USER DOCUMENTATION FOR NORTH DAKOTA GEOCHEMICAL DATA-BASE SOFTWARE SYSTEM--INPUT, UPDATE, AND RETRIEVAL PROCEDURES

By John W. Atwood

U.S. GEOLOGICAL SURVEY

Open-File Report 93-117

Prepared in cooperation with the

NORTH DAKOTA STATE PUBLIC SERVICE COMMISSION

Bismarck, North Dakota

1993

U.S. DEPARTMENT OF THE INTERIOR

BRUCE BABBITT, Secretary

U.S. Geological Survey

Dallas L. Peck, Director

Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

For additional information
write to:

District Chief
U.S. Geological Survey
Water Resources Division
821 East Interstate Avenue
Bismarck, ND 58501-1199

Copies of this report can be
purchased from:

U.S. Geological Survey
Books and Open-File Reports
Section
Federal Center, Box 25425
Denver, CO 80225

CONTENTS

ILLUSTRATIONS

USER DOCUMENTATION FOR NORTH DAKOTA GEOCHEMICAL DATA-BASE SOFTWARE SYSTEM--INPUT, UPDATE, AND RETRIEVAL PROCEDURES

By John W. Atwood

Abstract

A data-base software system has been developed for the management of geochemical data collected by the U.S. Geological Survey and cooperating agencies in North Dakota. This report describes the purpose, use, and design of the North Dakota geochemical data-base software system.

The software provides capabilities to input, update, and retrieve geochemical data. Data can be input to the data base either interactively or by loading in a file of specially formatted data records. Stored data can be viewed and updated interactively.

Data can be retrieved and output in the form of a table, or the data can be output to a file in a format suitable for input to other software packages. Access to the data can be restricted on a station-by-station basis to protect preliminary or proprietary data from unauthorized use.

The geochemical data-base software is designed as a single-user system. Only one user can update the data base at a time. However, if no user is updating the data base, multiple users can use the retrieval routines to retrieve and output data from the data base. The software is written in Fortran 77, but uses a number of system-dependent subroutines. Data are stored in a group of direct-access files.

1. Introduction

The U.S. Geological Survey has conducted several coal hydrology studies in cooperation with various State and Federal agencies. These studies have produced a large amount of geochemical data that could not be stored in the U.S. Geological Survey's National Water Information System (NWIS; Maddy and others, 1990) in the format desired.

A data-base software system was needed to store multiple sample data collected at the same location but at different times and depths. The sample records needed to be keyed or identified by site-identification number (site ID), date and time sampled, and depth of the sample. Both starting and ending values for date, time, and depth needed to be stored as part of the key or identifier. Although the NWIS Water-Quality System stores sample data keyed on the site ID and the date and time sampled and allows a depth to be specified for the sample, no provision is made to include the depth as part of the sample key or identifier. Neither is there a provision made for storing both starting and ending values for date, time, and depth as part of the key.

Preliminary data and data that are proprietary to the various cooperating agencies needed to be stored. These data needed to be protected against

unauthorized access. NWIS has no provision for restricting the access of individual station records.

1.1 Purpose

This report describes the North Dakota geochemical data-base software system. The system was developed by the U.S. Geological Survey as part of a cooperative project with the North Dakota State Public Service Commission. The data-base system was developed to store multiple sample data keyed on the site ID, starting date and time, ending date and time, starting depth, and ending depth. The data-base system was designed to store data from chemical analysis of overburden and spoil-pile materials and water-quality analysis of surface and ground water. Any numeric data that might be collected for a sample can be stored. The only requirements are that the data must be a single numeric value and must have a parameter code assigned to it. The data-base system also includes access controls that allow individual sites to be protected against unauthorized access or update.

1.2 Overview of the geochemical data-base software system

The geochemical data-base software system consists of the primary program called GCBASE and several utility programs. GCBASE is an interactive screen editor program that is used to input new data into the North Dakota geochemical data base and to update data that already have been entered into the data base. New data can be added to the data base by entering data on a form displayed on the user's terminal screen. Any data previously stored are displayed in the proper fields on the screen form. Previously stored data can be updated by positioning the cursor to the desired data field and making the desired updates using the program's editing functions. The cursor can be moved between the various data fields and can be placed on specific character positions within a data field. Cursor movement is accomplished through one or more of the codes described in the "Data-field editing control codes" section.

Updates are made directly to the data-base files. The data-base files are direct-access files and can be updated by only one user at a time. Multiple users may read the data if no user has the data base open for updating. An access control system has been built in that allows individual sites to be protected against unauthorized access or update.

The utility programs used in the geochemical data-base software system are GCUTIL, GCRTRV, GCINPT, AINITF, BPARAM, and BACCSS. The program GCUTIL is used to run special utilities against the data base. The program GCRTRV is used to retrieve and output selected data from the data base. The program GCINPT is used to input batch data into the data base. The program AINITF is used to initialize the data-base site and data files and is used only once. The programs BPARAM and BACCSS are used to create and update the parameter index/definition and user-access files.

1.3 Hardware and software requirements

A Prime 50 series minicomputer was used to develop the software. The software is known to execute under PRIMOS operating system versions 19, 20, 21, and 22. All of the software is written in Fortran 77 programming language (American National Standards Institute, Inc., 1978). A separate

software package called DISCUR (Atwood and Bartholoma, 1991), also written in Fortran 77, is required for cursor positioning and screen editing of the data. A few subroutines proprietary to Prime have been used for unformatted input and output and for access to the system clock. If the software is converted for use on another computer, references to these subroutines would need to be replaced by equivalent subroutines on that computer.

The GCBASE program currently accesses the NWIS site header and water-quality parameter files. These files on the Prime minicomputer are in Prime's proprietary MIDAS file format. The NWIS site header file (Mathey, 1990) is used by the GCBASE program to validate the site ID, and the water-quality parameter file is used to validate the specified parameter code. To convert the geochemical data-base software system to a non-Prime computer, the subroutines used to validate the site ID parameter codes would need to be removed or replaced with different subroutines in order to look up valid site ID's and parameters.

2. Input and update program (GCBASE)

The GCBASE program is used to input data into the data base and update data that already have been entered into the data base. If the program has been installed as a system command, GCBASE is invoked by typing

```
GCBASE
```

If the program has not been installed as a system command, GCBASE is invoked by typing

```
R GCBASE3>RUN>GCBASE
```

After GCBASE has been invoked, all input and display of the data is handled by screen subroutines. The program uses a number of pages or screens, each of which contains related information, to input, display, and edit the data. Data are entered into fields on the screen and can be edited using field-editing codes or commands. Descriptions of the field-editing codes or commands are given in the "Data-field editing control codes" section.

