

TIS FILE
RECORD COPY

SRL-NURE HYDROGEOCHEMICAL DATA MANAGEMENT SYSTEM

**J. H. Maddox, H. F. Wren, H. C. Honeck,
C. R. Tharin, and M. D. Howard**



**SAVANNAH RIVER LABORATORY
AIKEN, SOUTH CAROLINA 29801**

PREPARED FOR THE U.S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION UNDER CONTRACT AT(07-2)-1

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America

Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

Price: Printed Copy \$4.50; Microfiche \$2.25

662618

DP-1432

Distribution Category: UC-10

SRL-NURE HYDROGEOCHEMICAL DATA MANAGEMENT SYSTEM

by

J. H. Maddox, H. F. Wren, H. C. Honeck,
C. R. Tharin, and M. D. Howard

Approved by

J. W. Stewart, II, Research Manager
Computer Applications Division

and

R. L. Folger, Research Manager
Analytical Chemistry Division

Publication Date: July 1976

**E. I. DU PONT DE NEMOURS AND COMPANY
SAVANNAH RIVER LABORATORY
AIKEN, SOUTH CAROLINA 29801**

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION UNDER CONTRACT AT(07-2)-1

ABSTRACT

A data management system was developed to store and retrieve all physical, chemical, and geological data collected for the NURE Hydrogeochemical Reconnaissance program by the Savannah River Laboratory (SRL). In 1975, SRL accepted responsibility for hydrogeochemical reconnaissance of twenty-five states in the eastern United States as part of the National Uranium Resource Evaluation (NURE) program to identify areas favorable for uranium exploration.

The SRL-NURE hydrogeochemical data management system is written in FORTRAN IV for an IBM System 360/195 computer. The system is designed to accommodate the changes in the types of data collected about a sampling site and for the different numbers of samples taken at the sites. The data are accepted as they become available and are combined with relevant data already in the system.

CONTENTS

Introduction	5
Major Data Processing Steps	6
Special Design Considerations	7
Input Data	7
Data Volume	9
Output Criteria	9
File Structure	9
Data Types and Data Organization	9
Segmentation of Data	10
Segment Types	12
Segment Names	13
Data Element Names	13
Formal and Informal Names	14
System Tables	14
Scrub Tables	15
Process Control Tables	16
Table-Coded Data Tables	18
QUERY Operation	19
Specification of a QUERY	19
Printed Output	20
Appendices	
A - Data Types	21
B - Segment Structures and Addresses	23
C - UPDATE Function	26
D - QUERY Function	38
E - Modules to Maintain System Tables	47

LIST OF FIGURES

- 1 SRL-NURE Hydrogeochemical Data Management System: Overall Processing of Data 7
- 2 UPDATE Function for SRL-NURE Hydrogeochemical Data Management System 8
- 3 Physical File Structure 10
- 4 Logical Record Hierarchy 11
- 5 Typical Record Structure for Stream Site 11
- 6 Typical Record Structure for Ground Water Site 12
- 7 Examples of Formal and Informal Segment Names 14

LIST OF TABLES

- 1 Segment Definition 12
- 2 Portion of a Scrub Table 15
- 3 Definition of Scrub Options 16
- 4 Portion of Typical Process Control Table 17
- 5 Table-Coded Data Table 18
- 6 Typical QUERY for Sample Site Codes 20

SRL-NURE HYDROGEOCHEMICAL DATA MANAGEMENT SYSTEM

INTRODUCTION

The National Uranium Resource Evaluation (NURE) program was established to evaluate domestic uranium resources in the continental United States and to identify areas favorable for uranium exploration. The Grand Junction Office of the Energy Research and Development Administration (ERDA) is responsible for administering and coordinating NURE program efforts. Input to the NURE program comes from ERDA prime contractors, ERDA-sponsored research and development, the uranium industry, U.S. Geological Survey, U.S. Bureau of Mines, other government agencies, and independent sources.

In 1975, the Savannah River Laboratory (SRL) accepted responsibility for hydrogeochemical and stream sediment reconnaissance of twenty-five states in the eastern United States. Oak Ridge Gaseous Diffusion Plant (ORGP), Los Alamos Scientific Laboratory (LASL), and Lawrence Livermore Laboratory (LLL) have accepted responsibility for similar reconnaissance surveys in the rest of the continental United States, including Alaska. Variations in uranium and associated elements in surface and underground waters and in stream and lake sediments will be determined as guides for uranium search.

SRL personnel will conduct two types of surveys for each geological area. The first phase, or orientation study, will cover a small area with a relatively high sampling density (large number of sites sampled per unit area). Detailed data will be gathered about each sample site, and several samples will be taken at each site. The second phase, or reconnaissance, will gather only the data items identified as relevant by the orientation study; the sampling density will be lower, and fewer samples will be taken per site.

About 170,000 sites will be sampled through the SRL hydrogeochemical reconnaissance program. This will result in ~150 million bytes of data that must be stored, retrieved, and analyzed. The hydrogeochemical data management system is designed to store and retrieve the data. The system prints reports tabulating all or subsets of the data that are stored. The system can retrieve subsets of data and make them available to statistical programs for analysis. This report describes the SRL-NURE hydrogeochemical data management system.

The NURE program is a nationwide, multidisciplinary effort. References to "NURE data" or "NURE data management system" in this report generally refer to data collected by SRL for the hydrogeochemical reconnaissance portion of the NURE program and to the system developed to manage that data set.

MAJOR DATA PROCESSING STEPS

The major data processing steps required of the NURE data management system are shown in Figure 1. Input data both from cards and from measurement analysis programs are used by a program (UPDATE) to update an old master file and produce a new master file. Information is selected and extracted from the master file and reformatted by a module (QUERY) producing reports and records stored in JOSHUA data sets. JOSHUA is an operating system that provides the data handling services required to accomplish extensive, iterative, scientific calculations on a large, semipermanent data base.* JOSHUA contains several modules that perform mathematical operations such as regression analysis.

The UPDATE program maintains the NURE master file. This includes adding a new record to the file, adding data to or correcting portions of an existing record, and deleting a complete record. For the NURE master file, a record contains all data pertaining to a survey site plus the data that result from the analysis of the samples taken at the site.

The UPDATE program is separated into two modules (Figure 2). The structure of these modules is described in Appendix C. The Input Scrub and Sort module scrubs the input for syntactical errors and builds a file of syntactically correct input for the UPDATE module. The UPDATE module reads the input and the old master file. The records from the master file are changed to reflect the input and then written to the new master file.

The UPDATE module provides the flexibility required to accommodate anticipated changes in input, file structure, and similar details of the NURE data management system. The key to this flexibility is to control such details through a set of system tables defined by the user, which are stored in separately maintained JOSHUA files.

The function of the QUERY module (Figure 1) is to extract data from NURE master records and write the data in a form for processing by other modules, e.g., JOSHUA numerical analysis modules, and in reports. The QUERY module reads each record in the master file to determine if it meets the selection criteria supplied by the user. Output from QUERY consists of printed reports and JOSHUA records containing data fields as specified by the user. These records serve as input to various analysis programs which are not part of the data management system.

* H. C. Honeck. *The JOSHUA System*. USERDA Report DP-1380, Savannah River Laboratory, E. I. du Pont de Nemours and Co., Aiken, South Carolina (1975).

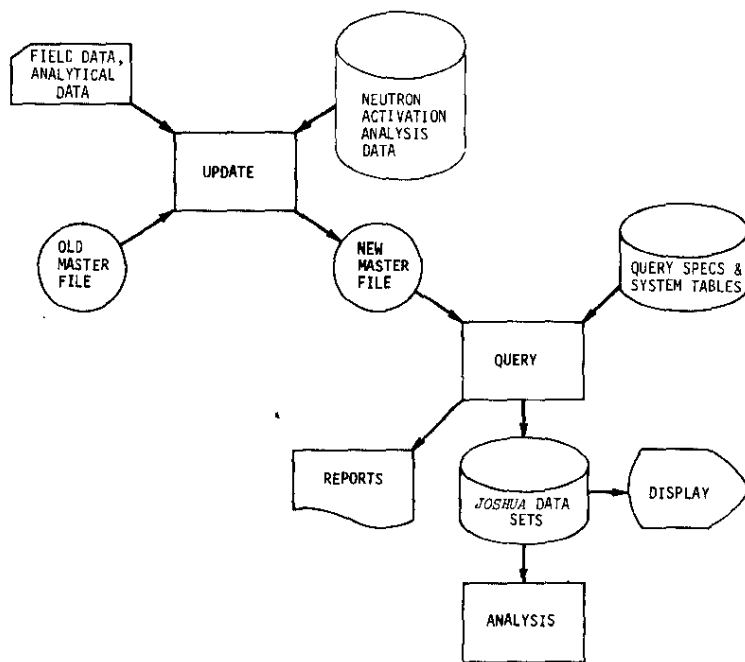


FIGURE 1. SRL-NURE Hydrogeochemical Data Management System: Overall Processing of Data

The current version of QUERY limits the number of items defining the records to be retrieved. The items defining the retrieval of records are survey site, sample source, sample type, and type of measurement.

The methods for specifying which data are to be retrieved and the criteria for selection are described in a later section (Specification of a QUERY, p 19). The structure of the QUERY module is presented in Appendix D.

The system tables that control the reading, scrubbing, and storing of NURE data are maintained by support modules (Appendix E). These support modules read cards, build the tables, and store them in a JOSHUA data set. The modules also read the tables from the JOSHUA data set and produce a printed report.

SPECIAL DESIGN CONSIDERATIONS

Input Data

Collection and analysis of data for the NURE program will span five or more years. As the program evolves, the type and volume of data maintained about a sample site will change. With time, the field sampling teams may be recording different data about sample sites, the receiving laboratory personnel may be recording different information about sample preparation, and the chemical analysis technicians may be analyzing the samples

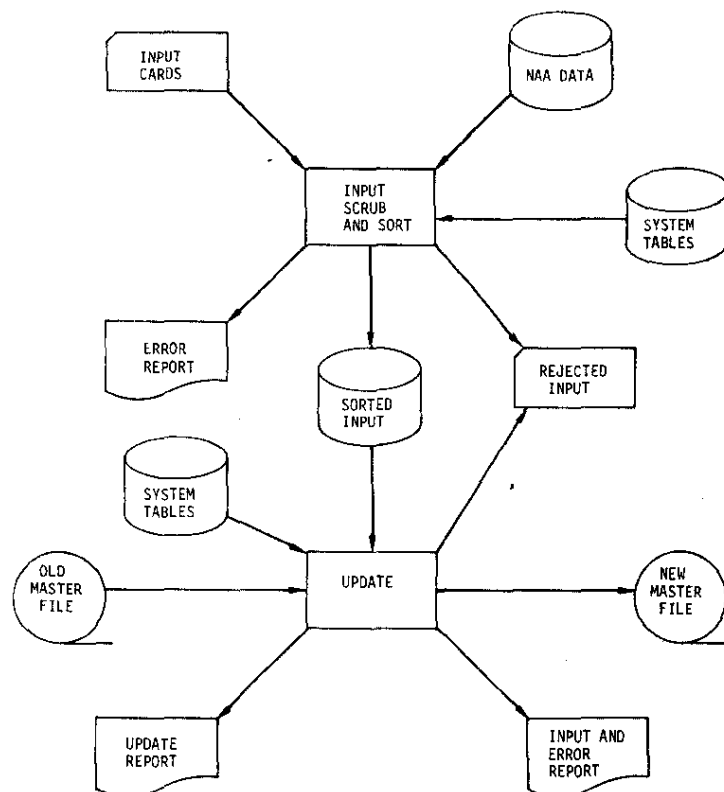


FIGURE 2. UPDATE Function for SRL-NURE Hydrogeochemical Data Management System

for different elements. In addition, the orientation studies will collect more data about the sample sites than the wide-area studies.

