real representations of (1) the raw-data strings, NSTRNG and STRING, (2) the master-file arrays, IMAGE and QIMAGE, and (3) the asterisk values, ISTAR and STAR, have been equivalenced in each case (line 156). The data fill is initialized (lines 157–158) for alphanumeric values of YES, XNO, STAR, QNEXT, and QUIT, as well as the variables IBLANK and QBLANK, which are used for inserting blank data fill.

The first action by this subroutine is the decoding of the selected record from its master-file format to the raw-data format (lines 159–199) in order to display the record to the data manager for review. This process begins by initializing to zero (line 159) the counter for the number of computer words, NWORDS, used to store the first "I" data elements in the raw-data string. Then, each of the NELEM data elements is translated into its proper position in the raw-data string, beginning with the DO statement (line 160).

The I-th data element is tested for type by comparing the value in the I-th position of the ITYPE array with the integer 2 (line 161). Thus, if the ITYPE value is 1, the data element is an integer, and control passes to statement 10. If the ITYPE value is 2, the data element is real, and control is transferred to statement 20. If the ITYPE value is greater than or equal to 3 (in fact, ITYPE values will be greater than 10 to include pointer codes for the dictionary categories), the data type is alphanumeric, and control will be transferred to statement 30, where the relevant word string will be accessed from the (ITYPE–10)-th dictionary category.

At statement 10 (line 167), the value of NWORDS is again incremented by the number of computer words, KWORD, reserved for the I-th data element. Then, the integer data in the I-th word of the master record array, IMAGE, is substituted into the NWORDS position of the raw-data string, NSTRNG (line 168). Next, control is transferred (line 169) to statement 99 for recycling through the DO loop for the next data element.

At statement 20 (line 175), the value of NWORDS is again incremented by the number of computer words, KWORD. The real data in the I-th word of the master record array, QIMAGE, is substituted into the NWORDS-th position of the raw-data string, STRING (line 176), and control is transferred (line 17) to statement 99 to continue through the DO loop, processing the next data element.

Statement 30 begins the logic of translating from the dictionary pointer of the data element to the alphanumeric string referenced by that pointer and contained in the dictionary category associated with that data element though the corresponding value in the ITYPE array. First, the IMAGE value is tested for blank data (line 186). If the value is blank, the length of the alphanumeric string, LT, is set to zero (line 187), and control is transferred (line 188) to statement 45. If the data in IMAGE is nonblank, control is transferred to statement 35, where subroutine SCANDC is used to select the corresponding alphanumeric entry from the dictionary (line 189). The calling sequence provides the dictionary category index, ITYPE minus 10, and the pointer, IMAGE. The array containing the alphanumeric string, NAME, and the value giving the number of computer words required for that string, LT, are returned from SCANDC through the calling sequence. If there is a blank dictionary entry corresponding to a nonblank integer value of the pointer, control is transferred to statement 996, where a message is displayed.

After the contents of the record have been displayed to the data manager, he is then prompted for the entry of the name of the data element to be changed (lines 205–206). If, instead of the data name, the characters, NEXT, or the characters, QUIT, are entered, action on the displayed record will be terminated. The entered value is first tested against the value of QNEXT (line 207), and if there is equality, the logic proceeds to statement 998, where a message is displayed (lines 283–285), and control is returned to the calling routine by a RETURN 1 statement (line 286). If the comparison with QNEXT fails, the variable name is next tested

against the value of QUIT (line 208). If equality exists, control is transferred to statement 999, where a message is displayed (lines 287–289), and the record is rewritten onto the subfile (line 290), thus posting all modifications to the record prior to entry of the characters QUIT for the data-element name. A standard return (line 291) is then taken to the calling routine. If the characters, NEXT, have been entered for input of the data-element name, then control is returned to the calling routine without first rewriting the record onto the subfile. Therefore, none of the prior modifications made to that record are posted; instead, the record will remain in the subfile with its original contents unchanged.

If the value of the data-element name, VLABEL, was not entered with the characters NEXT or QUIT, the entry is then tested against all of the data-element names in the VLIST array to ensure that it is a valid data-element name (lines 209–215). If there is a match with one of the data-element names (line 212), control is transferred to statement 80 to begin processing the change in data value for that data element. If there is no match, a message is displayed (lines 213–214), and control is returned (line 215) to statement 60 to re-enter the data-element name.

The value of ITEM has been determined (line 211) when the DO loop is exited by passing the test against the ITEM-th position of VLIST (line 212). This value of ITEM is now used to reference the ITYPE value corresponding to the selected data-element name to test for the data-element type (line 216).

If the ITYPE value is 1, the data type is integer, and control proceeds to statement 110 (line 221), where the data manager may input the replacement data value. If, instead, an asterisk is entered, the test against the value of ISTAR (line 222) is passed, and control transfers first to statement 997, where a message is displayed (lines 279–281), and then to statement 60 (line 282). A prompt is made for a new data-element name without any change to the value of the preceding data element. If a value other than an asterisk has been entered for the integer input, this value is transferred into the position of that data element in the IMAGE array (line 223), and then control is passed to statement 160 for a message query regarding the change of additional data elements for that record (lines 256–258).

If the ITYPE value is 2, the data type is real, and control passes to statement 120 (line 229) to prompt for decimal data entry. If, instead, an asterisk is entered, the test against the value of STAR (line 230) is passed, and control again transfers to statement 997, leaving the value for that data element unchanged. Otherwise, the decimal data value is transferred into the proper data-element position of the QIMAGE array (line 231), and control passes to statement 160 for the message query regarding further modifications.

If the ITYPE value is greater than 10, the data type is alphanumeric, and control passes to statement 130 (line 239), the start of the logic for the entry of alphanumeric data. The value of KSUM, the number of computer words required for the storage of all data elements prior to the ITEM-th plus the first word of the ITEM-th position, is computed (lines 239–242), as is the value of KSUMP1 (line 243), the number of computer words reserved to store all data elements through the ITEM-th position. Next, the data manager is prompted for entry of an alphanumeric data string in the KSUM through KSUMP1 positions of NSTRNG (lines 244–245). The first word of the string is compared with ISTAR (line 246) and, if equal, control is transferred to statement 997, the data element is left unchanged, and an informational message is printed.