2.1 Site identification selection

The first screen contains prompts for the agency code and site ID for the site to be modified or added. The site ID is expected to contain either an 8-digit ID number (surface water only) or a 15-digit latitude-longitude-sequence number. No additional checking is done on an 8-digit ID number; a 15-digit number is checked to assure that the latitude and longitude values are within the appropriate geographic range, and a warning is given if an invalid number is found. After the agency code and site ID have been checked, the user is prompted as to whether the input values are correct. If they are not correct, the user can reenter the correct values after answering "N" to the prompt. If the prompt is answered with a "Y", a search is made in the NWIS site header file for the specified site ID, and control is passed to the next screen. If the specified site ID is not found in the NWIS site header file, an error message is given, and the user is given the opportunity to select another site. If the specified site ID is found, then the program proceeds to the sample header screen.

2.2 Data-entry screens

The program uses two data-entry screens. The first is the sample header screen, and the second is the sample data screen.

2.2.1 Sample header screen

The sample header screen consists of the dates, times, and depths that uniquely identify each sample. A new sample is identified or an existing sample is selected using this screen. The sample header screen is shown in figure 1.

2.2.2 Sample record screen

The sample record screen consists of the sample parameter data. The sample parameter codes and their respective values are input using this screen. A maximum of 450 parameters per sample is allowed. The sample record screen is shown in figure 2.

```
                          ***** Sample Header *****


Agency Code USGS                                        Site ID 463711103154201

Date/Start Date 1=                                      Time/Start Time 2=
       End Date 3=                                             End Time 4=

Top Depth 5=                                        Depth/Bottom Depth 6=

Medium Code 7=









Optns  S.Fld  N.Fld  L.Fld  Repnt  In:Ed  Abort  End  Blank  Ins.C  Del.C  No.Ctrl  Ys.Ctrl
  ?      $      >      <      %      &      !     *     #      _      \       "        ^
```

Figure 1.--Sample header screen.

2.3 Required fields

Each screen has certain fields that are required. The data items that are used by the data-base software system to identify the record (primary key) always are required. Other important data items also are required. For example, a value must accompany any parameter that is specified. Data must be input for all required fields before the program will allow the data to be stored.

2.4 Error checking

The program performs limited error checking on the data as they are input. For some fields, such as the numeric fields, nonvalid characters are rejected immediately upon input. Other data are checked for valid choices where a list of valid codes is available. Dates are checked to see that they are not for nonexistent days or for dates after the current date. Latitude and longitude values also are checked for validity. Parameter codes are checked to see that they have been defined. Where a list of valid codes is available, the codes

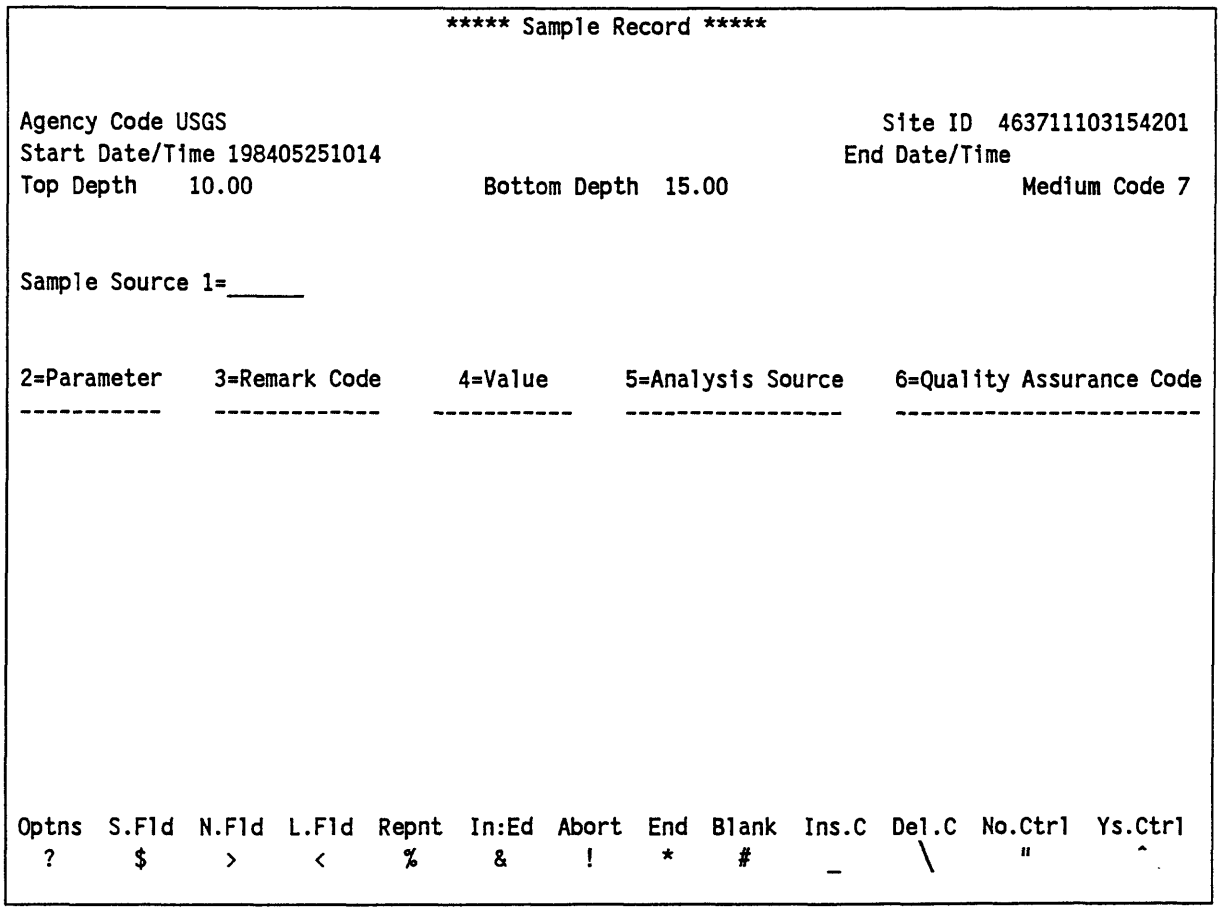

Figure 2.--Sample record screen.

can be listed by inputting a "?" in column one of the field or by entering a "?" while in the temporary control mode (preceded by a "^"). If an error is detected, a message indicating that an error has been made is displayed at the bottom of the screen.