If data items are dropped from the input, the structure of the master record does not have to change. The data field in the record can be set to "no data." But, as data items are added to the input, they must be added to the master record. Therefore, the record structure must be flexible.

In order to be cost-effective and reliable, the data management system must be able to cope with continually changing input formats and master record structures. System tables have been developed to direct the processing and storage of the data. These tables are called Scrub Tables, Process Control Tables, and Table-Coded Data Tables and are described in the System Tables section (p 14).

All the sample data are not available simultaneously. The site data recorded in the field are the first data available. Several weeks or months later, the receiving laboratory prepares the samples for chemical analysis and records these data. Even later, the results of the analyses become available for input.

If the time lag were short enough, these data could be collected and presented to the data manager as a single entry. However, the time lag may be as long as six months. The best way to prevent data loss is to add the data to the master file as it becomes available. Therefore, the data manager must be able to build the master file records in segments. Since there is a natural sequence for data arrival at the computer, the data manager monitors the data for sequence and tells the user precisely what data are missing.

Data Volume

It is estimated that there will be about 170,000 sample sites, and that about 900 bytes of data will be recorded for each site, producing about 150 million bytes of data. Because of the expected data volume, the NURE data management system is designed to maintain sequential tape files.

Output Criteria

NURE is an interlaboratory program. Each regional laboratory (LASL, LLL, ORGDP, SRL) must be able to provide the national NURE data center with data files on magnetic tape. These data files and printed reports must be in a form suitable for reporting in the open literature.

SRL's NURE data manager maintains internal files in a format that is efficient for the updating and retrieval of data. The NURE system will generate the required external reports and tape files on command.

FILE STRUCTURE

Data Types and Data Organization

Sample collectors provide field measurements of each sampling site, collect samples, and send the description and samples to SRL. The samples are then prepared for analysis. The results of the analysis are element concentrations. The four major types of data collected are:

- Site Location
- Site Description
- Sample Description
- Analytical Results

There are several types of sites (stream, well, etc.), each having unique descriptive data. There are several types of samples (water, filtrate, sediment, etc.) and preparation methods, each having its own descriptive data; and several types of analysis (neutron activation, gamma activity, mass spectrometer, etc.), each resulting in a list of element concentrations.

Segmentation of Data

These data fall into a natural organization called the Physical File Structure (Figure 3). A *record* contains all data for a site, with the site name as the record name. A collection of these records is a *file*. Each record consists of *segments*. Each segment consists of a header portion (control and accounting information) and a data portion.

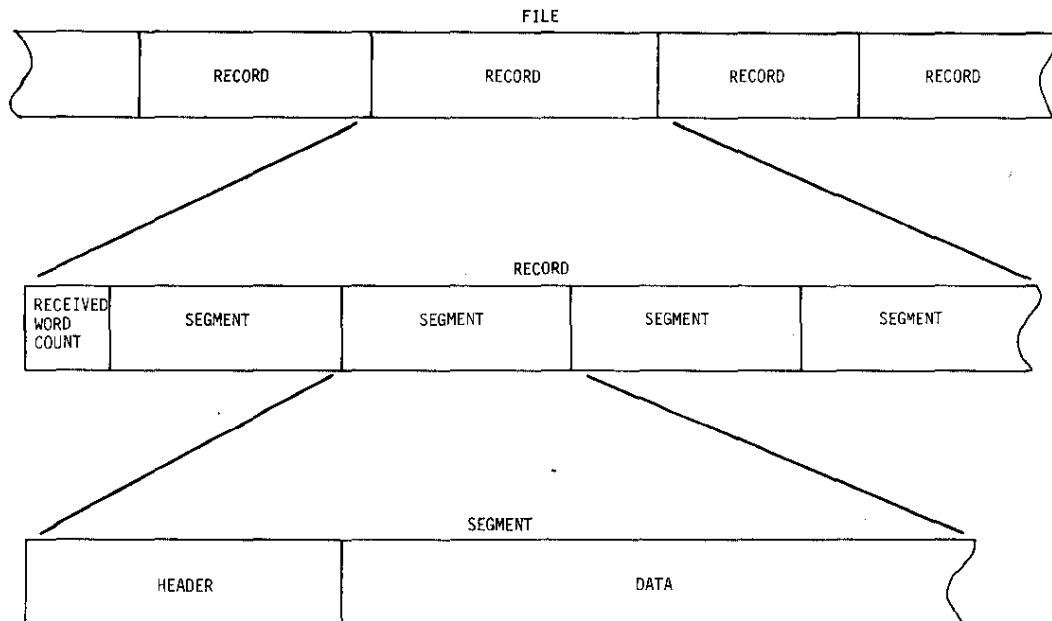


FIGURE 3. Physical File Structure

While segments are physically stored in a linear fashion, they can be viewed logically as a hierarchal structure (Figure 4). The header portion of each segment includes "pointers" that point to the next segment at the same level (brother) and to the first segment at the next lower level (son). Comparison of this logical structure with the four major collections of NURE data indicates that each collection becomes a level of the hierarchy. Two typical record structures are illustrated in Figures 5 and 6. Segment structures are discussed more fully in Appendix B.

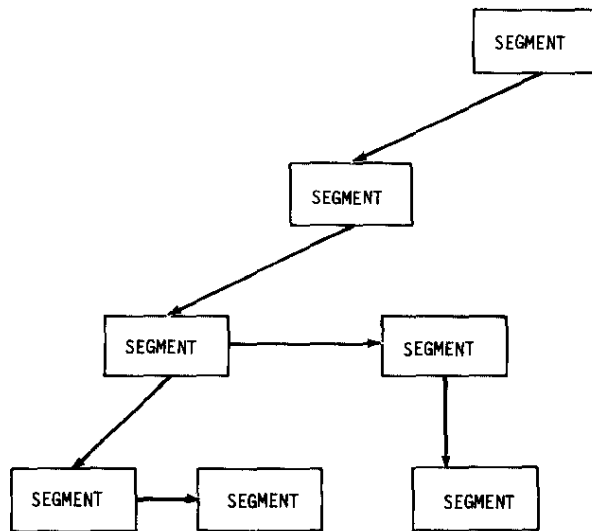


FIGURE 4. Logical Record Hierarchy

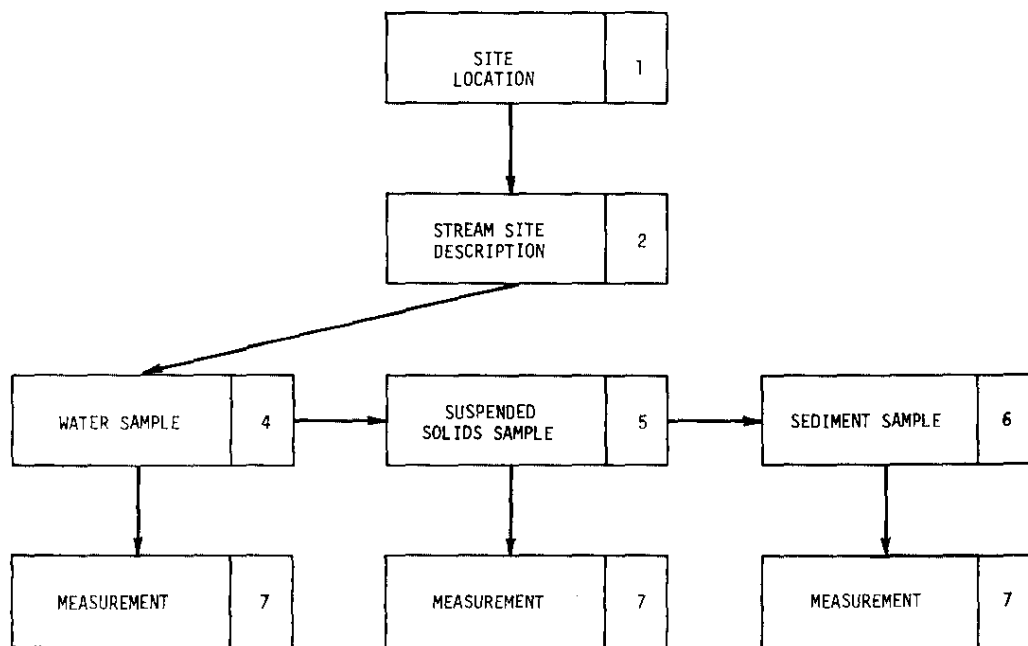


FIGURE 5. Typical Record Structure for Stream Site

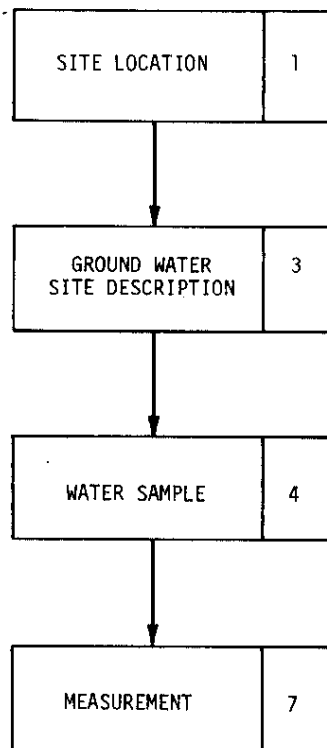


FIGURE 6. Typical Record Structure for Ground Water Site

Segment Types

Seven types of segments are required (Table 1). Other types can be added as needed.

TABLE 1

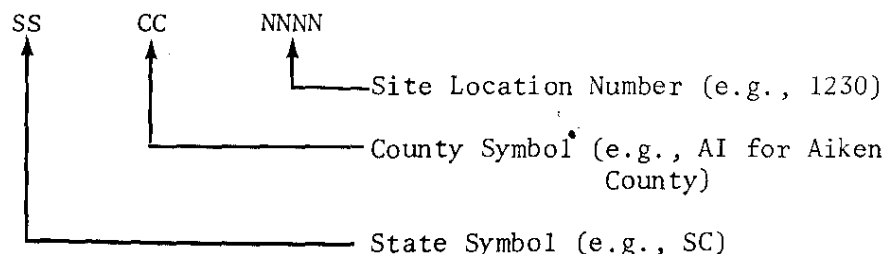
Segment Definition

<i>Segment Type</i>	<i>Segment Description</i>
1	Site Location
2	Stream Site Description
3	Ground Water Site Description
4	Water Sample
5	Suspended Solids Sample
6	Sediment Sample
7	Measurement

A segment type may occur more than once within the same level (e.g., multiple water samples, multiple measurements on the same sample, etc.).

Segment Names

Each segment has an 8-character name qualifier. The site-location name qualifier is the record name, which has the form:



The name qualifiers for the remaining levels are arbitrary, but they must be unique at each level.