The value of LAST is set equal to the current value from the KWORD array (line 247) for use as a limit for the DO loop index, and the value of KSUMM1 is set to the value of KSUM minus one (line 248), for use in computing the NSTRNG index (line 252). After all seven words of the NAME array are blanked out (lines 249–250), the most recent values placed in the NSTRNG array are also transferred into the NAME array (lines 251–252). The dictionary category number, NDIC, is computed by subtracting 10 from the ITEM-th position of the ITYPE array (line 253).

Finally, subroutine MATCH is called (line 254) to search the specified dictionary category for an entry matching the new alphanumeric data input. The alphanumeric data are communicated through the calling sequence by the NAME array, and the dictionary category is communicated by NDIC. If there is a dictionary match, the subroutine returns the number of computer words in the dictionary entry, LT, the key value for the dictionary entry, MKEY, and the number of dictionary entries contained in that category, NUM. If there is no match, control is transferred to statement 995 or, if the data are blank, to statement 996.

If a dictionary match has been found, a standard return is executed from MATCH, and the following statement (line 255) computes the value of the

pointer to that dictionary entry for storage in the IMAGE array. The user is then prompted for a YES or NO entry for a change of additional data elements belonging to the current record (lines 256–259). The YES or NO response is tested to ensure that one of the two allowable responses has been entered (line 260), recycling of the YES or NO prompt, if the test fails. If a proper response has been given, the response is then tested for a YES value (line 262), which will cause control to be transferred to statement 60 (line 205), where a new data-element name will be entered for data-value modification. If the test fails, the IMAGE array containing changed data values will be rewritten onto the subfile over the old data for that record (line 263), and a RETURN 1 exit is taken from subroutine MODIFY (line 264).

If there was no dictionary match found for the new alphanumeric input, the user is prompted with a message giving the data-element name and the newly entered data and asking if it is to be added to the dictionary (lines 265–269). Again, the YES or NO response is tested to ensure that one of the two allowable values has been correctly entered (line 270), returning to the prompt at statement 170, if the test fails. If the response is NO, the test (line 271) is satisfied, and control is returned to statement 60 for selection of another data-element name for data revision. If the response is YES, the number of entries in the given dictionary category is increased by 1 (line 272), and subroutine ADDICT is called (line 273) to enter that alphanumeric data into the dictionary. The following variables are transferred to ADDICT through the calling sequence: the NAME array, containing the new alphanumeric data; the length, LT, of the array; the dictionary category number, NDIC; and the new number of entries for that category, NUM.

The value of NUM, the dictionary pointer for the new entry, is transferred into the ITEM-th position of the IMAGE array (line 274), and control is transferred (line 275) to statement 160 for a prompted response to continue with data revision for that record or to go on to another record.

If the new data entered consist only of blank data, the return from subroutine MATCH transfers control to statement 996, where a message is displayed stating that blank data are not legitimate (lines 276–277). The blank data value is *not* substituted for the data element, and control is returned to statement 160 for the user's selection of additional data elements to be revised.