2.5 Site-specific access

A routine to restrict access to data on a site-by-site basis is incorporated in the North Dakota geochemical data-base software system. There are 32 possible special access groups available. A group may represent the agency that owns the data or the project for which the data were collected. Access to each site may be restricted to any 1 or to many of the 32 possible groups. Within each group, access may be set to read, update, delete, and protect access rights. Read access allows a user to retrieve the data for a site, but no modifications are permitted. Update access allows a user to modify analysis data for a site, but the sample header cannot be modified. Delete access allows the user to delete an entire sample for the site. Protect access rights allow the user to modify the access rights to a site. Access rights to a site are modified using the GCUTIL program. If a user has no access to a site, no retrieval or modification of a site's data is permitted.

2.6 Default data

Default answers to prompts are provided in some data fields. These default answers are displayed in the data field and may be used, without modifying the data field, by inputting a carriage return. If the default answer is not what is desired, the default answer can be replaced with the desired data.

2.7 Data-field justification

Character fields are left justified, which means that the characters are shifted to the left until the leftmost nonblank character is in the leftmost field character position. Number fields are right justified, which means that the characters are shifted to the right until the rightmost nonblank character is in the rightmost field character position.

3. Data-field editing control codes

While the GCBASE program is executing, two lines that contain some of the data-field editing control codes are maintained at the bottom of the screen. These same two lines are used to display error messages, warnings, and special prompts. If an error is detected, an error message is displayed on the bottom line, and a continuation prompt is displayed on the line above the error message. Special messages and prompts are displayed in the same manner. When the special message or prompt is no longer needed, the data-field editing control codes are displayed again. The data-field editing control codes control the positioning of the cursor and string pointer. The cursor is positioned on the character that is to be changed on the terminal screen. The string pointer is positioned on the character that is to be changed in the stored character string.

The special data-field editing control codes are as follow:

A BACKSPACE leaves the character at the current position unchanged and moves the cursor and string pointer one character back.

A BREAK stops execution of the program and returns to the operating system.

A DELETE blanks the entire string and sets the cursor and string pointer to character position one in the field.

A RETURN enters the string as it currently appears. Control and the cursor are then passed to the next field.

A SPACE leaves the character at the current position unchanged and moves the cursor and string pointer one character forward.

A TAB enters the string as it currently appears. Control and the cursor are then passed to the next subfield.

An * terminates the string with the previous character and deletes itself and all characters in the field that follow. Control and the cursor are then passed to the next field.

A ^ puts the program in temporary control mode. This allows one of the characters that is executed as control in column one, or one of the other special control characters, to be executed as control anywhere in the field. The cursor and string pointer are left at their current positions, and the next character causes the special function to be performed. If a character that has no special function is entered as the character following the ^, then it is ignored. The program returns to normal mode after the control character is executed. No change is made to the data field at the cursor position.

A " puts the program in temporary data mode. This allows the next single character to be accepted as data, even if it is one of the special control codes. The cursor and string pointer are left at their current positions, and the next character input is accepted as data at that position if it is a legal character. After the data character has been input, the program proceeds in the normal mode.

A \ deletes the character at the current character position, then shifts the characters that follow to the left. The cursor and string pointer are left at their current positions.

A # replaces the character at the current position with a blank if a blank is permitted at that location.

An _ inserts a blank at the current character position, then shifts the characters that follow to the right. The cursor and string pointer are left at their current positions.

Any legal noncontrol character will replace the character at the current character position within the field.

If either end of the field is passed, the string is entered as it currently appears. Control is then passed to the next or to the previous field, depending on which end of the field was passed.

The following characters are executed as control when occurring in column one, or when the program has been placed in temporary control mode (following a ^):

An & switches screen modes between input and edit. If the program is in input mode, and if all of the required fields contain valid data, the program enters edit mode, which allows the selective editing of screen data. If the program is in input mode, and one or more of the required fields does not contain valid data, the cursor is positioned to the next required field and remains in input mode. If the program is in edit mode, the program reenters the input mode at the current field.

A $ starts the current string over again. The field is reinitialized to its former content.

An ! exits the current program level, aborting any changes that have been made in the current screen.

A > enters the string as it currently appears. Control is then passed to the next field.

A < enters the string as it currently appears. Control is then passed to the previous field.

A % repaints the entire screen. The % is used to repair screens that have somehow been overwritten with system messages or otherwise garbled.

A ? produces a list of the valid options if there are any. These are listed near the bottom of the screen.

The following characters position the cursor to the specified location within the field when used in the temporary control mode (following a ^):

An A positions the cursor one character past the last nonblank character in the field. If the position would be past the end of the field, then the cursor is placed on the last character in the field. (Append)

A B positions the cursor on the first character in the field. (Beginning)

An E positions the cursor on the last character in the field. (End)

An F positions the cursor on the first nonblank character in the field. (First)

An L positions the cursor on the last nonblank character in the field. (Last)

A P positions the cursor one character before the first nonblank character in the field. If the position would be before the beginning of the field, then the cursor is placed on the first character in the field. (Prefix)

Many of Prime's EMACS commands (Buck, 1982) also are accepted as control. In the list that follows a ^ means that the control key is pressed simultaneously with the specified character. <ESC> means that the ESC or escape key is pressed before the specified character or characters in a sequence. Many of the functions are duplicated because each EMACS command was mapped to the closest equivalent in the GCBASE program. Where applicable, the non-EMACS equivalent code has been given in parentheses following the command description. These commands are as follow:

^A positions the cursor on the first character in the field. (B)

^B positions the cursor backward one character. (<BACKSPACE>)

^C reexecutes the most recent command.

^D deletes the character after the cursor.

^E positions the cursor on the last character in the field. (E)

^F positions the cursor forward one character. ( )

^G cancels a multiple character command that has been started.

^H (also the backspace key) does not delete a character as it does in EMACS, but instead positions the cursor backward one character. (<BACKSPACE>)

^I (also the tab key) positions the cursor to the next subfield. (<TAB>)

^K kills or blanks the field. (<DELETE>)

^L repaints the screen. (%)

^M (also the return key) positions the cursor to the next field. (<RETURN>)

^N positions the cursor to the next field. (>)

^O opens the line and inserts a blank at the current character position. (_)

^P stops execution of the program and returns to the operating system. (<BREAK>)

^Q inserts the next character as data (X-ON on most terminals and should not be used). (")

^R positions the cursor to the previous field. (<)

^S positions the cursor to the next field (X-OFF on most terminals and should not be used). (>)

^V switches from input mode to edit mode or from edit mode to input mode. (&)

^W kills or blanks the field after moving the field contents to the kill buffer.