Concatenation of name qualifiers uniquely references a segment and is the segment name. For example,

SCAI1230.STREAM

SCAI1230.STREAM.WATER

SCAI1230.STREAM.WATER.NAA

reference the left-most segments in the lower levels of Figure 5.

In some cases, the segment name qualifier implies the segment type (e.g., STREAM implies Segment Type 2). This is because only one occurrence of that segment type is allowed. When multiple segments of the same type occur, the name qualifiers must be different (e.g., WATER1, WATER2, and WATER3 may be the name qualifiers for three occurrences of Segment Type 4).

Data Element Names

If S is a concatenated segment name, and E is the name qualifier of a data element in S, then S.E is the complete data element name. If the element has multiple occurrences in the segment, then the occurrence number can be appended to complete the name (e.g., S.E1, S.E2, etc.).

Formal and Informal Names

The formal naming structure is essential for organizing the data into records and maintaining the data. However, this structure is cumbersome for preparing input data. A shorthand or informal naming convention identifies input and is translated by tables, algorithms, and implication into the formal name structure. An example of formal and informal segment names is given in Figure 7.

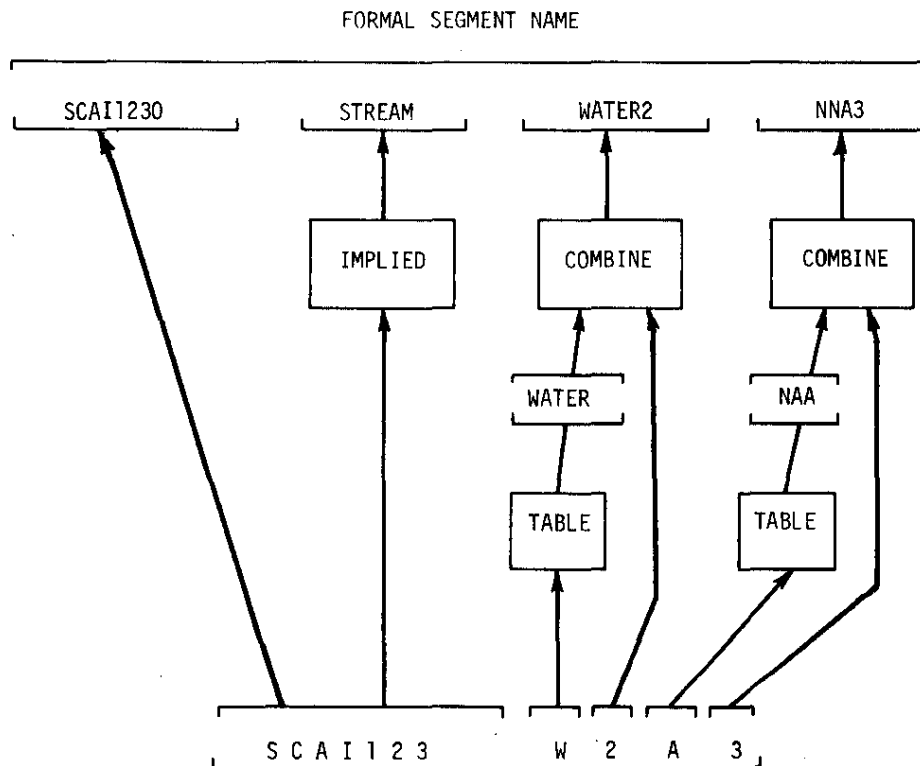


FIGURE 7. Examples of Formal and Informal Segment Names

SYSTEM TABLES

System tables are collections of "semipermanent" input data. These are of three general types, called Scrub Tables, Process Control Tables, and Table-Coded Data Tables. Scrub Tables define validation criteria for the primary input data (site, sample, and analysis data). Process Control Tables define rules for transforming input data from its input format to its storage (master file) format. Table-Coded Data Tables define standard values and labels for some data. Each is discussed in more detail below.

The basic function of system tables is to provide the flexibility to alter the formats of input and output data without having to reprogram. This flexibility is provided by programming the formatting and scrubbing rules in terms of general parameters and supplying values for these parameters as input. Changes in formats are expected, but not very frequently; hence the "semipermanent" nature of this input data.

System tables are built and maintained by separate modules (described in Appendix E) that write the information into JOSHUA data sets. The UPDATE and QUERY modules then read the information as part of their input.

Scrub Tables

Scrub Tables direct the scrubbing of the input cards by the Input Scrub and Sort Module. Each table is identified by a type number, as is each input card. The type number from the input card locates the proper table. The Scrub Table (Table 2) defines a scrub option for each column in the input card. In this case, Columns 1, 2, 3, and 4 are Option 1. This option represents any alphabetic character (Table 3).

The user or a programmer must define a Scrub Table for each card type. If the format of a card changes, the Input Scrub and Sort module does not have to be changed. Only the Scrub Table is changed. But, if a new scrub option should be needed, the option must be added to the module by reprogramming.

TABLE 2

Portion of a Scrub Table

<i>Card Column</i>	<i>Scrub Option</i>	<i>Card Column</i>	<i>Scrub Option</i>	<i>Card Column</i>	<i>Scrub Option</i>
1	1	5	4	9	0
2	1	6	5	10	5
3	1	7	5	11	6
4	1	8	9	12	4

TABLE 3

Definition of Scrub Options

<i>Option</i>	<i>Character String</i>
0	No scrub
1	A-Z
2	A-Z or blank
3	0
4	0-9
5	0-9 or blank
6	0-9, blank or -
7	0-9, blank or A-Z
8	1-9
9	Blank
10	Decimal or blank
11	Blank, 0, 1, or 2
12	Decimal
13	+, -, or blank

Process Control Tables

Process Control Tables relate data fields in the input card to locations in the master file. These tables are used by the UPDATE module to read, scrub, and store data in the master file. The QUERY module uses the same tables to locate the data fields to be retrieved. A Process Control Table must be defined for each card type. The first 56 columns of a NURE input card for a Process Control Table are shown on Table 4. The "Definition" column of the table is the annotated name of the data field whose position is defined by the "Columns, Start End" columns.

The "Data Type" column of the table defines how the data in the associated data field is to be processed and represented. A data type is presupposed to have a specific processing pattern, no matter in what input card type or field within a card it originated. Data type defines the internal and external representation of that data field. (See Appendix A, Table A-1.) The data manager translates the data from one representation to another.

TABLE 4

Portion of Typical Process Control Table

Definition	Columns		Data Type	Seg Type	Byte Start	Proc
	Start	End				
State	1	2	TS	0	0	0
Sampdate	9	14	DT	1	9	0
Samptime	15	18	IN	1	11	0
Teams	19	21	TS	1	13	0
Latitude	22	28	FX	1	1	0
Longitud	29	35	FX	1	5	0
Strwidth	41	43	FX	2	5	0
Strdepth	44	47	FX	2	9	0
Velocity	48	51	FX	2	13	0
Strshape	52	0	TC	2	17	0
Strbotom	53	0	TC	2	19	0
Strbotom	54	0	TC	2	21	0
Strbed	55	0	TC	2	23	0
Strlevel	56	0	TC	2	25	0

The "Seg Type" and "Byte Start" columns of the table define where the field is stored in the master record. If the field cannot be processed in the standard manner for that data type, the special processing column "Proc" of the translation table has a numeric entry. Special programming must be included in the data manager to process that field correctly.

To illustrate the above: Assume that a Card of Type 10 has been read in as a character string (78 characters long), and Columns 1-8 have been processed. Line 2 describes one data item or field in the card. The field name is Sampdate, and it starts in Column 9 and ends in Column 14. The field name is 6 columns long and has a data type 'DT'. As shown in Appendix A, DT represents date and is a 6-column integer of the form MMDDYY. The input character string is translated to an integer and separated into three two-digit numbers; the numbers can now be scrubbed. The MM portion must have values between 1 and 12 inclusive and DD between 1 and 31 inclusive. After scrubbing, the date is converted to an integer Julian sequence number. Next, the date must be stored in the master record in Segment Type 1 starting in Byte Position 9. The Data Type 'DT' is stored as a 2-byte integer. So Sampdate is stored in Segment 1, Bytes 9 and 10.

The other fields in the card are processed in the same manner. Each field is located and translated into a format for scrubbing, the appropriate scrub is performed, and the data value is stored in its proper segment.

The Process Control Tables are actually JOSHUA records that the data manager reads and stores in tabular form. If the variation in NURE input data requires that a new card type or a new data type be added to the system, it can be done by adding the required JOSHUA record(s) and adding the required coding to the data manager without disturbing any of the presently available, checked-out coding.

As long as the special processing is done for less than about 10% of the data, the table-driven technique of translating the input data into the file format is cost-effective and easy to maintain.

Table-Coded Data Tables

Table-Coded Data Tables (Table 5) were devised to satisfy a variety of needs. For field personnel, the tables serve as multiple-choice lists, which allow them to describe site or sample qualifications by checking (or writing) a one-digit code on the input form, rather than writing out the qualification in words. For the data analysis, the tables provide a means of defining standard units and increments for such qualifications.

TABLE 5

Table-Coded Data Table

Table Name	- WELFREQ	
Table Description	- Use Frequency of Well	
<i>Code</i>	<i>Shorthand</i>	<i>Description</i>
1	Constant	Continuous (more than once-a-day)
2	Daily	Once-a-day
3	Weekly	Once-a-week
4	Monthly	Once-a-month
5	6 Months	Once every six months
6	Yearly	Once-a-year
7	Rarely	Less than once a year

For the computer program, the tables provide a flexible (user-controlled) medium for prescribing report labels. (The "shorthand" and "description" labels are used in different reports.) They also provide standard lists of allowable input data (either the "code" values or the "shorthand" labels) which are used to scrub data from the field. Finally, the "code" values allow very compact storage of site and sample qualifications in the master file.

QUERY OPERATION

The QUERY module retrieves data from the NURE master file. The user defines the data to be retrieved, its selection criteria, and the type of output desired. QUERY produces two types of output: printed reports and collections of data stored in JOSHUA data sets.

The user specifies up to 30 sets of selection criteria for retrieval of data from the NURE master file. Each record in the master file is compared to each set of criteria. If the record satisfies the criteria and printed reports are requested, the data are printed. If the user also wants JOSHUA output, one or more JOSHUA records are written.

All data that satisfy a specified set of criteria are written to a subset in the JOSHUA data base. Each such subset serves as input to the mathematical modules that analyze the data. (These JOSHUA analytical modules are not part of the data management system described here.)

Specification of a QUERY

The user must define the selection criteria and the type of data to be retrieved.

The present version of QUERY has a limited set of selection criteria. These include sample site code, sample source, sample type, and type of analytical measurement.

The sample site code is the index of the master record. The user is allowed to specify five sets of sample site codes for a single query. The sample request shown in Table 6 would retrieve data for the states of Georgia and Alabama and for any county in South Carolina that starts with the letter 'A'. The user could further restrict the type of data retrieved to be either from streams or ground water; to be for sample types of sediment, water, filter, resin, and/or magnetic properties; and to be for the analytical technique applied for determining the sample composition.