The statements in MODIFY which are master file dependent are the calls to the record display routines (lines 201 and 203) based on the value of the master-file index, KDIC. One INFONET-dependent subroutine, GETKEY, is called (line 200) to obtain the key value for the last record read from file 21.

```
100   SUBROUTINE MODIFY(*,IMAGE,QIMAGE)
101   %
102   %     SUBROUTINE 'MODIFY' IS DESIGNED TO READ A RECORD FROM
103   %     AN EXISTING, KEYED, MASTER FILE, OR A SUBFILE
104   %     OF THE MASTER FILE, AND DISPLAY THE DATA SO THAT
105   %     THE USER IS ABLE TO MODIFY THE INDIVIDUAL DATA
106   %     ELEMENTS AS REQUIRED.
107   %
108   %     NELEM  = NUMBER OF DATA ELEMENTS IN A MASTER RECORD.
109   %     VLIST  = DATA ELEMENTS NAMES AS THEY APPEAR IN THE
110   %              DEFINITION AND MASK FILES.
111   %     ITYPE  = TYPE DESIGNATION OF THE DATA ELEMENTS AS
112   %              LISTED IN THE MASK FILE.
113   %              ITYPE = 1...INTEGER TYPE DATA
114   %              ITYPE = 2...REAL TYPE DATA
115   %              ITYPE > 10...DATA CONSISTS OF A DICTIONARY
116   %                POINTER TO REFERENCE A LOOKUP IN THE
117   %                (ITYPE - 10)TH DICTIONARY.
118   %     IMAGE  = INTEGER REPRESENTATION OF MASTER DATA
119   %              ELEMENTS, INCLUDING INTEGER DATA AND
120   %              DICTIONARY POINTERS.
121   %     QIMAGE = REAL REPRESENTATION OF MASTER DATA ELEMENTS
122   %              THROUGH EQUIVALENCE WITH 'IMAGE.'
123   %     NSTRNG = INTEGER REPRESENTATION OF THE DATA STRING FOR
124   %              ALL DATA ELEMENTS (INTEGER & ALPHANUMERIC)
125   %              BELONGING TO A MASTER RECORD.
126   %     STRING = REAL REPRESENTATION (THROUGH EQUIVALENCE
127   %              STATEMENT) OF THE DATA STRING FOR ALL REAL-
128   %              VALUED DATA ELEMENTS BELONGING TO A
129   %              MASTER RECORD.
130   %     ITEM   = DATA ELEMENT POSITION INDEX FOR EACH RECORD.
131   %     NAME   = ARRAY (UP TO 7 COMPUTER WORDS) OF NEW ALPHA-
132   %              NUMERIC DATA FOR COMPARISON WITH EXISTING
133   %              DICTIONARY ENTRIES TO  1) DETERMINE DICT-
134   %              IONARY POINTER OR  2) TO DETERMINE THAT A
135   %              DICTIONARY ENTRY DOES NOT EXIST FOR THE
136   %              GIVEN DICTIONARY CATEGORY.
137   %     LT     = NUMBER OF WORDS (LESS THAN OR EQUAL TO 7)
138   %              NEEDED TO STORE THE ALPHANUMERIC ENTRY IN
139   %              "NAME."
140   %     KWORD  = AN ARRAY SPECIFYING THE MAXIMUM NUMBER OF
141   %              COMPUTER WORDS RESERVED FOR EACH RAW DATA
142   %              ELEMENT.
143   %     NWORDS = THE TOTAL MAXIMUM NUMBER OF COMPUTER WORDS
144   %              NEEDED TO STORE RAW DATA FOR THE 1ST THROUGH
145   %              THE I-TH DATA ELEMENTS.
146   %     NDIC   = DICTIONARY CATEGORY REFERENCE NUMBER.
147   %     NUM    = TOTAL NUMBER OF DICTIONARY ENTRIES IN THE
148   %              "NDIC" DICTIONARY CATEGORY.
149         DOUBLE PRECISION                    VLABEL,        %
150         VLIST,          QNEXT,              QUIT
151         DIMENSION       IMAGE(400),         NSTRNG(400),   %
152         ITYPE(400),     QIMAGE(400),        STRING(400),   %
```

```
153         VLIST(400),     KWORD(400),      NAME(7)
154         COMMON     NAMED(800),ITYPE,PAD(401),VLIST,NELEM
155         COMMON /LOG/ MONTH,IDAY,IYEAR,ILOG,MAXREC,KWORD,KDIC,IRITE
156         EQUIVALENCE (ISTAR,STAR),(IMAGE,QIMAGE),(NSTRNG,STRING)
157         DATA IBLANK,QBLANK,YES,XNO,STAR/' ',' ','YES','NO','*'/
158         DATA QNEXT,QUIT/'NEXT','QUIT'/
159         NWORDS  =  0
160         DO 100 I=1,NELEM
161         IF(ITYPE(I) - 2) 10,20,30
162   %
163   % THE FOLLOWING STATEMENTS ACCEPT THE DATA ELEMENTS
164   % WHICH ARE INTEGER WORDS AND ADD THEM TO THE TOTAL
165   % RECORD STRING.
166   %
167   10    NWORDS  =  NWORDS + KWORD(I)
168         NSTRNG(NWORDS)  =  IMAGE(I)
169         GO TO 99
170   %
171   % THE FOLLOWING STATEMENTS ACCEPT THE DATA ELEMENTS
172   % WHICH ARE REAL WORDS AND ADD THEM TO THE TOTAL
173   % RECORD STRING.
174   %
175   20    NWORDS  =  NWORDS + KWORD(I)
176         STRING(NWORDS)  =  QIMAGE(I)
177         GO TO 99
178   %
179   % THE FOLLOWING STATEMENTS BRING THE ALPHANUMERIC
180   % ELEMENTS (REFERENCED BY THE CODED POINTERS) FROM
181   % THE APPROPRIATE DICTIONARIES AND ADD THESE
182   % ELEMENTS TO THE TOTAL RECORD STRING, INCLUDING
183   % THOSE PORTIONS OF THE DICTIONARY WORDS WHICH
184   % ARE BLANK.
185   %
186   30    IF(IMAGE(I) .NE. IBLANK) GO TO 35
187         LT  =  0
188         GO TO 45
189   35    CALL SCANDC(ITYPE(I)-10,IMAGE(I),NAME,LT,$996)
190         DO 40 J=1,LT
191   40    NSTRNG(J+NWORDS)  =  NAME(J)
192         IF(LT .GE. KWORD(I)) GO TO 55
193   45    LAST  =  KWORD(I)
194         LTP1  =  LT + 1
195         DO 50 J=LTP1,LAST
196   50    NSTRNG(J+NWORDS)  =  IBLANK
197   55    NWORDS  =  NWORDS + KWORD(I)
198   99    CONTINUE
199   100   CONTINUE
200         CALL GETKEY(21,KEYR)
201         IF(KDIC .EQ. 1) CALL OUTUSC(NSTRNG,STRING,KEYR,NWORDS)
202         IF(KDIC .EQ. 2) CALL OUTUSA(NSTRNG,STRING,KEYR,NWORDS)
203         IF(KDIC .EQ. 3) CALL OUTUSC(NSTRNG,STRING,KEYR,NWORDS)
204   1002  FORMAT(A8)
205   60    READ(5,1002,PROMPT='NAME OF DATA ELEMENT TO%
206    BE CHANGED: ') VLABEL
```

```
      IF(VLABEL .EQ. QNEXT) GO TO 998
      IF(VLABEL .EQ. QUIT) GO TO 999
      ITEM  =  0
      DO 70 I=1,NELEM
      ITEM  =  ITEM + 1
70    IF(VLABEL .EQ. VLIST(I)) GO TO 80
      WRITE(6,1003)
1003  FORMAT(/' ITEM NOT FOUND IN LIST OF DATA ELEMENTS,REENTER')
      GO TO 60
80    IF(ITYPE(ITEM)-2) 110,120,130
%
% THE FOLLOWING STATEMENTS PERMIT REPLACEMENT OF THE INTEGER
% DATA CONTAINED IN THE 'IMAGE' ARRAY.
%
110   READ(5,*,PROMPT='ENTER VALUE: ') INPT
      IF(INPT .EQ. ISTAR) GO TO 997
      IMAGE(ITEM)  =  INPT
      GO TO 160
%
% THE FOLLOWING STATEMENTS PERMIT REPLACEMENT OF THE REAL
% DATA CONTAINED IN THE 'IMAGE' ARRAY.
%
120   READ(5,*,PROMPT='ENTER DECIMAL VALUE: ') QPUT
      IF(QPUT .EQ. STAR) GO TO 997
      QIMAGE(ITEM)  =  QPUT
      GO TO 160
%
% THE FOLLOWING STATEMENTS PERMIT REPLACEMENT OF ALPHA-
% NUMERIC DATA, GENERATE NEW POINTERS, AND UPDATE THE
% DICTIONARY IF THE SPECIFIED DATA VALUE DOES NOT
% ALREADY RESIDE THERE.
%
130   KSUM  =  1
      ITEM1  =  ITEM - 1
      DO 140 I=1,ITEM1
140   KSUM  =  KSUM + KWORD(I)
      KSUMP1  =  KSUM + KWORD(ITEM) - 1
1006  FORMAT(7A4)
      READ(5,1006,PROMPT='ENTER DATA: ')  (NSTRNG(I),I=KSUM,KSUMP1)
      IF(NSTRNG(KSUM) .EQ. ISTAR) GO TO 997
      LAST  =  KWORD(ITEM)
      KSUMM1  =  KSUM - 1
      DO 145 I=1,7
145   NAME(I)  =  IBLANK
      DO 150 I= 1,LAST
150   NAME(I)  =  NSTRNG(I+KSUMM1)
      NDIC  =  ITYPE(ITEM) - 10
      CALL MATCH (NAME,NDIC,LT,MKEY,NUM,$995,$996)
      IMAGE(ITEM)  =  MKEY - NDIC*10000
160   WRITE(6,1007)
1007  FORMAT(/' DO YOU WISH TO CHANGE ANY MORE DATA ELEMENTS'/%
      ' BELONGING TO THIS RECORD?  ENTER ')
165   READ(5,1008,PROMPT='"YES" OR "NO": ')TEST
      IF((TEST .NE. YES) .AND. (TEST .NE. XNO)) GO TO 165
```

```
1008 FORMAT(A4)
     IF(TEST .EQ. YES) GO TO 60
     WRITE(21,KEY=KEYR) (IMAGE(I),I=1,NELEM)
     RETURN 1
995  WRITE(6,2001) VLABEL,NAME
2001 FORMAT(/' THERE IS NO DICTIONARY MATCH FOR DATA NAME: '%
     ,A7/' ALPHANUMERIC DATA: ',7A4/%
     ' DO YOU WISH TO ENTER THIS DATA IN THE DICTIONARY?')
170  READ(5,1008,PROMPT='"YES" OR "NO": ') TEST
     IF((TEST .NE. YES) .AND. (TEST .NE. XNO)) GO TO 170
     IF(TEST .EQ. XNO) GO TO 60
     NUM  =  NUM+1
     CALL ADDICT(NAME,LT,NDIC,NUM)
     IMAGE(ITEM)  =  NUM
     GO TO 160
996  WRITE(6,1009)
1009 FORMAT(/' BLANK DICTIONARY ENTRIES ARE NOT LEGITIMATE.')
     GO TO 160
997  WRITE(6,1011) VLABEL
1011 FORMAT(/' YOU HAVE ENTERED AN ASTERISK,  *,  TO LEAVE',%
     1X,A7,' UNCHANGED.'/)
     GO TO 60
998  WRITE(6,1012) KEYR
1012 FORMAT(/' YOU HAVE ENTERED "NEXT" WHICH LEAVES THE RECORD' %
     ' (KEY =',I8,') UNREVISED.')
     RETURN 1
999  WRITE(6,1013)
1013 FORMAT(/' YOU HAVE ENTERED "QUIT" TO TERMINATE THIS'      %
     ' REVISION PROCEDURE.')
     WRITE(21,KEY=KEYR) (IMAGE(I),I=1,NELEM)
     RETURN
     END
```