^X is used as a prefix to other commands.

^Y replaces the field with the contents of the kill buffer.

^Z positions the cursor to the previous field. (<)

^_ lists any valid values for field. (?) (ignored on most terminals)

<DEL> does not delete a character as it does in EMACS, but instead kills or blanks the field. (<DELETE>)

^X^C aborts the screen. (!)

^X^G ignores the ^X prefix.

^X^H kills or blanks the field left of the cursor.

^X^I inserts a blank at the current character position. (_)

^X^K kills or blanks the field left of the cursor.

^X^L converts entire field to lowercase characters.

^X^S switches from input mode to edit mode or from edit mode to input mode (X-OFF on most terminals and should not be used). (&)

^X^U converts entire field to uppercase characters.

^X^W switches from input mode to edit mode or from edit mode to input mode. (&)

^X^Z is used as a prefix to other commands.

^XQ puts the program in temporary data mode. (")

^XR repaints the screen. (%)

^XS switches from input mode to edit mode or from edit mode to input mode. (&)

^XV positions the cursor to the previous field. (<)

^X[positions the cursor to the previous field. (<)

^X] positions the cursor to the next field. (>)

^X^Z^A positions the cursor to the previous field. (<)

^X^Z^E positions the cursor to the next field. (>)

^X^Z^G ignores the ^X^Z prefix.

^X^Z^H kills or blanks the field left of the cursor.

^X^Z^K kills or blanks the field right of the cursor. The character in the position of the cursor also is deleted.

^X^Z^Y replaces the field with the contents of the kill buffer.

<ESC>^D kills or blanks the field right of the cursor. The character in the position of the cursor also is deleted.

<ESC>^G ignores the <ESC> prefix.

<ESC>^O opens the line and inserts a blank. (_)

<ESC>^V positions the cursor to the next field. (>)

<ESC>^Y replaces the field with the contents of the kill buffer.

<ESC><SP> (escape followed by space) deletes a character at the current character position. (\)

<ESC>< positions the cursor to the previous field. (<)

<ESC>> switches from input mode to edit mode or from edit mode to input mode. (&)

<ESC>? lists any valid values for field. (?)

<ESC>@ terminates the field at the cursor position. (*)

<ESC>A positions the cursor to the previous field. (<)

<ESC>B positions the cursor to the previous subfield.

<ESC>D kills or blanks the field. (<DELETE>)

<ESC>E positions the cursor to the next field. (>)

<ESC>F positions the cursor to the next subfield. (<TAB>)

<ESC>G switches from input mode to edit mode or from edit mode to input mode. (&)

<ESC>K kills or blanks the field right of the cursor. The character in the position of the cursor also is deleted.

<ESC>L converts entire field to lowercase characters.

<ESC>M positions the cursor to the first nonblank character. (F)

<ESC>N positions the cursor to the next field. (>)

<ESC>P positions the cursor to the previous field. (<)

<ESC>R positions the cursor to the previous field. (<)

<ESC>S positions the cursor to the next field. (>)

<ESC>U converts entire field to uppercase characters.

<ESC>V positions the cursor to the previous field. (<)

<ESC>W copies field into kill buffer.

<ESC>Y replaces the field with the contents of the kill buffer.

<ESC>~ reinitializes the field to its former content. ($)

<ESC><DEL> kills or blanks the field. (<DELETE>)

4. Special utilities program (GCUTIL)

The GCUTIL program is used to run special utilities against the data base. If the program has been installed as a system command, GCUTIL is invoked by typing

```
GCUTIL
```

If the program has not been installed as a system command, GCUTIL is invoked by typing

```
R GCBASE3>RUN>GCUTIL
```

The program options available are:

1. Exit GCUTIL Program
2. Modify a Site's Access

4.1 Exit GCUTIL program

The Exit GCUTIL Program option closes the opened files and exits the GCUTIL program.

4.2 Modify a site's access

The Modify a Site's Access option is used to modify the user access groups for a site. Only a user who has protection rights to a site can change the access groups. To modify the access groups for a site, select option 2 of GCUTIL. Select the site to be modified when prompted by the program and then specify the access groups for each of the READ, UPDATE, DELETE, and PROTECT attributes.

5. Retrieve and output selected data (GCRTRV)

The GCRTRV program is used to retrieve and output selected data from the data base. If the program has been installed as a system command, GCRTRV is invoked by typing

```
GCRTRV
```

If the program has not been installed as a system command, GCRTRV is invoked by typing

```
R GCBASE3>RUN>GCRTRV
```

The program options available are:

1. Exit GCRTRV Program
2. Retrieve By Station No.
3. Create Table Listing
4. Create P-STAT Data File
5. Create Generic Transfer File
6. Dump Sample Headers

5.1 Exit GCRTRV program

The Exit GCRTRV Program option closes the opened files and exits the GCRTRV program.

5.2 Retrieve by station no.

The Retrieve by Station No. option is used to retrieve records from the data base using a list of site IDs. The list can be provided interactively, or it can be provided in the form of a flat file containing the agency code in columns 1-5 and the site ID in columns 7-21. The NWIS retrieval routines can be used to create the flat-file site list, or the list can be created using any text editor. The beginning and ending dates for samples to be included in the retrieval also can be specified.

5.3 Create table listing

The Create Table Listing option is used to produce a table of the retrieved data. Two table types are provided. One type is a listing of all of the data for each selected record. The other type allows the output of as many as 169 specified parameters.

5.4 Create P-STAT data file

The Create P-STAT Data File option is used to produce an output file for input to/P-STAT. P-STAT (P-STAT, Inc., 1986) is a statistical package used by the U.S. Geological Survey on the Prime minicomputer. At the beginning of the output file, a header block that is used by P-STAT to identify the columns of parameters is included. Two output formats are provided. One format includes the sample header information and the other does not. The selected parameters are printed six to one line. A maximum of 169 parameters can be output.