TABLE 6

Typical Query for Sample Site Codes

<i>Number of Sample Code Characters to Compare (1-7)</i>	<i>Sample Code Range</i>	
	<i>Low</i>	<i>High</i>
2	GA	GA
2	AL	AL
3	SCA	SCA
0		
0		

After specifying a set of selection criteria, the user must name the data items being retrieved. By referring to the Process Control Tables, the user can determine the name of each data item stored in a master record. Then, by simply listing the names of the items in the order they are to be written into a JOSHUA record, the desired information can be extracted from the master record. By specifying the atomic symbol, the user may also retrieve elemental concentrations for any site he desired. Any data item not named is not written to the JOSHUA data set.

Printed Output

A master record contains all the data about a sample site and the samples taken at that site. This volume of data cannot be printed in a single line of 132 print positions wide. Therefore, nine types of printed reports are used to display various groupings of data. The user is allowed to select what report(s) he wants. If a master record satisfies a set of selection criteria, data from the record are printed in each requested format.

The first type of output is a printout containing all the data stored about a sample site, naming the data field being printed, and following it with the information from the record. The second type of output is an abbreviated version of the first type and shows only sample site descriptive data. The remaining seven outputs are tabular reports of measured data. The first six tables display elemental concentrations, each output reporting a particular collection of elements. The seventh output tabulates field data (pH, conductivity, etc.).

APPENDIX A - DATA TYPES

The data types currently being used by the NURE data management system are shown in Table A-1. Each data type has an external representation (for input and output) and an internal representation (for storage in the master files). Comments on the non-standard types are given below:

Date (DT)

Date is a six-position number whose input format is month/day/year. The values of month and day must be in the ranges 1 to 12 and 1 to 31, respectively. If the date passes these scrub criteria, it is then converted to a Julian sequence number for storage in a master file record.

Table-Coded (TC, TS)

Data tables are used to represent semiquantitative, discrete-valued data of the sort shown in Table A-2. The three columns of a data table list three alternative representations of the same data value. The "code" is the usual form in which input from the field is presented, although the "shorthand" terminology is used in some cases. This is the distinction between TC (Table, Code) and TS (Table, Shorthand). "Shorthand" and "description" forms both serve as report labels. When "shorthand" input is supplied, the tables are consulted to translate to the numeric "code" representation. This number is then converted to I*2 form for internal storage. Maintenance of Table-Coded Data Tables is discussed in Appendix E.

TABLE A-1

Data Type Definitions

<i>Description of Type</i>	<i>Type Code</i>	<i>Representation</i>		<i>Print</i>
		<i>External</i>	<i>Internal</i>	
Integer	I4	I*4	I*4	XXXXXXXX
Integer, Short	IN	I*4	I*2	bbbXXXXX
Date	DT	I*4	I*2	MMDDYY
Floating Point	FX	R*4	R*4	XXX.XXXX
Character	CH	nL*1	nL*1	XXXXXXXX
Table-Coded:				
Code	TC	I*4	I*2	bbbXXXXX
Shorthand	TS	8L*1	I*2	XXXXXXXX

TABLE A-2

Table for Data Retrieval

Table Name - WELFREQ
 Table Description - Use Frequency of Well

<i>Code</i>	<i>Shorthand</i>	<i>Description</i>
1	Constant	Continuous (more than once-a-day)
2	Daily	Once-a-day
3	Weekly	Once-a-week
4	Monthly	Once-a-month
5	6 Months	Once every six months
6	Yearly	Once-a-year
7	Rarely	Less than once-a-year

APPENDIX B - SEGMENT STRUCTURES AND ADDRESSES

Structures

Each segment consists of a header portion and a data portion. Under the format in which data are stored in the master file, the data portion of a given segment follows immediately the header portion of that segment. But when stored in core, the header and data portions are stored in separate arrays.

The internal structure is the same for all header portions. The standard format is illustrated in Table B-1. The "words" referred to are the conventional four-byte words. Note that, in some cases, distinct information is stored in individual bytes of a single word. However, in one case (the segment name), a single item of information occupies two full words.

The internal structures of data portions differ from one segment type to another and may change with time for a given segment type. This structure is defined by the Process Control Tables (one of the system tables discussed in the text). The current structure of one particular segment type is illustrated in Table B-2.

Addresses

At the beginning of the preceding section, header and data portions of a segment were stated to be contiguous in the master file, but were stored independently in core. During storage in the master file, the data portion of a segment always begins 5 words (20 bytes) after the beginning of the header portion.

Relative addresses of related segments are included as part of the information in the header portion of a given segment. These are the "son" and "brother" pointers. One segment is called a "son" of another if it is directly subordinate. For example (see Figure 5 of the text), a water sample segment would be a "son" of a particular stream site segment if the water sample were taken from that stream site. One segment is called a "brother" of another if both qualify as "sons" of the same segment. For example (see Figure 5), a suspended solids sample segment would be a "brother" of a particular water sample segment if both samples were taken from the same stream site. Addresses for one "son" and one "brother" in the header portion of a given segment will suffice to locate all related segments. If several "sons" exist, only one need be listed (addressed) explicitly as a "son"; any others will be listed as "brothers."

A string of "brothers" is tied together by having each segment list the next "brother" in the string.

The addresses (called pointers) listed in the header portion of a segment actually represent the displacement of the cited segment with respect to the citing segment. Thus, if K is the address of the citing segment, which includes "son" pointer LSPT, then the address of its "son" segment is $K+1+LSPT$. The displacement is measured in units of words (4 bytes per word).

TABLE B-1

Segment Header Structure

<i>Word No.</i>	<i>Byte Start</i>	<i>Symbol</i>	<i>Data Type</i>	<i>Description</i>
1	1	NWRD	IN	Segment word count including header
	3	ITYP	CH	Segment type code
	4	IPRV	CH	Segment privacy code ^a
2	5	LBPT	IN	Brother pointer (word displacement)
	7	LSPT	IN	Son pointer (word displacement)
3	9	DSNM	CH	Segment name
5	17	MSED	DT	Segment entry date (Julian)
	19	MLMD	DT	Last modification date (Julian)

^a. The privacy code field has not been implemented.

TABLE B-2

Typical Data Segment Structure

Segment 1 - General Site Data

<i>Byte^a</i> <i>Start</i>	<i>WTH</i>	<i>Type</i>	<i>Description</i>
1	04	FX	Latitude
5	04	FX	Longitude
9	02	DT	Field sampling date
11	02	IN	Field sampling time
13	02	TC	Team number code
15	02	CH	Unused
17	04	FX	PH
21	04	FX	EH (millivolts)
25	04	FX	Temperature (°C)
29	04	FX	Specific conductivity (µmhos/cm)
33	04	FX	Dissolved oxygen (µg/l)
37	04	FX	Sulphate concentrations
41	04	FX	Phosphate concentrations
45	04	FX	Nitrate concentrations
49	04	FX	Ammonia concentrations
53	04	FX	Alkalinity (drops of sulfuric acid)
57	04	FX	Water in filtrate (ml)
61	04	FX	Water per 10g resin (ml)

^a. Relative to beginning of data segment.

APPENDIX C - UPDATE FUNCTION

This appendix outlines the logical structure of the SCRUB and UPDATE modules, and defines the function of each subroutine but does not present details. The source code listing provides the details. The function of each subroutine and highlights of areas of special interest are described below. All variables are defined in the programming.

Input Scrub and Sort Module

The functional hierarchy of the SCRUB module is shown in Figure C-1. The first function is to read and store the scrub specifications for the input cards. Next, an input card is read and printed. Then the card is scrubbed, column by column, according to its card type (found in Columns 79-80). If any errors are found in the card, a message is printed. After all the columns (1-78) have been scrubbed, the card is written to the SORT data set if no errors were found; otherwise, a card is punched.

When all the input cards have been processed, the SCRUB module processes the neutron activation analysis (NAA) data. These data are read, printed, and written to the SORT data set. No scrub checks are done.

The SORT data set record contains 97 bytes: Bytes 1-80 contain the input card image, and Bytes 81-97 contain a sort key. The sort key is a sample code (Bytes 81-92) and a numeric code (Bytes 93-97). For input cards, the sample code is obtained from Columns 1-8 and padded (on the right) with blanks for the sort key. The sample code in the NAA data is found in Columns 1-12 and is copied to the sort key. The numeric code portion of the sort key is the card code for input cards. The numeric sort code for NAA data is arbitrarily defined at 5000 and incremented by one for each NAA SORT record written.

Update Module

The UPDATE module performs a standard sequential file update. The overall hierarchy is shown in Figure C-2. The first function assigns initial values to (initializes) the COMMON areas and system tables. Then the updating process is started by reading a complete transaction from the SORT data set. A transaction contains all data about a single sample site. A record is then read from the old master file. The UPDATE module decides what data are to be written to the new master file by comparing the transaction and master record indices (sample site identifiers). A new record is created when the transaction index is lower than

the master record index. The master record is modified or deleted when the indices are equal. The master record is not changed when the master record index is less than the transaction index.

The COMMON areas and system tables are initialized by a group of subroutines (Figure C-3). The error messages are read from a table-coded JOSHUA record and stored (PUTERR). This allows the text of the error message to be changed without re-compiling the program. Next, the Process Control Tables are read and stored (PUTMAP). The data types for each card are scanned, and, if table-coded data types are present, subroutine READRC reads and stores the appropriate tables. The Process Control Tables are stored for future development of a table defining the layout of each segment (FIELDS). This table later determines the data fields in an existing segment that are being modified and decodes a segment for printing.

The update process schematic (Figure C-4) contains blocks, each representing a function performed by one or more subroutines. Figure C-4 also shows the logical relationship of these functions.

The function "Get Transaction" (Figure C-5) reads all of the input for a sample site and places the data in transaction data segments. Each time subroutine READTX is called, it initializes the transaction data segments to "no data" (CLRTRN). Then it reads, prints (READTX and HEADIN), and processes each card for a specific sample site. The card type and the sample code both develop the segment names (DECODE). The correct Process Control Table is found for each card type. The table defines for each data field:

- Location of data in the input card.
- External and internal representation.
- Scrub checks.
- Locations to store data in the transaction data segments.

Each field is converted from extended binary-coded-decimal interchange code (EBCDIC) to external representation and scrubbed by specialized subroutines (Figure C-6). If the field passes the scrub, it is stored in the proper transaction data segment (MOVANS). If the field does not pass, an error message is printed (ERROR, ERRORN).

If all data fields for the transaction pass the scrub checking, the complete transaction is returned. If any field does not pass the scrub, the complete transaction is punched (WRTRJT). The coding then cycles back to the top of subroutine READTX to process the next transaction.

After the first transaction has been processed, the record is read from the old master file (READMS). Next, the indices of the transaction and master record are compared. If the indices are equal, the master record is to be modified or deleted. Each transaction record has a data field that specifies if the master record is to be modified or deleted. If the action is to modify, the master record is converted from a linear string of segments that were read from the master file to a hierarchal arrangement of segments (see Segmentation of Data Section). This conversion is done by the UNPACK function (Figure C-7). An essential feature of UNPACK is the Expanded Record Storage, a set of labeled COMMON areas which contain the data in simple FORTRAN arrays.

The master record can be modified from the hierarchal format (Figure C-8) by locating each field in the transaction data segment and checking the field to see if a new value is present. Each new value is located and copied to an identical position in a corresponding master record segment (CORECT).