SUBROUTINE SCANDC

Subroutine SCANDC is used to locate the value of the alphanumeric data array, NAME, which is referenced by the dictionary category number, NDIC, and the dictionary entry pointer, IPOINT. A listing of subroutine SCANDC appears at the end of this discussion. The subroutine returns the data value from the dictionary and the number of computer words, LT, required by the data string.

The DATA statement (line 112) is used to initialize the value of IBLANK as blank data. Then, all seven words of the NAME array are set to blank values (lines 113–114). The key value, LKEY, is computed (line 115) and used to read the number of dictionary entries, NUM, contained in the NDIC category (line 116) as well as K, which, although not used in this subroutine, has the same value as NDIC. The key value for the dictionary entry, NKEY, is calculated from LKEY and the dictionary pointer, IPOINT (line 117). If LKEY is too large for the number of entries in that dictionary category (line 118), control is transferred to statement 100, where an error message is dispalyed (lines 121–127), and a nonstandard return is made (line 128) to the calling subroutine.

If the value of NKEY is acceptable, the number of computer words, LT, and the dictionary entry, NAME, are read from the dictionary on file 9.

All statements in subroutine SCANDC are written in standard FORTRAN, no statement being INFONET-dependent. No statement appearing in this subroutine exists to accommodate a specific data base.

```
100   SUBROUTINE SCANDC(NDIC,IPOINT,NAME,LT,*)
101   %
102   % THIS IS SUBROUTINE 'SCANDC.'
103   % GIVEN THE DICTIONARY CATEGORY NUMBER, 'NDIC,' AND THE
104   % CATEGORY ELEMENT POINTER, 'IPOINT,' THIS SUBROUTINE
105   % READS THE DICTIONARY ELEMENT NAME, 'NAME,' AND THE
106   % LENGTH OF THE NAME IN WORDS, 'LT,' AND RETURNS THESE
107   % VALUES TO THE CALLING ROUTINE FOR DISPLAY TO THE USER
108   % FOR THE PURPOSE OF IDENTIFICATION, DATA REVIEW, AND
109   % POSSIBLE REVISION.
110   %
111         DIMENSION      NAME(7)
112         DATA IBLANK/' '/
113         DO 10 I=1,7
114   10    NAME(I)  =  IBLANK
115         LKEY  =  NDIC*10000
116         READ(9,KEY=LKEY) NUM,K
117         NKEY  =  LKEY + IPOINT
118         IF(NKEY .GT. LKEY + NUM)  GO TO 100
119         READ(9,KEY=NKEY) LT,(NAME(I),I=1,LT)
120         RETURN
121   100   WRITE(6,1001) IPOINT,NUM,NDIC
122   1001  FORMAT(/' DICTIONARY POINTER (',I8,%
123         ') EXCEEDS NUMBER OF ENTRIES (',I8,')'/%
124         ' IN CATEGORY ''',I2,''',',/%
125         ' EITHER THE DICTIONARY POINTER IS IN ERROR, OR'/%
126         ' DICTIONARY LIST EXCEEDS 10000 ENTRIES.  CONTACT'/%
127         ' PROGRAMMER.')
128         RETURN 1
129   END
```

SUBROUTINE ADDICT

This subroutine is used to add new dictionary entries to the specified dictionary category. A listing of subroutine ADDICT appears at the end of this discussion. The alphanumeric data for the entry are contained in the NAME array, the number of computer words needed for the data is contained in LT, the dictionary category number is contained in NDIC, and the number of entries in the dictionary category is contained in NUM, all of which enter the subroutine through the calling sequence.