5.5 Create generic transfer file

The Create Generic Transfer File option is used to create a file suitable for transfer of data from one data base to another. Two file types are provided. One type contains all of the data for each selected record. The other type allows the output of as many as 169 specified parameters. This option is similar in content and format to the create table listing option. The exceptions are: (1) No page headers are produced, a special one-line header that is used to identify the file is produced; (2) the file width is limited to 80 characters per line; (3) each sample header is preceded on the line with a "0", and each sample data line is preceded on the line with a "1".

5.6 Dump sample headers

The Dump Sample Headers option is used to output the raw header of each sample selected. All sample information except the parameter codes and values are listed. The option selects all samples between the specified dates and prints them in an unformatted list. Each header is output as a single text string exactly as stored in the file. Caution should be used with this option because it can take a very long time and can produce a very large output file.

6. Input batch data (GCINPT)

The GCINPT program is used to input batch data (such as NWIS) into the data base using an ASCII sequential access file for batch input. If the program has been installed as a system command, GCINPT is invoked by typing

```
GCINPT
```

If the program has not been installed as a system command, GCINPT is invoked by typing

```
R GCBASE3>RUN>GCINPT
```

The program options available are:

1. GCRTRV P-STAT Input File (With Header)
2. QWDATA (NWIS) P-STAT Input File
3. Generic Transfer File

6.1 GCRTRV P-STAT Input File (With Header)

The GCRTRV P-STAT Input File (With Header) option is used as it is output from the GCRTRV program.

6.2 QWDATA (NWIS) P-STAT Input File

The QWDATA (NWIS) P-STAT Input File option reads a file that is created by concatenating the two files output by NWIS. The NWIS P-STAT command file is placed in front of the NWIS P-STAT data file in the new GCINPT input file. No further changes should be made to the concatenated file.

6.3 Generic Transfer File

The Generic Transfer File option is a file produced by the GCRTRV program and can be used to transfer data from one data base to another. Batch data from other sources can be placed in a generic transfer file for inclusion into the data base.

An output file called T$GCINPT.ERR is created during processing. This file contains the data that were not processed, for any reason, during the transfer. The file is written in the generic transfer format (section 10.3) and can be edited and/or saved for later processing. However, the name of the T$GCINPT.ERR file must be changed or the file must be deleted before another batch file can be processed.

7. Create subdirectory structure and initialize the data-base site and data files (AINITF)

CREATE_GCBASE creates the data-file subdirectory structure and runs the AINITF program. CREATE_GCBASE is invoked by attaching to the directory GCBASE3>RUN, and typing

R CREATE_GCBASE.CPL

The AINITF program is used to initialize the data-base site and data files. It is used only once during the initial installation of the data-base system. The AINITF program should never be run outside of the CREATE_GCBASE shell. After initialization of the data base, the CREATE_GCBASE.CPL and AINITF.RUN files may be deleted from the directory GCBASE3>RUN if desired. No options or other input are required.

8. Create and update parameter index/definition file (BPARAM)

The BPARAM program is used to create and update the parameter index/definition file. BPARAM is invoked by attaching to the directory GCBASE3>RUN and typing

R BPARAM.RUN

Every parameter that is not defined in the NWIS system and is used with GCBASE, must be defined in the GC.PARAMS file. At least one parameter must be defined in the file in order for the GCBASE program to run. The GC.PARAMS file is located in the RUN directory and is a direct access file. All updates to the file must be made through the BPARAM program. The BPARAM program prompts are for a five-digit parameter code number, the data type (C=character, N=numeric), the number of digits left and right of the decimal (if it's numeric), the parameter field length, and a descriptive title. The current value of a parameter definition can be seen by inputting an empty <carriage return> when prompted for the data type (character or numeric). Updates are made by specifying the parameter code to be updated and inputting the new values. A parameter can be deleted by inputting a "D" when prompted for the data type (character or numeric). To exit the program, enter an empty <carriage return> when prompted for the parameter code.

9. Create and update user access file (BACCSS)

The BACCSS program is used to create and update the user access file. BACCSS is invoked by attaching to the directory GCBASE3>RUN and typing

```
R BACCSS.RUN
```

Each user of the geochemical data-base system must be defined in the user access file. The GC.ACCESS file is located in the RUN directory and is a direct access file. All updates to the file must be made through the BACCSS program. The program prompt is for the user ID of the individual to be added or modified and the access groups to which the user belongs. If the specified user does not exist in the file, the program prompts if the user is to be added. If the specified user does exist in the file, the program prompts if the user is to be deleted. Place a "1" under the digit representing each special access group to which the user belongs, then press the carriage return. The current access allowed to a specific user is displayed above the input field. The entire list of access groups must be specified each time the field is updated. To leave the current access groups unchanged, answer the access group prompt with a "Q" or an "N". To exit the program, enter an empty <carriage return> when prompted for the user ID.

The names of the access groups are hard coded in the BACCSS program. To add or modify the names of the access groups, the source file must be modified as described in the "Program installation" section, step E. Then the program must be rebuilt using the commands

```
A GCBASE3
R BLD.BACCSS.CPL
```

10. Geochemical data-base software system design

The geochemical data-base software system consists of several groups of files. Two auxiliary files are contained in the RUN subdirectory in the GCBASE3 directory. Most of the files are contained in a subdirectory called DATA, also located in the GCBASE3 directory. Also, access is made to the site and parameter code files in the NWIS package. All of the files are direct-access files with the exception of the NWIS files. Only the GCBASE system software should be used to access the files. Never should a text editor be used to access the files. The directory structure of the GCBASE system is shown in figure 3.

10.1 Auxiliary files

The auxiliary files GC.ACCESS and GC.PARAMS are located in the RUN subdirectory. These files are accessed by all of the geochemical data-base programs.