Only sample weight data and elemental weight measurement data are not processed in this manner. The elemental data are stored in the master file as concentrations but are measured as weights. Before storing, the elemental data are converted to concentrations by dividing each elemental weight by the appropriate sample weight. If a sample weight is modified and elemental data for the sample are present in the system, the elemental concentrations must be recalculated (RECALC). When elemental concentrations are being modified, the concentrations must be calculated. The subroutine PUSHDN adds a new element to an existing list.

The modified or newly created master record is then printed (Figure C-9). The subroutine PRTALL locates each data field and then builds a print line. The subroutines CONVRT and SDESCP convert a value from its internal representation to an 8-character EBCDIC field. Subroutine EDITMF prints the line of data passed to it and maintains line and page count.

The master record must be converted from the hierarchal arrangement of segments to a linear string of segments before it can be written to the new master file. This conversion is done by the function PACK (Figure C-10). The new master record is then written to the master file by subroutine RITEMS.

Master records are created if the transaction index is less than the master record index. The CREATE function (Figure C-11) is similar to the previously discussed MODIFY function. The main difference is that CREATE initializes the Expanded Record Storage area for the new master file record.

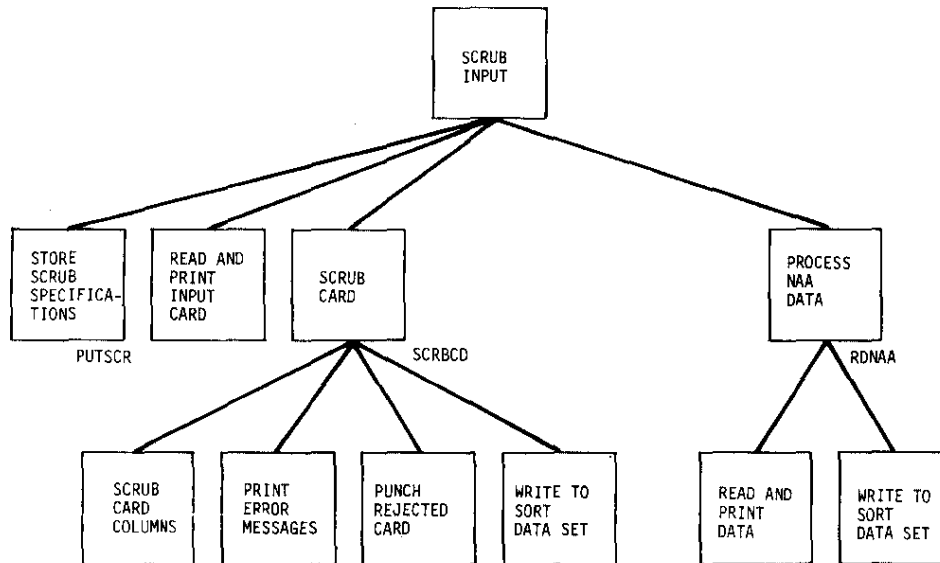


FIGURE C-1. Functional Hierarchy of SCRUB Module

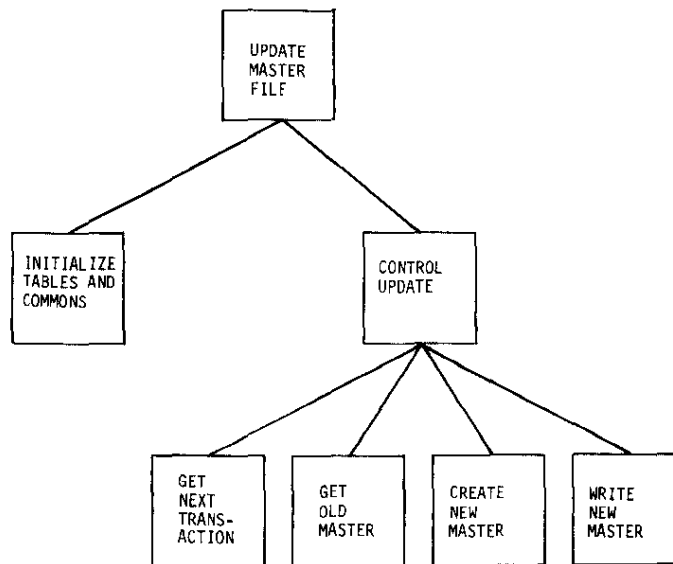
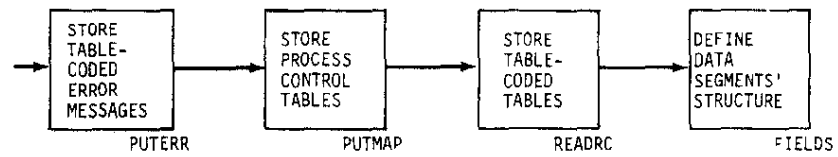


FIGURE C-2. Hierarchy of UPDATE Module

A. FUNCTIONAL FLOW



B. HIERARCHY

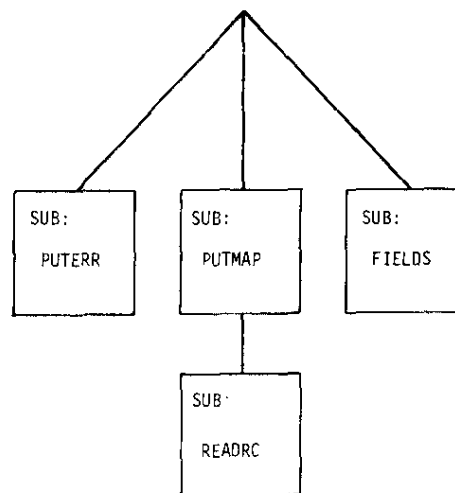


FIGURE C-3. Subroutines for Initializing System Tables and COMMON Areas

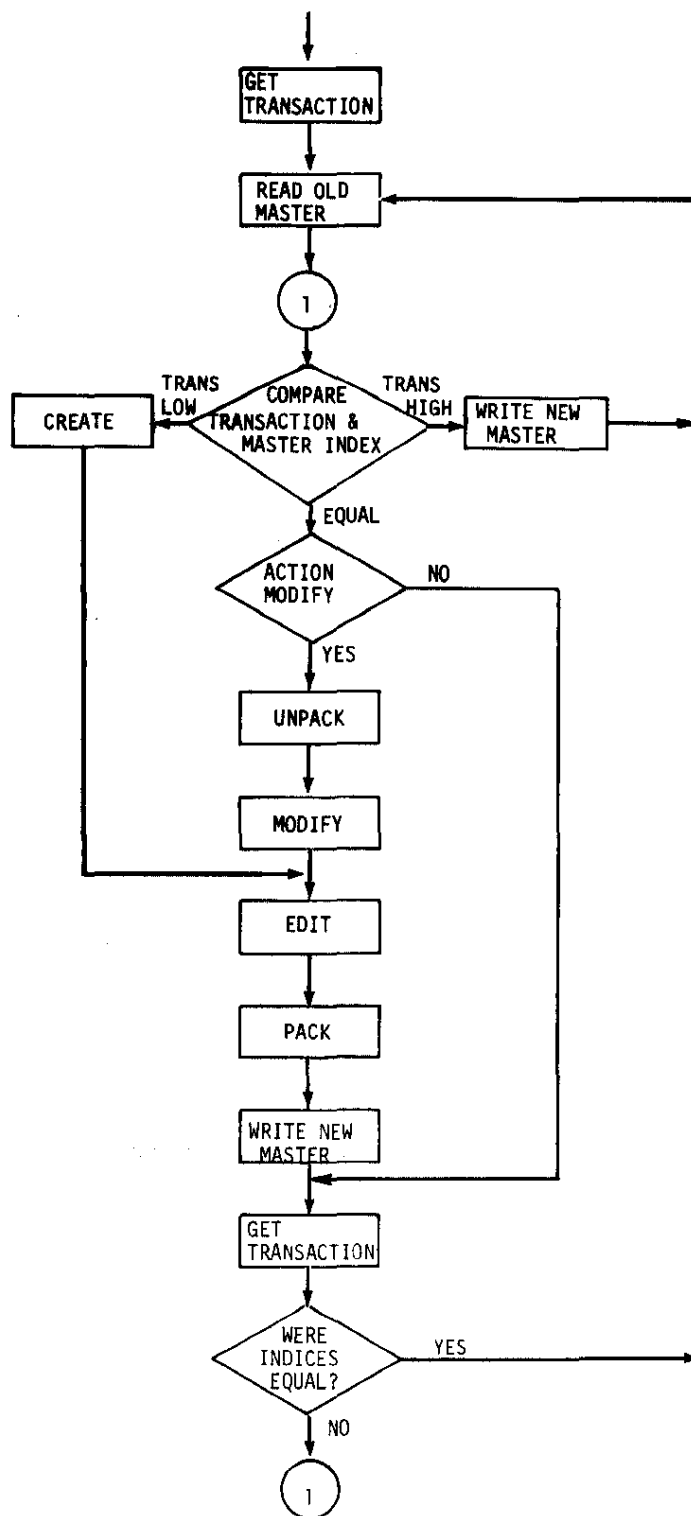
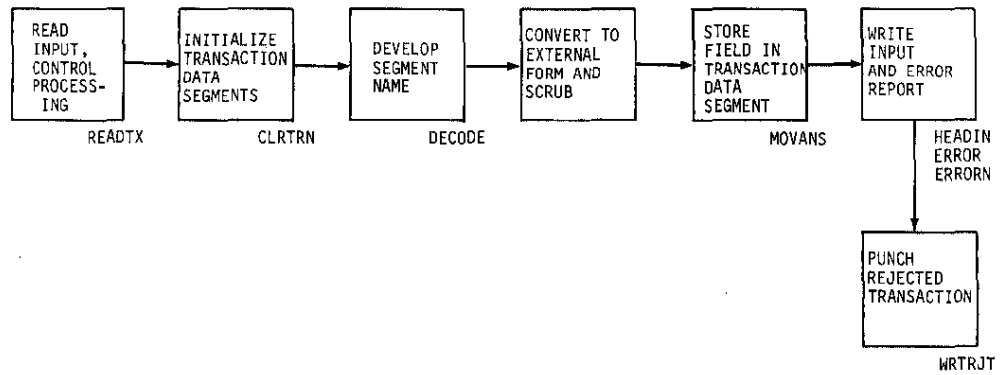


FIGURE C-4. Control UPDATE (Subroutine CONTRL)