The value of NUM is tested (line 120) to ensure that it does not exceed the number of entries reserved for each dictionary category. If the test fails, control is transferred to statement 994, where a diagnostic message is written (lines 129–133), and execution is stopped (line 134). If the test of NUM is passed, file 9 is rewound (line 121) so that it can be placed in the update mode (line 122). The keys for the number of entries, NUM, in a dictionary category and for the new dictionary entry are computed (lines 123–124). Then, the number of dictionary entries, NUM, and the corresponding dictionary category number, NDIC, are written on the IKEY-th line, and the length, LT, and the nonblank words in data array NAME are written on the NKEY-th line of the dictionary file (lines 125–126). The dictionary file is then closed with a rewind (line 127), and control is returned to the calling routine with a standard return (line 128).

There is only one nonstandard FORTRAN statement in subroutine ADDICT. This is the call to UPDATE (line 122), which opens the dictionary file in the update mode so that records can be written onto the file without destroying the other data residing in that file.

```
100   SUBROUTINE ADDICT(NAME,LT,NDIC,NUM)
101   %
102   % THIS IS SUBROUTINE 'ADDICT.'
103   % IT IS DESIGNED TO ADD SPECIFIED ALPHANUMERIC DATA
104   % ELEMENTS TO THE DICTIONARY LIST FOR EACH REVISED
105   % DICTIONARY CATEGORY.  IF A CATEGORY LIST EXCEEDS
106   % 10000 ENTRIES, AN ERROR RETURN MESSAGE TO CONTACT
107   % THE PROGRAMMER WILL BE DISPLAYED AND EXECUTION
108   % WILL BE TERMINATED.
109   %
110   %     NAME    = ARRAY (UP TO 7 COMPUTER WORDS) OF NEW ALPHA-
111   %               NUMERIC DATA FOR ENTRY INTO THE APPROPRIATE
112   %               DICTIONARY CATEGORY.
113   %     LT      = NUMBER OF WORDS (LESS THAN OR EQUAL TO 7)
114   %               NEEDED TO STORE THE ALPHANUMERIC ENTRY IN
115   %               "NAME."
116   %     NDIC    = DICTIONARY CATEGORY REFERENCE NUMBER.
117   %     NUM     = TOTAL NUMBER OF DICTIONARY ENTRIES IN THE
118   %               "NDIC" DICTIONARY CATEGORY.
119         DIMENSION       NAME(7)
120         IF(NUM .GE. 9999) GO TO 994
121         REWIND 9
122         CALL UPDATE(9,1)
123         IKEY  =  NDIC*10000
124         NKEY  =  IKEY + NUM
125         WRITE(9,KEY=IKEY) NUM,NDIC
126         WRITE(9,KEY=NKEY) LT,(NAME(I),I=1,LT)
127         REWIND 9
128         RETURN
129   994   WRITE(6,1010)
130   1010  FORMAT(/' DICTIONARY KEY EXCEEDS PERMISSABLE CAPACITY%
131    FOR THE GIVEN '/' DATA ELEMENT NAME.'/%
132    ' EXECUTION TERMINATED!'/%
133         ' CONTACT PROGRAMMER.')
134   STOP
135   END
```

SUBROUTINE MATCH

This subroutine searches all dictionary entries in a given dictionary category until either it finds a matching entry for new alphanumeric data input or it determines that no dictionary entry matches the new data input. A listing of subroutine MATCH appears at the end of this discussion. The inputs that are transmitted to subroutine MATCH through the calling sequence are the alphanumeric data string of the array, NAME, and the number of the dictionary category, NDIC. The subroutine returns the following values: LN, the number of computer words used by the data string; NKEY, the value of the key of the matching dictionary entry; NUM, the number of dictionary entries in the dictionary category. It also may execute two nonstandard returns.

The DATA statement (line 125) is used to initialize the contents of IBLANK as blank data. The value of IKEY identifies the line number where the value of NUM is stored for the dictionary category number, NDIC. The value of IKEY is computed (line 126) and used to read the values of NUM and K (line 127). (The value of K exists on the file, but it is not used in this subroutine). The number of nonblank words, LN, is computed for the input data string, NAME, with a RETURN 2 exit taken if LN is equal to zero (lines 136–140).

The actual dictionary search for a matching entry takes place in the DO loop (lines 150–158) with the DO loop index, I, going from one to NUM, the number of dictionary entries. The seven-word array, ITANK, is used to read in each dictionary entry. First, all words in the array are set to blanks (lines 151–152). Then, the length of data string in computer words, LT, and the same number of computer words from the dictionary entry are read into the ITANK array. If the input array and the dictionary array do not contain the same number of computer words of data (that is, if LN≠LT), control then passes (line 154) to the end of the DO loop. Each word of the NAME array is checked against each word of the ITANK array, and, if for any word of the string there is a mismatch (line 156), control again proceeds to the end of the DO loop at statement 50.

If the test (line 156) fails for all LN nonblank words, then there is a data match with that dictionary entry. Control transfers (lines 157) to statement 90, where subroutine GETKEY is called (line 160) to obtain the line key for the matching dictionary entry. A standard return (line 161) is taken to end the search for a matching dictionary entry.