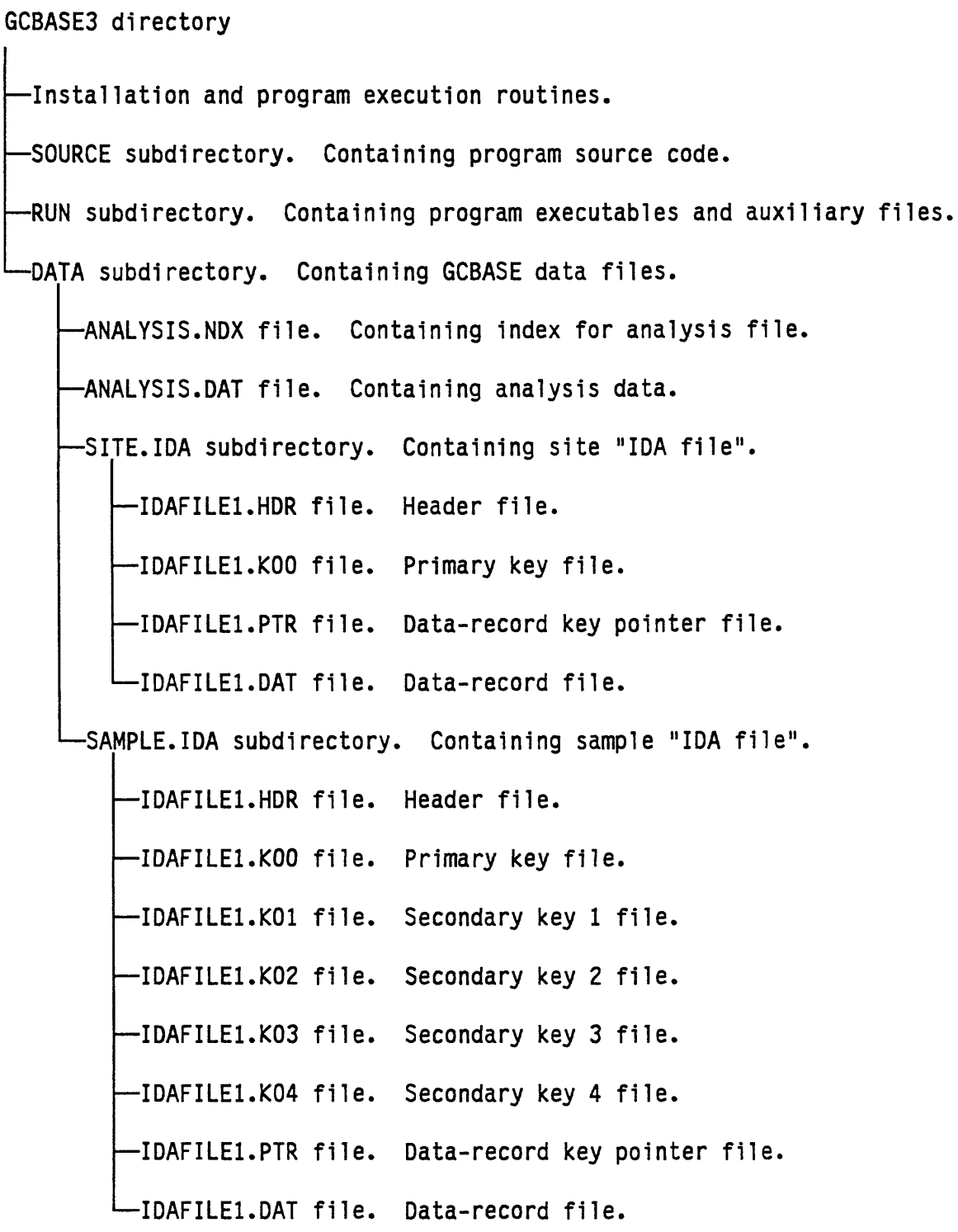

Figure 3.--Directory structure of the GCBASE system.

The GC.ACCESS file contains a list of users who are permitted to use the data-base system along with the access groups to which they belong. A description of the GC.ACCESS file follows:

File records - record length 36 bytes.

[File header record]

Bytes
1-6 = number of records in character format
7-36 = blank

[Data record]

1-32 = ID of a user authorized to use the data-base system
33-36 = 32 1-bit flags corresponding to each of the 32 access groups.

A "0" means that the user is not a member of the group; a "1" means that the user is a member of the group. The file is sorted on the user ID field.

The GC.PARAMS file contains a list of parameter codes that are not contained in the NWIS data base, but are proper codes for use in this data-base system. A description of the GC.PARAMS file follows:

File records - record length 80 bytes.

[File header record]

Bytes
1-6 = number of records in character format
7-80 = blank

[Data record]

Bytes
1-5 = parameter code
6-20 = four numbers in character format padded with blanks
Bytes
7 = field type and is represented by a "1" if the data type is character and a "2" if the data type is numeric
8 = blank
9-11 = maximum number of digits before and after the decimal point if the data type is numeric
12 = blank
13-15 = maximum number of digits before and after the decimal point if the data type is numeric
16 = blank
17-19 = length of the field
20 = blank
21-80 = short description of the parameter.

The file is sorted on the parameter code.

10.2 Data files

The geochemical data-base files are located in the DATA subdirectory. The data files consist of an index file with an accompanying analysis file and two subdirectories containing the site and sample data.

The ANALYSIS.NDX file is an index file containing internal pointers to the data file. The ANALYSIS.DAT file contains the analysis data. A description of the ANALYSIS.DAT file follows:

File records - record length 256 bytes.

Bytes	
1-16	= four internal pointers
17-256	= analysis data.

Positioning within the 240-byte data area is dependent on the data parameters present.

The site and sample data directories contain specially formatted and indexed direct files. Within the software package, the structures of the site and sample directories are referred to as Indexed Direct Access (IDA) files. Each IDA "file" consists of a subdirectory structure that contains a header file, several key index files, a data-record key pointer file, and a data-record file. The data base is designed to allow one reader/writer or many readers.

Key values are not deleted from an key index file. Data records may be deleted, but the key values in the key index files are maintained in the index. If a large number of deletions have been made, it may be desirable to pack the data base by dumping the entire data base to a generic transfer file, deleting the old data, and then loading the data back into the data base using GCINPT.

The maximum number of secondary keys allowed in an IDA file is 30. The maximum key length is 224, and the maximum data-record length is 2,048. A description of each of the individual IDA files follows:

"IDA" header file - record length 4 bytes.