A. FUNCTIONAL FLOW



B. HIERARCHY

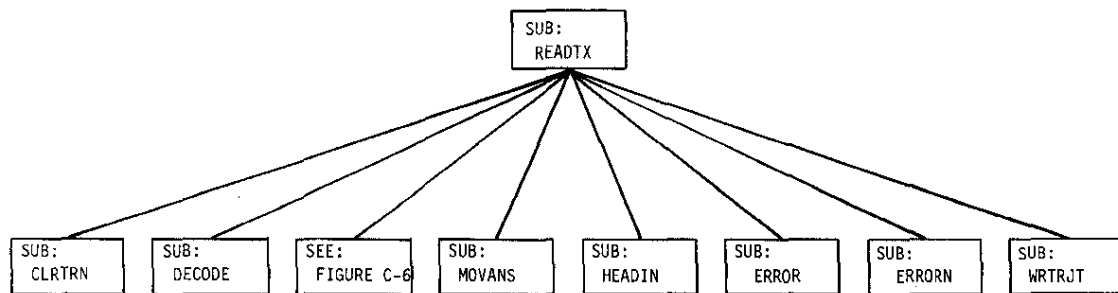


FIGURE C-5. "Get Transaction" Function

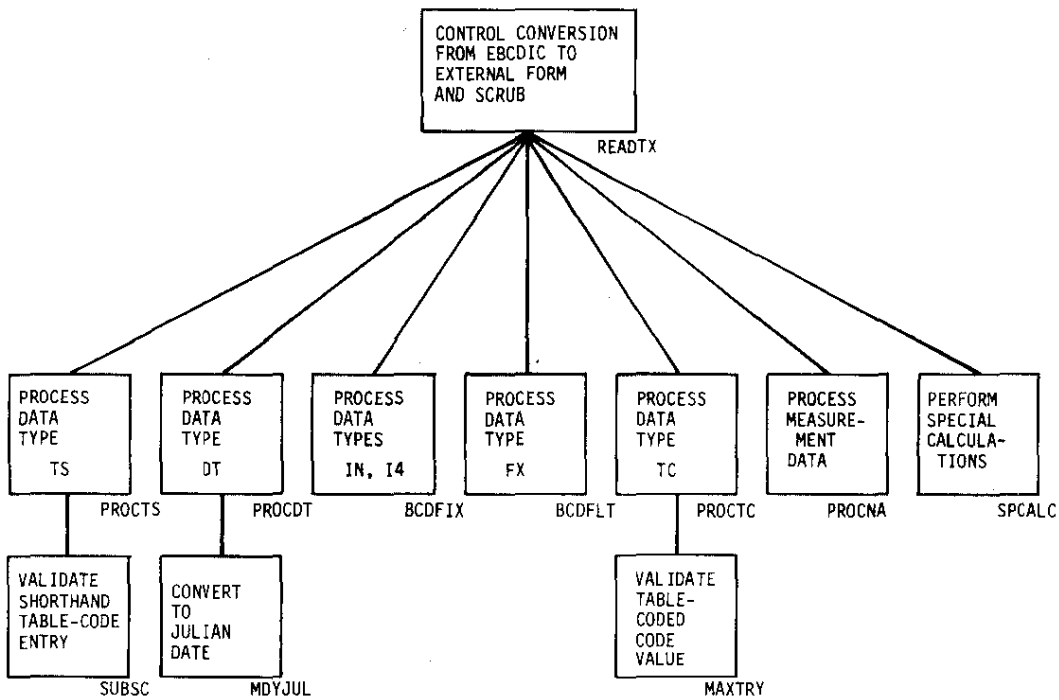
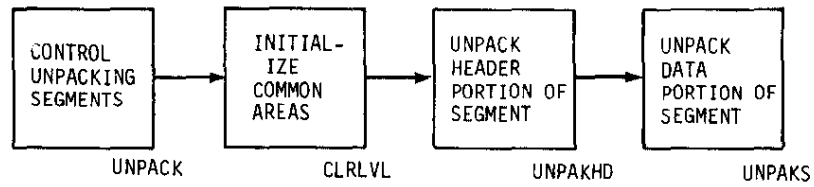


FIGURE C-6. Specialized Subroutines for Data Conversion and Scrubbing

A. FUNCTIONAL FLOW



B. HIERARCHY

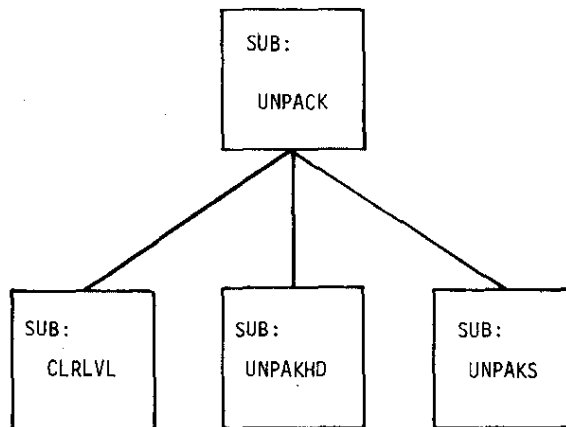
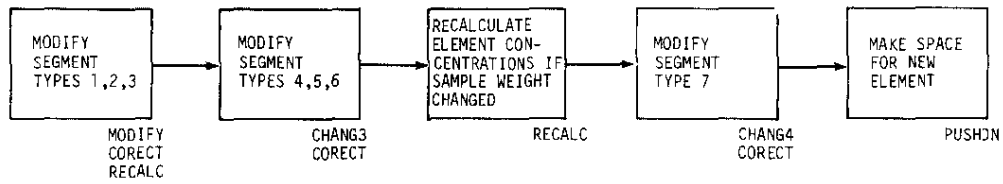


FIGURE C-7. Converting From Sequential Structure to Hierarchical Structure

A. FUNCTIONAL FLOW



B. HIERARCHY

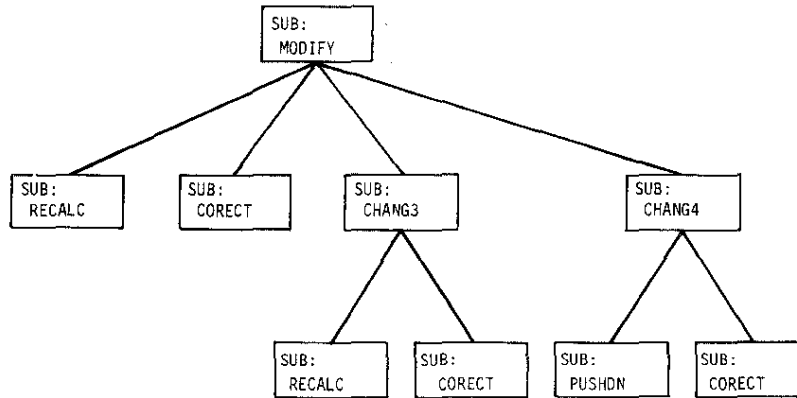
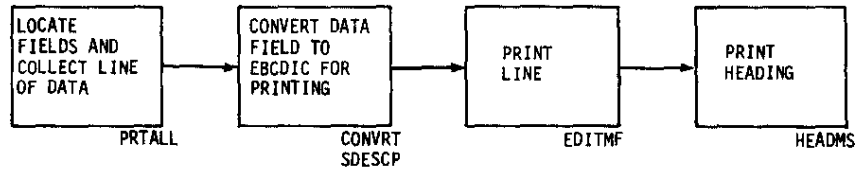


FIGURE C-8. Modifying an Existing Record

A. FUNCTION FLOW



B. HIERARCHY

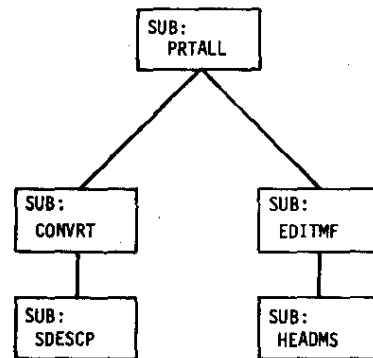


FIGURE C-9. Edit Function

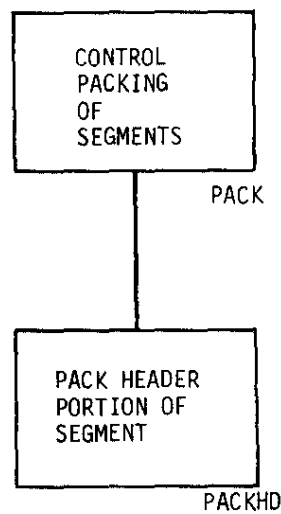
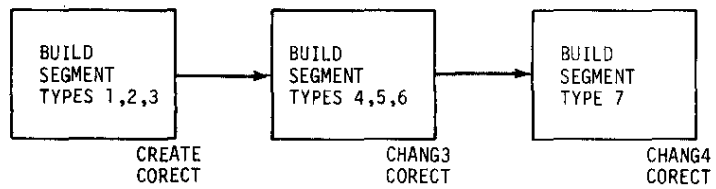


FIGURE C-10. PACK Function

A. FUNCTIONAL FLOW



B. HIERARCHY

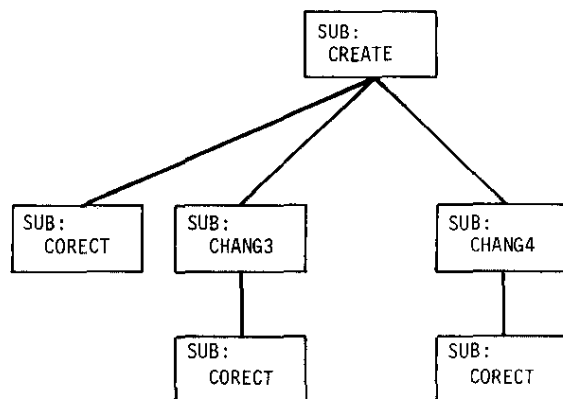


FIGURE C-11. Subroutines for Creating a New Record

APPENDIX D - QUERY FUNCTION

The overall hierarchy of the QUERY module is shown in Figure D-1. The first function initializes the COMMON areas and Process Control Tables. Next, a master record is read and unpacked. The unpacked record is then passed to the QUERY function. For those records which meet the query specifications, selected data fields may be written to JOSHUA data sets, and reports may be printed. After the entire NURE master file has been processed, a JOSHUA directory record listing the informal segment name retrieved by QUERY is written (Subroutine SUBGRP).

The COMMON areas and control tables are initialized by a group of subroutines (Figure D-2). Subroutines PUTMAP, READRC, and FIELDS, which are also used by the UPDATE module, are used by QUERY to initialize system tables (Appendix C). The selection criteria are in the form of JOSHUA records. Subroutine PUTQRY reads and stores the data defining the criteria to QUERY. PUTQRY also reads and stores report option data. Subroutine PUTEXT reads and stores the records defining the data fields to be extracted from the master record for writing to the JOSHUA output data sets.

The next function is to read and unpack a master record. Subroutines READMS and UNPACK, used in the UPDATE module, perform this function (Appendix C). Now the unpacked master record is ready to be queried. The query process (Figure D-3) involves:

- Passing the unpacked record to a subroutine (QUERY1), which queries the record according to the criteria supplied by the user.
- Extracting data from the record if the record meets query specifications (EXCTRL).
- Writing the extracted data to a JOSHUA data set to be used by analysis programs (EXCTRL).
- Printing records which meet query specifications (PRTALL, PRTFMD, PRTFM, PRTOPF).

Subroutine QUERY1 "loops through" the queries that were input to the module and compares the segment names with the selection criteria. A record is selected if it has segments whose names agree with the query specification.

If a record is selected, it is passed to the subroutine package (EXCTRL) to extract and write data to JOSHUA data sets (Figure D-4). Subroutine EXCTRL initializes the JOSHUA output array and controls the data extraction by calling subroutines

to process the various types of segments. Subroutine EXLEV1 searches the extraction specification table (built by subroutine PUTEXT) for any data fields to be extracted from Segment Type 1. The table specifies segment type, the address of the data field in the segment, and the type of the data field. For each data field to be extracted from Segment Type 1, subroutine EXLEV1 calls subroutine STOREX to store the data field in the JOSHUA output array. Subroutine EXLEV2 extracts and stores data for Segment Types 2 and 3.