No special logic is required in this routine to accommodate differences in the record structure for each master file. However, one INFONET-dependent subroutine, GETKEY, is used to obtain the line key value, NKEY, for the matching dictionary entry (line 160).

```
100   SUBROUTINE MATCH(NAME,NDIC,LN,NKEY,NUM,*,*)
101   % THIS IS SUBROUTINE ''MATCH.''
102   % IT IS DESIGNED TO READ THE KEYED DICTIONARY FILE,
103   % SEARCH FOR THE DICTIONARY ELEMENT REQUESTED, AND
104   % RETURN THE KEY ASSOCIATED WITH THAT ELEMENT
105   % ALONG WITH THE ELEMENT NAME, IF THERE IS A MATCH.
106   %
107   %     NAME    = ARRAY (UP TO 7 COMPUTER WORDS) OF NEW ALPHA-
108   %               NUMERIC DATA FOR COMPARISON WITH EXISTING
109   %               DICTIONARY ENTRIES TO  1) DETERMINE DICT-
110   %               IONARY POINTER OR  2) TO DETERMINE THAT A
111   %               DICTIONARY ENTRY DOES NOT EXIST FOR THE
112   %               GIVEN DICTIONARY CATEGORY.
113   %     LN      = NUMBER OF WORDS (LESS THAN OR EQUAL TO 7)
114   %               NEEDED TO STORE THE ALPHANUMERIC ENTRY IN
115   %               "NAME."
116   %     NDIC    = DICTIONARY CATEGORY REFERENCE NUMBER.
117   %     NUM     = TOTAL NUMBER OF DICTIONARY ENTRIES IN THE
118   %               "NDIC" DICTIONARY CATEGORY.
119   %     ITANK   = ALPHANUMERIC DICTIONARY ENTRY (AN ARRAY OF UP
120   %               TO 7 COMPUTER WORDS)
121   %     LT      = NUMBER OF WORDS (LESS THAN OR EQUAL TO 7)
122   %               NEEDED TO STORE THE ALPHANUMERIC ENTRY IN
123   %               "ITANK."
124         DIMENSION NAME(7),ITANK(7)
125         DATA IBLANK/' '/
126         IKEY  =  NDIC*10000
127         READ(9,KEY=IKEY) NUM,K
128   %
129   %     NOTE: 'K' IS THE NUMBER OF THE DICTIONARY CATEGORY AND
130   %           'NUM' IS THE NUMBER OF ELEMENTS IN THAT CATEGORY.
131   %
132   %
133   %     THE FOLLOWING SET OF STATEMENTS FIND THE NUMBER, 'LN,'
134   %     OF NON-BLANK WORDS IN THE ELEMENT TO BE LOCATED.
135   %
136         LN  =  0
137         DO 25 I=1,7
138         IF(NAME(I) .EQ. IBLANK) GO TO 30
139   25    LN  =  I
140   30    IF(LN .EQ. 0) RETURN 2
141   %
142   % THESE STATEMENTS SEARCH THROUGH THE NUMBER OF DICTIONARY
143   % ELEMENTS, 'NUM,' IN THE SPECIFIED DICTIONARY CATEGORY
144   % TO LOCATE A DATA ELEMENT MATCH.  IF THERE IS NO MATCH,
145   % A 'RETURN 1' EXIT RETURNS EXECUTION TO THE CALLING SUB-
146   % ROUTINE WHERE THE USER WILL BE QUERIED AS TO WHETHER
147   % OR NOT HE WISHES TO HAVE HIS UNMATCHED ALPHANUMERIC
148   % DATA ADDED TO THE DICTIONARY LISTING.
149   %
150         DO 50 I=1,NUM
151         DO 35 L=1,7
152   35    ITANK(L)  =  IBLANK
```

```
      READ(9) LT,(ITANK(K),K=1,LT)
      IF(LN .NE. LT) GO TO 50
      DO 40 K=1,LN
40    IF(NAME(K) .NE. ITANK(K)) GO TO 50
      GO TO 90
50    CONTINUE
      RETURN 1
90    CALL GETKEY(9,NKEY)
      RETURN
END
```

SUBROUTINE OUTUSC

Subroutine OUTUSC is a special subroutine written to provide a formatted display of the raw data of the WCOAL and ECOAL files for review and update. A listing of subroutine OUTUSC appears at the end of this discussion. The subroutine is totally dependent on the data-element names and the record structure of these files. The raw data from NSTRNG and STRING are transferred into the arrays NST and XST (lines 109–111). This is done because NST has been dimensioned to the exact number of computer words needed for the integer and alphanumeric data, so that the output operation occurs as a block write (line 112) rather than as an implied DO write. The block form of input/output is considerably more efficient on the INFONET system. After the record has been written according to the accompanying format, control returns to the calling routine for further operation.

```
100         SUBROUTINE OUTUSC(NSTRNG,STRING,KEYR,KWT)
101   %
102   %   THIS IS AN OUTPUT SUBROUTINE THAT HAS BEEN FORMATTED
103   %   FOR TERMINAL OUTPUT OF THE "USCOAL" RECORD WITH THE
104   %   DATA ELEMENTS DISPLAYED IN THE SAME FORM AS THEY HAVE
105   %   BEEN ENTERED INTO THE RECORD.
106   %
107         DIMENSION       NSTRNG(400),      STRING(400),    NST(75)
108         EQUIVALENCE     (NSTRNG,STRING)
109         DO 10 I=1,75
110   10    NST(I)  =  NSTRNG(I)
111         XST  =  STRING(76)
112         WRITE(6,1001) NST,KEYR,XST
113   1001  FORMAT(/30X '* * * * * * * * * * *'//                          %
114         '   STATE: ',4A4,8X,'COUNTY: ',6A4/                           %
115         '  PMERID: ',I2,2X,'TWNSHIP: ',I3,' NS: ',A1,3X,              %
116         'RANGE: ',I3,' EW: ',A1,'  SECTION: ',I2,4X,                  %
117         'AAPGPRV: ',I3//                                              %
118         ' COALPRV: ',4A4,3X,'REGION: ',6A4,7X,'THICKNS: ',I1/         %
119         '   FIELD: ',4A4,2X,'DISTRCT: ',6A4,7X,'OVRBRDN: ',I1/        %
120         ' FORMATN: ',4A4,6X,'BED: ',5A4,11X,'RELIABL: ',I1//          %
121         '  SYSTEM: ',4A4,5X,'QUAD: ',7A4,6X,'BYEAR: ',I2/             %
122         '  SERIES: ',4A4,3X,'SOURCE: ',5A4,2X,'YEAR: ',I4//           %
123         '    RANK: ',2A4,A1,6X,'ID: ',I8,' ... KEY: ',I8,9X,          %
124         'TONNAGE: ',F8.2/30X '* * * * * * * * * * *'//                )
125         RETURN
126         END
```