Data-record file record length	(integer*4)
Number of data records in the file	(integer*4)
Number of secondary keys (0-30)	(integer*4)
Primary key starting position	(integer*4)
0 - 30 secondary key starting positions	(integer*4)
Primary key length	(integer*4)
0 - 30 secondary key lengths	(integer*4)
Number of primary key index records	(integer*4)
0 - 30 number of secondary key index records	(integer*4)
Primary key index root record	(integer*4)
0 - 30 secondary key index root records	(integer*4)
Primary key index first record	(integer*4)
0 - 30 secondary key index first records	(integer*4)
Primary key index last record	(integer*4)

```
0 - 30 secondary key index last records                    (integer*4)
Number of deleted data records                             (integer*4)
0 - nn deleted data-record pointers                        (integer*4)

"IDA" key index file - record length 32 bytes plus the number of bytes
                       (n) contained in the key.

[Subrecords 1 . . .] - key index record


Record number of parent record                             (integer*4)
Record number of left child                                (integer*4)
Record number of right child                               (integer*4)
Record number of previous key                              (integer*4)
Record number of next key                                  (integer*4)
Number of data records using key                           (integer*4)
First data record using key                                (integer*4)
Last data record using key                                 (integer*4)
Key value                                                  (character*n)

"IDA" data-record pointer file - record length 16 bytes plus 12 bytes for
                                 each secondary key.

[Subrecords 1 . . .] - data-record key pointers

Record status (0=active, -1=deleted)                       (integer*4)
Primary key index number                                   (integer*4)
0 - 30 secondary key index numbers                         (integer*4)
Primary key previous data-record number                    (integer*4)
0 - 30 secondary key previous data-record numbers          (integer*4)
Primary key next data-record number                        (integer*4)
0 - 30 secondary key next data-record numbers              (integer*4)

"IDA" data-record file - record length determined by amount of data to be
                         stored.

[Subrecords 1 . . .] - data record

Data                                                       (character*n)
```

The SITE.IDA file contains a record for each site that has data in the geochemical data-base system. The access groups for a site are contained in this file. A description of the SITE.IDA file follows:

Data-record file - record length 36 bytes.

Bytes
1-20 = site ID
21-36 = four sets of 32 1-bit flags corresponding to each of the 32 access groups.

A "0" means that the group does not have the specified access to the site, and a "1" means that the group has the specified access to the site. The four sets of access flags correspond to "read", "update", "delete", and "protect"

and allow specific groups of users to have selected rights to the data. The primary key consists of the site ID. There are no secondary keys to this file.

The SAMPLE.IDA file contains the information on each sample. Pointers within this file point to the analyses located in the analysis file. A description of the SAMPLE.IDA file follows:

Data-file records - record length 256 bytes.

Bytes
1-20 = site ID
21-32 = starting date and time
33-44 = ending date and time
45-60 = starting and ending depth
61 = medium code
62-72 = record ID, sample source, and number of records
73-256 = number of analyses and as many as 45 pointers to the analysis file.

The primary key consists of bytes 1-61 and includes the site identifier, starting date and time, ending date and time, starting and ending depth, and medium code. There are four secondary keys; secondary key 1 consists of the site ID, secondary key 2 consists of the owner agency, secondary key 3 consists of the site number, and secondary key 4 consists of the sample source.

10.3 Generic transfer file

The generic transfer file provides a vehicle to transfer data from one system or data base to another. The file is a card image file of the following format:

Generic transfer file - record length 80 bytes.

[File header record]

Columns
1 - 34 = "* GEOCHEMICAL DATA TRANSFER FILE *"
35 - 80 = blank

[Sample identification record]

Columns
1 - 1 = "0"
2 - 6 = agency code
7 - 21 = site identifier
22 - 22 = blank
23 - 30 = starting date in YYYYMMDD format
31 - 31 = blank
32 - 35 = starting time in HHMM format
36 - 36 = blank
37 - 44 = ending date in YYYYMMDD format
45 - 45 = blank

```
46 - 49 = ending time in HHMM format
50 - 50 = blank
51 - 58 = starting depth
59 - 59 = blank
60 - 67 = ending depth
68 - 68 = blank
69 - 69 = medium code ("-" = not reported)
70 - 70 = blank
71 - 75 = sample source
76 - 76 = blank
77 - 80 = number of parameters reported

[Sample parameter record]

Columns
 1 -  1 = "1"
 2 -  4 = blank
 5 -  9 = parameter code
10 - 11 = blank
12 - 12 = remark code ("-" = not reported)
13 - 14 = blank
15 - 25 = parameter value
26 - 27 = blank
28 - 32 = analysis source ("  -  " = not reported)
33 - 34 = blank
35 - 36 = quality assurance code ("A " = not reported)
37 - 40 = blank
41 - 45 = parameter code
46 - 47 = blank
48 - 48 = remark code ("-" = not reported)
49 - 50 = blank
51 - 61 = parameter value
62 - 63 = blank
64 - 68 = analysis source ("  -  " = not reported)
69 - 70 = blank
71 - 72 = quality assurance code ("A " = not reported)
73 - 80 = blank

[Trailer record]

   Columns
 1 -  1 = "$"
 2 - 80 = blank
```

10.4 National Water Information System files (NWIS)

The NWIS site and parameter files are located in the NWIS data directories. The files are in Prime's proprietary MIDAS format and are accessed using standard NWIS file-access routines. The NWIS files are used for reference only, and no provision is made in the geochemical data-base software system to modify the NWIS files. If the data-base system is converted to a non-Prime minicomputer, or installed on a Prime minicomputer that does not have the NWIS software, then the links to NWIS must be removed. If the links to NWIS are

removed, no validation of the site will be available. Removal of the links will not affect the overall purpose of the data-base system.

11. Software system installation

The instructions for installing the software on a Prime minicomputer containing the U.S. Geological Survey's NWIS and DISCUR software packages are listed below. Steps A-J must be performed in the order listed, and each step must be completed before the next step is started.

(A) Install the input and update program. This is done by issuing the following commands:

(Note: The NWIS and DISCUR software packages must have been installed previously.)

```
A GCBASE3
R INS.GCBASE.CPL          -or-          PH INS.GCBASE.CPL
```

The log file INS.GCBASE$COMO is created during the installation process. This file can be looked at to make sure the installation was successful. During the installation process, the program is compiled, loaded, and copied to the directory GCBASE3>RUN. After loading, the binary and other temporary files are automatically deleted, and the source directory is cleaned up.

(B) Install the special utilities program. This is done by issuing the following commands:

```
A GCBASE3
R INS.GCUTIL.CPL          -or-          PH INS.GCUTIL.CPL
```

The log file INS.GCUTIL$COMO is created during the installation process. This file can be looked at to make sure the installation was successful. During the installation process, the program is compiled, loaded, and copied to the directory GCBASE3>RUN. After loading, the binary and other temporary files are automatically deleted, and the source directory is cleaned up.

(C) Install the retrieval and output program. This is done by issuing the following commands:

```
A GCBASE3
R INS.GCRTRV.CPL          -or-          PH INS.GCRTRV.CPL
```

The log file INS.GCRTRV$COMO is created during the installation process. This file can be looked at to make sure the installation was successful. During the installation process, the program is compiled, loaded, and copied to the directory GCBASE3>RUN. After loading, the binary and other temporary files are automatically deleted, and the source directory is cleaned up.