If data are to be extracted from sample segments (Types 4, 5, and 6) or measurement segments (Type 7), subroutine CTRL34 is called (Figure D-5). This subroutine controls "looping through" the sample segments and their associated (son) measurement segments. For each sample segment which satisfies the query specifications, subroutine EXLEV3 is called to extract data from Segment Types 4, 5, or 6, and subroutine EXLEV4 is called to extract data from Segment Type 7.

After all the data for a sample within the record have been extracted, subroutine WRTCID is called to write the data to the JOSHUA data set. This extraction procedure continues until the entire master record has been processed. Control is then returned to subroutine QUERY1 to process the next query through the unpacked master record.

Finally, when all the queries have been processed through the master record, it may be printed if it meets at least one query. The various edit options are specified by the user as input. If none are specified, the record is not printed.

Subroutine PRTALL, also used in the UPDATE function, prints all the data in the master record (Appendix C and Figure C-9). An edit of several field measurement data in tabular form is another of the report options (Figure D-6). Subroutine PRTFMD collects the field measurement data from the record and calls subroutine SDESCP to convert table-coded values to 8-character EBCDIC fields. Subroutine FMDATA prints an edit line and maintains page and line count. Another report option is an edit of element concentration data (Figure D-7). Subroutine PRTFM collects the element concentration data and calls subroutine EDITM, which prints an edit line and maintains page and line count. Finally, a report of all site description data may be requested (Figure D-8). Subroutine PRTOFF locates each data field pertaining to site description and builds a print line. Subroutines CONVRT and SDESCP convert a value from internal representation to an 8-character EBCDIC field. Subroutine EDITMF prints the line of data passed to it, and calls subroutine HEADMS to print header lines and to maintain page and line count.

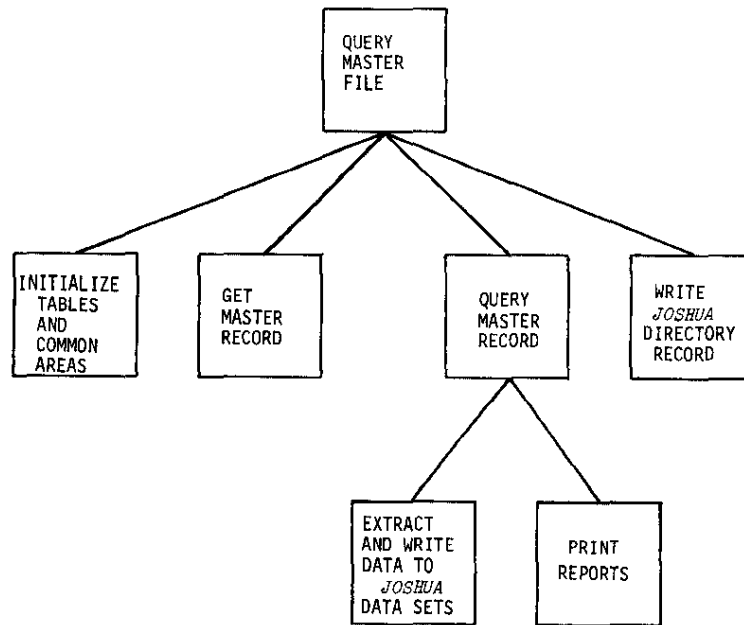
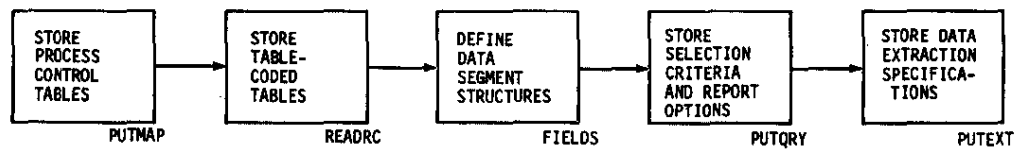


FIGURE D-1. Hierarchy of QUERY Module

A. FUNCTIONAL FLOW



B. HIERARCHY

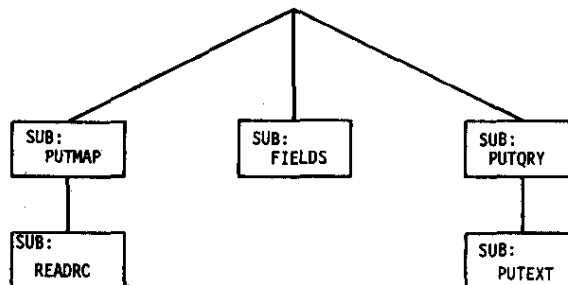
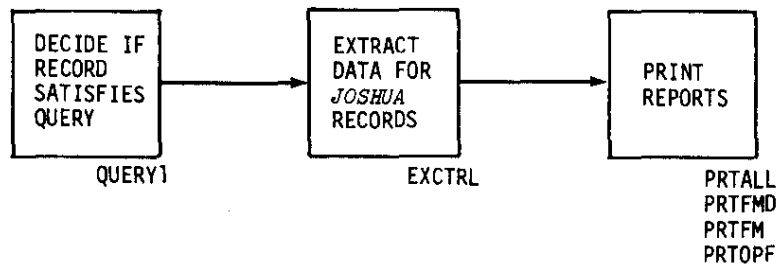