SUBROUTINE OUTUSA

Subroutine OUTUSA is a special subroutine written to provide a formatted display of the raw data of the USALYT file for review and update. A listing of subroutine OUTUSA appears at the end of this discussion. The subroutine is totally dependent on the data-element names and the record structure of this file. The raw data from NSTRNG and STRING are transferred into the arrays NST and XST (lines 110–113). This is done because NST has been dimensioned to the exact number of computer words needed for the integer and alphanumeric data, and XST has been dimensioned to the exact number of computer words needed for the real data so that the output operation occurs as a block write (line 114) rather than as an implied DO write. The block form of input/output is considerably more efficient on the INFONET system. After the record has been written according to the accompanying format, control returns to the calling routine.

```
100         SUBROUTINE OUTUSA(NSTRNG,STRING,KEYR,KWT)
101   %
102   %   THIS IS AN OUTPUT SUBROUTINE THAT HAS BEEN FORMATTED
103   %   FOR TERMINAL OUTPUT OF THE "USALYT" RECORD WITH THE
104   %   DATA ELEMENTS DISPLAYED IN THE SAME FORM AS THEY HAVE
105   %   BEEN ENTERED INTO THE RECORD.
106   %
107         DIMENSION       NSTRNG(400),    STRING(400),    %
108                         NST(81),        XST(13)
109         EQUIVALENCE     (NSTRNG,STRING)
110         DO 10 I=1,81
111   10    NST(I)  =  NSTRNG(I)
112         DO 20 I=1,13
113   20    XST(I)  =  STRING(I+81)
114         WRITE(6,1001) NST,XST,KEYR
115   1001  FORMAT(/30X'* * * * * * * * * * *'//                          %
116         '   STATE: ',4A4,8X,'COUNTY: ',6A4/                           %
117         '  PMERID: ',I2,2X,'TWNSHIP: ',I3,' NS: ',A1,3X,              %
118         'RANGE: ',I3,' EW: ',A1,'  SECTION: ',I2,4X,                  %
119         'AAPGPRV: ',I3//                                              %
120   ' COALPRV: ',4A4,3X,'REGION: ',6A4/                                 %
121   '   FIELD: ',4A4,2X,'DISTRCT: ',6A4/                                %
122   ' FORMATN: ',4A4,6X,'BED: ',5A4/                                    %
123   '  SYSTEM: ',4A4,5X,'QUAD: ',7A4/                                   %
124         '  SERIES: ',3A4,7X,'SOURCE: ',5A4,2X,'YEAR: ',I4//          %
125   '    RANK: ',2A4,A1,4X,'API ID: ',A4,I8,4X,'S-TYP: ',I1,3X,'A-%
126   TYP: ',I1,3X,'VAL/REP: ',I1/                                        %
127   '   TRACE: ',A1,12X,'HGRIND: ',I3,16X,'OTHER TESTS: ',A1//          %
128   '     BTU: ',I5,4X,'ASH:(DEFORM) ',I4,2X,'(SOFT) ',I4,2X,'(F%
129   LUID) ',I4,3X,'FRE-SWEL: ',F3.1//                                   %
130   '     MOISTUR  VOL-MAT  FIXED-C  ASH-PRX  CARBON   HYDROGN  %
131   OXYGEN (PERCENT)'/7X,F4.1,6(5X,F4.1)//                              %
132   '     NITROGN  SULFUR   ORG-SUL  PYR-SUL  SULFATE (PERCENT) %
133           KEY: '/7X,F4.1,4(5X,F4.1),16X,I8//30X'* * * * * * * * %
134    * *'//)
135   RETURN
136   END
```

SUBROUTINE REDUSC

This subroutine is designed to read a raw-data record placed in a separate system file for the purpose of processing the record and adding it to either the WCOAL or the ECOAL files. A listing of subroutine REDUSC appears at the end of this discussion. For maximum efficiency of the input/output routine on the INFONET system, this subroutine performs the read and decode operations as block operations rather than as implied DO operations.

Use of the SYCOR system for data input requires that, for transmission, a raw-data record must be broken up into separate blocks, each containing 256 or fewer characters (or 64, 4-byte words). For the purpose of executing these read operations, IBLOK1 has been dimensioned to 64 and IBLOCK2 dimensioned to 8. These two block read operations are executed in lines 126 and 131. For purposes of decoding, the INFONET system restricts a formatted input/output operation to 144 characters or less. Therefore, IBLOCK1 has been broken into two separate strings for decoding through NST1 (line 127) and NST2 (line 130). Before the second decoding operation takes place, the last 30 words of IBLOK1 are transferred into the first 30 positions of IBLOCK1 (lines 128–129). The string IBLOK2 is decoded into the block, NST3, KKK (the record key number), and XST (the real value associated with the last word of the data string).

```
100   SUBROUTINE REDUSC(NSTRNG,STRING,KKK,*)
101   %
102   %  THIS SUBROUTINE IS DESIGNED TO READ THE RAW DATA FILE
103   %  WHICH HAS BEEN TRANSMITTED VIA THE "SYCOR" TERMINAL TO
104   %  THE "INFONET" HOST COMPUTER FOR ENTRY INTO THE "WCOAL"
105   %  AND "ECOAL" MASTER FILES.
106   %
107   %  THIS ROUTINE IS, BY NECESSITY, PECULIAR IN TWO RESPECTS:
108   %      1)  THE "ITS" SOFTWARE USED BY THE "SYCOR" TERMINAL
109   %          CAN ONLY TRANSMIT RECORDS OF 256 OR FEWER CHAR-
110   %          ACTERS.  THIS MEANS THAT THE RAW DATA FILE RECORD
111   %          HAS BEEN BROKEN UP INTO TWO CHARACTER STRINGS,
112   %          ONE WITH A LENGTH OF 64 WORDS AND THE SECOND WITH
113   %          A LENGTH OF 8 WORDS, FOR TRANSMISSION AND, HENCE,
114   %          READING PURPOSES.
115   %      2)  THE "INFONET"  "READ"/"DECODE" ROUTINES WILL
116   %          ACCEPT ONLY STRINGS OF 144 CHARACTERS OR LESS IN
117   %          THE FORMATTED DATA MODES.  THIS INCLUDES THOSE
118   %          PORTIONS OF THE DATA STRING WHICH ARE SKIPPED VIA
119   %          THE X-FIELDS IN THE FORMAT.  THIS MEANS THAT THE
120   %          RECORDS MUST BE BROKEN DOWN FURTHER INTO BLOCKS
121   %          OF 144 CHARACTERS OR LESS.
122   %
123         DIMENSION      NSTRNG(400),   STRING(400),   IBLOK1(64),
124         IBLOK2(8),     NST1(38),      NST2(32),      NST3(4)
125         EQUIVALENCE   (STRING,NSTRNG)
126         READ(31,END=999) IBLOK1
127         DECODE(IBLOK1,1001,ERR=997) NST1
128         DO 10 I=1,30
129   10    IBLOK1(I)  =  IBLOK1(I+34)
130         DECODE(IBLOK1,1004,ERR=997) NST2
131         READ(31,END=999) IBLOK2
132         DECODE(IBLOK2,1005,ERR=997) NST3,KKK,XST
133   1001  FORMAT(10A4,I2,I3,A1,I3,A1,I2,I3,10A4,I1,10A4)
134   1004  FORMAT(I1,9A4,I1,3A4,A1,7A4,I2,8A4,I4)
135   1005  FORMAT(2A4,A1,I8,I7,F8.2)
136         DO 110 I=1,38
137   110   NSTRNG(I)  =  NST1(I)
138         DO 120 I=1,26
139   120   NSTRNG(I+38)  =  NST2(I)
140         NSTRNG(65)  =  '    '
141         DO 130 I=1,6
142   130   NSTRNG(I+65)  =  NST2(I+26)
143         DO 140 I=1,4
144   140   NSTRNG(I+71)  =  NST3(I)
145         STRING(76)  =  XST
146         RETURN
147   997   WRITE(6,1003)
148   1003  FORMAT(/' DATA ERROR.  CHECK DATA INPUT FILE.'/)
149         STOP
150   999   RETURN 1
151   END
```

SUBROUTINE REDUSA

This subroutine is designed to read a raw-data record placed in a separate system file for the purpose of processing the record and adding it to the USALYT file. A listing of subroutine REDUSA appears at the end of this discussion. For maximum efficiency of the input/output routine on the INFONET system, this subroutine performs the read and decode operations as block operations rather than as implied DO operations.

Use of the SYCOR system for data input requires that, for transmission, a raw-data record must be broken up into separate blocks, each containing 256 or fewer characters (or 64, 4-byte words). For the purposes of executing these read operations, IBLOK1 has been dimensioned to 34, IBLOK2 dimensioned to 32, and IBLOK3 dimensioned to 28. These three block read operations are executed in lines 127, 129, and 131. They are then decoded, IBLOK1 being translated according to format into NST1 (line 128), IBLOK2 translated into NST2 (line 130), and IBLOK3 translated into NST3, XST1, and KKK (line 132), where NST1, NST2, NST3 are blocks of integer and alphanumeric data, XST1 is a block of real data, and KKK is the record key value. These block values are then transferred into the proper positions of the NSTRNG and STRING arrays (lines 136 and 143), and a standard return is made to the calling subroutine. A STOP is reached following an error message if input difficulties have been encountered (lines 145–147).

```
100   SUBROUTINE REDUSA(NSTRNG,STRING,KKK,*)
101   %
102   %  THIS SUBROUTINE IS DESIGNED TO READ THE RAW DATA FILE
103   %  WHICH HAS BEEN TRANSMITTED VIA THE "SYCOR" TERMINAL TO
104   %  THE "INFONET" HOST COMPUTER FOR ENTRY INTO THE "USALYT"
105   %  MASTER FILE.
106   %
107   %  THIS ROUTINE IS, BY NECESSITY, PECULIAR IN TWO RESPECTS:
108   %      1)  THE "ITS" SOFTWARE USED BY THE "SYCOR" TERMINAL
109   %          CAN ONLY TRANSMIT RECORDS OF 256 OR FEWER CHAR-
110   %          ACTERS.  THIS MEANS THAT THE RAW DATA FILE RECORD
111   %          HAS BEEN BROKEN UP INTO TWO CHARACTER STRINGS,
112   %          ONE WITH A LENGTH OF 34 WORDS, THE SECOND WITH A
113   %          LENGTH OF 32 WORDS, AND THE THIRD WITH A LENGTH OF
114   %          28 WORDS FOR THE PURPOSE OF TRANSMISSION.
115   %      2)  THE "INFONET" "READ"/"DECODE" ROUTINES WILL
116   %          ACCEPT ONLY STRINGS OF 144 CHARACTERS OR LESS IN
117   %          THE FORMATTED DATA MODES.  THIS INCLUDES THOSE
118   %          PORTIONS OF THE DATA STRING WHICH ARE SKIPPED VIA
119   %          THE X-FIELDS IN THE FORMAT.  THIS MEANS THAT THE
120   %          RECORDS MUST BE BROKEN DOWN FURTHER INTO BLOCKS
121   %          OF 144 CHARACTERS OR LESS.
122   %
123         DIMENSION      NSTRNG(400),   STRING(400),   XST1(13),      %
124         NST1(37),      NST2(32),      NST3(12),      IBLOK1(34),    %
125         IBLOK2(32),    IBLOK3(28)
126         EQUIVALENCE   (STRING,NSTRNG)
127   100   READ(31,END=999) IBLOK1
128         DECODE(IBLOK1,1001,ERR=99) NST1
129         READ(31,END=999) IBLOK2
130         DECODE(IBLOK2,1002,ERR=99) NST2
131         READ(31,END=999) IBLOK3
132         DECODE(IBLOK3,1003,ERR=99) NST3,XST1,KKK
133   1001  FORMAT(4A4,6A4,I2,I3,A1,I3,A1,I2,I3,4A4,6A4,4A4,6A4)
134   1002  FORMAT(4A4,5A4,3A4,A1,7A4,3A4,5A4,I4,3A4)
135   1003  FORMAT(A4,I8,3I1,A1,I3,A1,1X,I5,3(1X,I4),F3.1,12(1X,F4.1),I8)
136         DO 110 I=1,37
137   110   NSTRNG(I)   =  NST1(I)
138         DO 120 I=1,32
139   120   NSTRNG(I+37)  =  NST2(I)
140         DO 130 I=1,12
141   130   NSTRNG(I+69)  =  NST3(I)
142         DO 140 I=1,13
143   140   STRING(I+81)  =  XST1(I)
144         RETURN
145   99  WRITE(6,1004)
146   1004  FORMAT(/' DATA ERROR. CHECK DATA INPUT FILE. '/)
147   STOP
148   999 RETURN 1
149   END
```