FIGURE D-2. Subroutines for Initializing System Tables and Common Areas

A. FUNCTIONAL FLOW



B. HIERARCHY

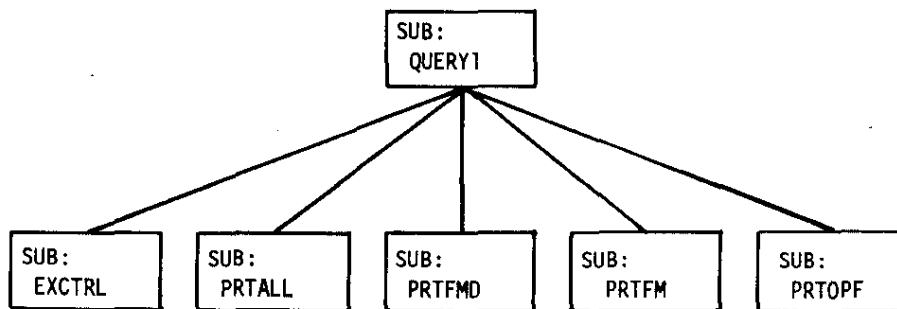


FIGURE D-3. Querying the Master Record

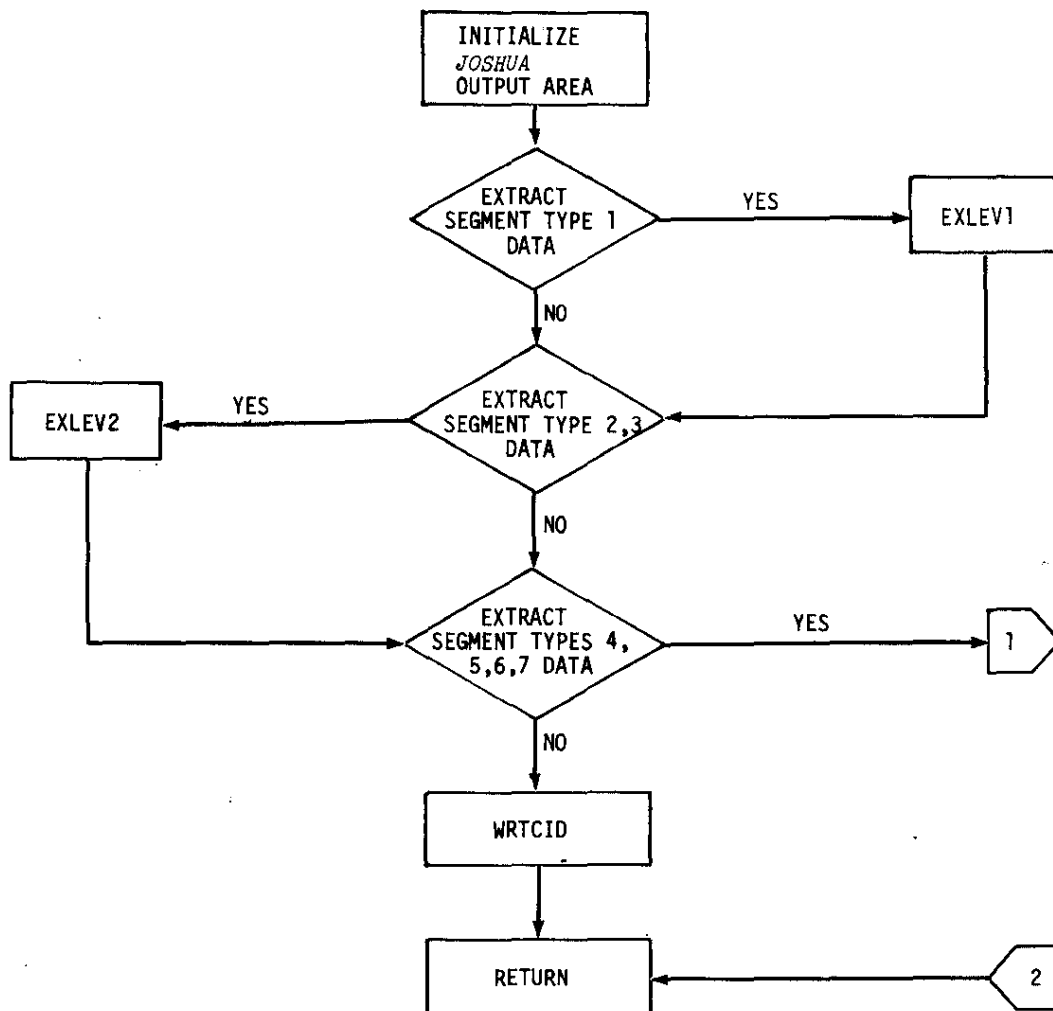


FIGURE D-4. Logic of Data Extraction Process-I

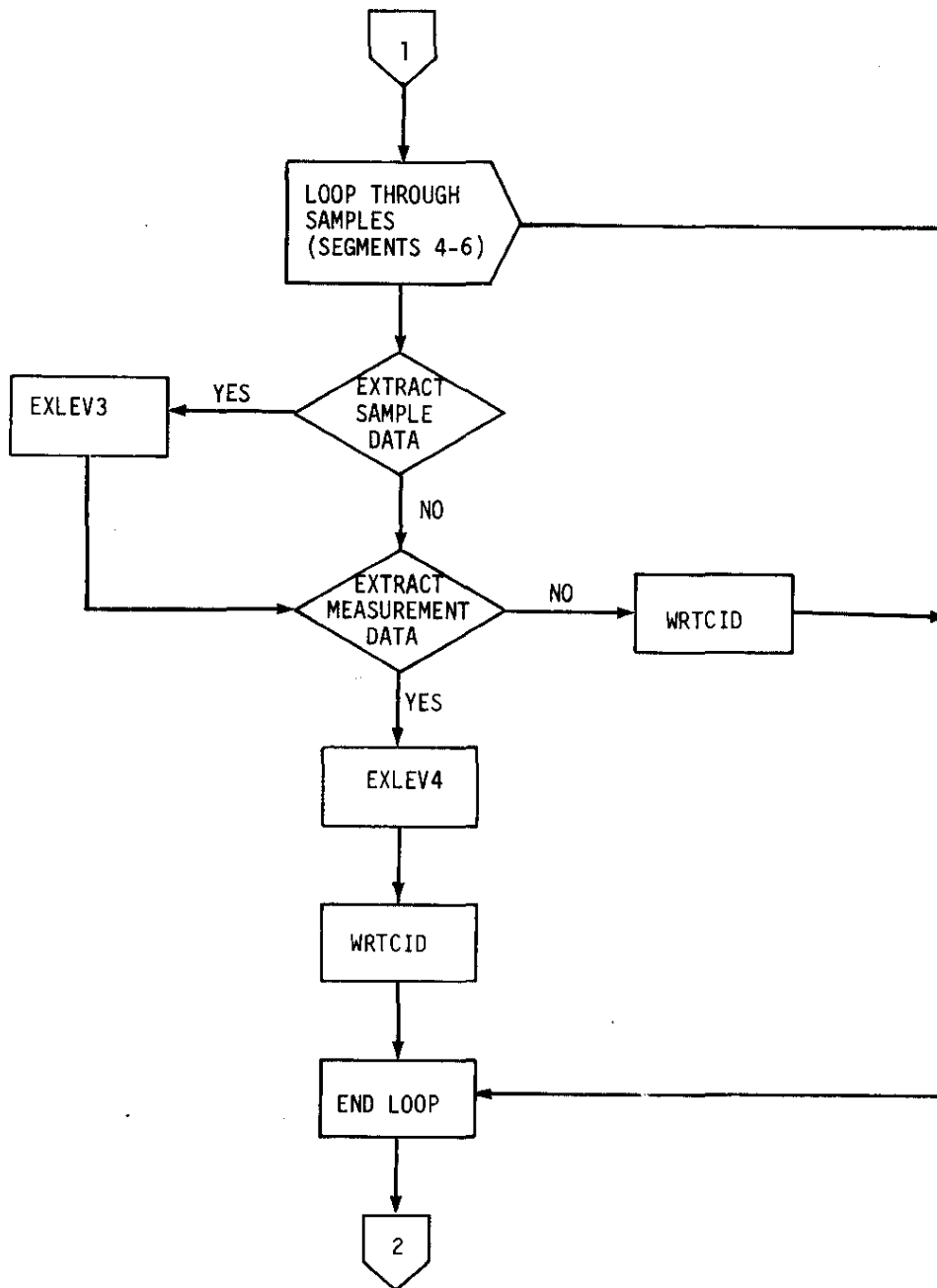
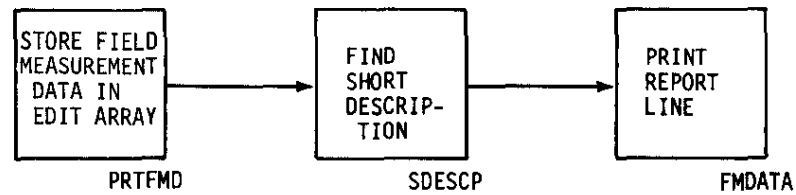


FIGURE D-5. Logic of Data Extraction Process-II

A. FUNCTIONAL FLOW



B. HIERARCHY

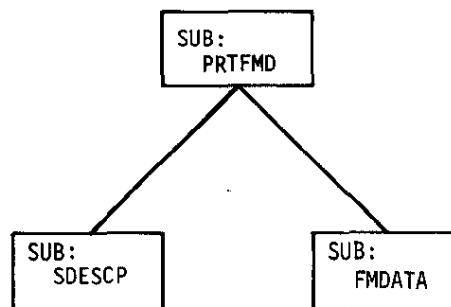
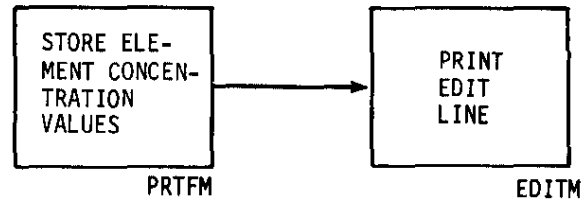


FIGURE D-6. Print Field Measurement Data

A. FUNCTIONAL FLOW



B. HIERARCHY

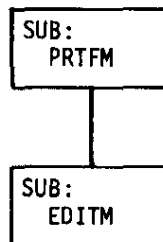
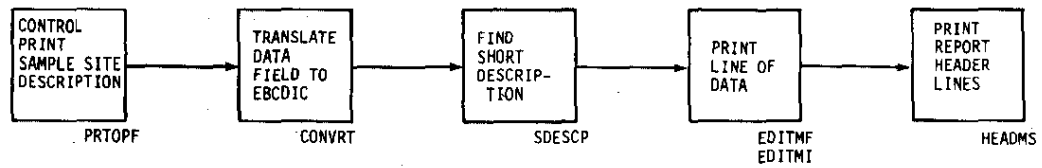


FIGURE D-7. Print Element Concentration Data

A. FUNCTIONAL FLOW



B. HIERARCHY

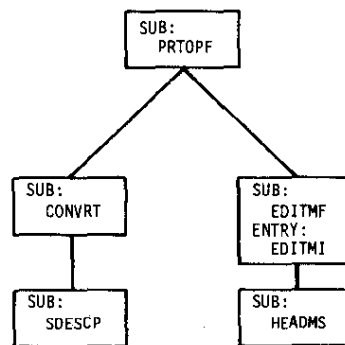


FIGURE D-8. Print Site Description Data

APPENDIX E - MODULES TO MAINTAIN SYSTEM TABLES

Support modules are available to update and edit the system tables. Each module (Figure E-1) does the same function for a specific type of table. A module reads the input, builds records, and writes them to a JOSHUA data set (Figure E-2). After the input is processed, all records are read from the data set, and tables are printed. If there is no input, the contents of the data set are printed.

The tables may be updated via a standard JOSHUA terminal. Also, the tables may be viewed at the terminal and printed using standard JOSHUA features. Normally, the support module is used to create a set of tables and to print them in a compact format. The JOSHUA features can update a field within a table and then recall selected tables for viewing.

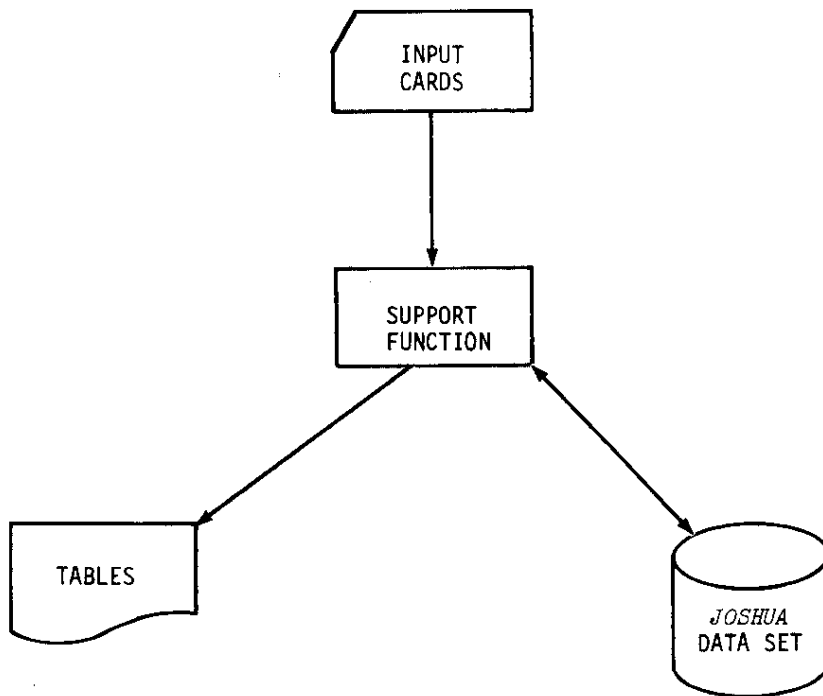


FIGURE E-1. Support Module for System Tables

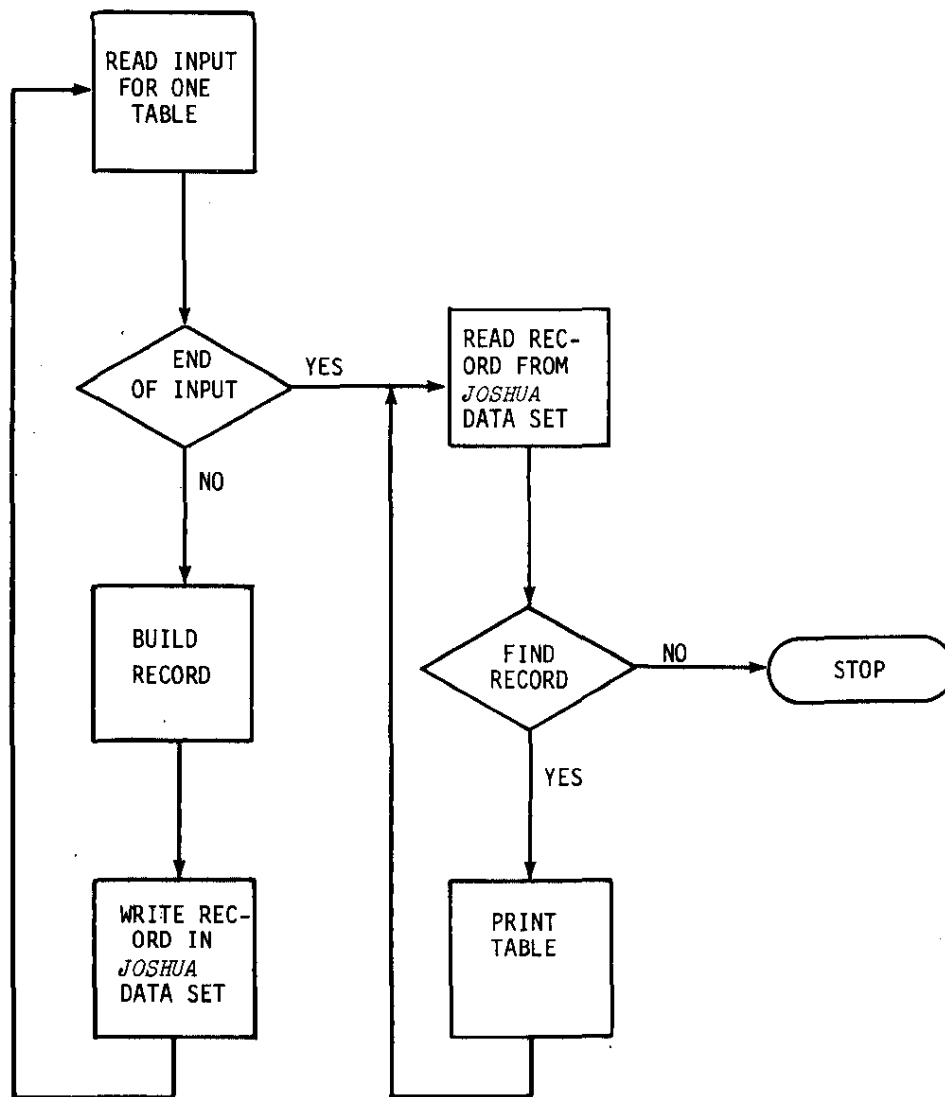


FIGURE E-2. Support Module Logic