(D) Install the input batch data program. This is done by issuing the following commands:

```
A GCBASE3
R INS.GCINPT.CPL          -or-          PH INS.CGRTRV.CPL
```

The log file INS.GCINPT$COMO is created during the installation process. This file can be looked at to make sure the installation was successful. During the installation process, the program is compiled, loaded, and copied to the directory GCBASE3>RUN. After loading, the binary and other temporary files are automatically deleted, and the source directory is cleaned up.

(E) Edit the file GCBASE3>SOURCE>BACCSS.MAIN.F77 and place the names of the special access groups in the array provided. The lines to be changed begin on about line 72 and look similar to the following:

```
 DATA GRP / ' 1) USGS/ND District','17)                ',
*           ' 2) USGS/Other     ','18)                ',
*           ' 3) State Water Comm','19)                ',
*           ' 4) Public Srvc Comm','20)                ',
*           ' 5)                ','21)                ',
*           ' 6)                ','22)                ',
*           ' 7)                ','23)                ',
*           ' 8)                ','24)                ',
*           ' 9)                ','25)                ',
*           '10)                ','26)                ',
*           '11)                ','27)                ',
*           '12)                ','28)                ',
*           '13)                ','29)                ',
*           '14)                ','30)                ',
*           '15)                ','31)                ',
*           '16)                ','32)                '/
```

Care must be taken not to change the length of the array fields. Each field is 20 characters long, but only the last 16 characters should be modified. A maximum of 32 special access groups may be specified. The names of the access groups are used only by the BACCSS program. Only the group number is used in the other GCBASE programs.

(F) Install the data-base initialization and special file update programs. This is done by issuing the following commands:

```
A GCBASE3
R BLD.AINITF.CPL
R BLD.BPARAM.CPL
R BLD.BACCSS.CPL
```

No log file is created during these installations. During the installation process, each program is compiled, loaded, and copied to the directory GCBASE3>RUN. After loading, the binary and other temporary files are automatically deleted, and the source directory is cleaned up.

(G) Create the subdirectory structure and initialize the data-base site and data files. This is done by issuing the following commands:

```
A GCBASE3>RUN
R CREATE_GCBASE
```

CREATE_GCBASE creates the data-file subdirectory structure and then runs the program AINITF to initialize the data-base site and data files. Step G should

be performed only during the initial installation of the data-base system. If step G is performed during a reinstallation of the software, existing data will be destroyed.

(H) Create and update the parameter index/definition file. This is done by issuing the following commands:

```
A GCBASE3>RUN
R BPARAM
```

The program displays prompts for the descriptions of the parameters to be used in the data base. Answer the prompts with the five-digit parameter code number, the data-type code (C=character, N=numeric), the number of digits left and right of the decimal (for numeric fields), the parameter field length, and a descriptive title. The current value of a parameter definition can be displayed by entering a carriage return when prompted for the data-type code. Changes are made by specifying the parameter code to be changed and entering the corrected values. To exit BPARAM, enter a carriage return when prompted for the parameter code.

(I) Create and update the user access file. This is done by issuing the following commands:

```
A GCBASE3>RUN
R BACCSS
```

The program displays prompts for descriptions of authorized users. Answer the prompts with the user ID of the individual authorized to use the data base, and the access that the user is to be allowed. Place a "1" under the digit representing each special access group to which the user belongs, then press the carriage return. To exit BACCSS, enter a carriage return when prompted for the user ID.

(J) Set the access rights and read/write locks on each of the subdirectories and files. Each of the subdirectories should have ALL access set for the data-base administrator(s). All data-base users need list, use, and read access to the GCBASE3 directory, and the RUN and DOCUMENTATION subdirectories. Data-base users also need list, use, read, and write access to the DATA subdirectory.

In the RUN subdirectory, the program files AINITF, BPARAM, and BACCSS should be protected from general use. Only the data-base administrator(s) should have access to these programs. The AINITF program is used only once and may be deleted after the data base has been initialized (step F).

In the DATA subdirectory, all of the files should be set to allow one writer or many readers. The files in the two subdirectories below DATA (SITE.IDA and SAMPLE.IDA) also should be set to allow one writer or many readers.

References

American National Standards Institute, Inc., 1978, American national standard programming language FORTRAN: New York, American National Standards Institute, Inc.

Atwood, J.W., and Bartholoma, S.D., 1991, Supporting different types of terminals in a distributed-information environment, in Bathrop, B.H., and Terry, J.E., eds., U.S. Geological Survey National Computer Technology Meeting--Proceedings, Phoenix, Arizona, November 14-18, 1988: U.S. Geological Survey Water-Resources Investigations Report 90-4162, p. 11-20.

Buck, Barbara, 1982, IDR5026 EMACS reference guide, revision 18.3: Framingham, Mass., Prime Computer, Inc.

Maddy, D.V., Lopp, L.E., Jackson, D.L., Coupe, R.H., and Schertz, T.L., 1990, National Water Information System user's manual, v. 2, chap. 2. Water-Quality System: U.S. Geological Survey Open-File Report 89-617.

Mathey, S.B., ed., 1990, National Water Information System user's manual, v. 2, chap. 4. Ground-Water Site Inventory System: U.S. Geological Survey Open-File Report 89-587.

P-STAT, Inc., 1986, P-STAT user's manual, version 8--Data management and statistical software for use on mainframe, mini and personal computers: Boston, PWS Publishers, Duxbury Press.

Glossary

AINITF Program used to initialize the data-base site and data files.

ASCII American Standard Code for Information Interchange. A system of representing the alphabet, numerals, punctuation, and special control codes using computer binary storage units. ASCII codes use 7 bits of an 8-bit byte. The eighth bit (called the parity bit) sometimes is used for error checking.

BACCSS Program used to create and update the user access file.

Bit A binary storage unit consisting of one cell. A bit can hold only one of two values. It is either "off" ("zero") or "on" ("one"). Bits usually are combined into groups called bytes.

BPARAM Program used to create and update the parameter index/definition file.

Byte A binary storage unit consisting of a group of bits (usually eight) used to store one piece of information. The byte can hold one of 256-bit patterns. On most systems, one 8-bit byte can store one ASCII character. Bytes are combined into "words" of two or more bytes to represent numbers and machine instructions.

DISCUR Distributed Information System Cursor Control Package.

EMACS A screen-oriented text editer available on many systems. The control code sequences available in the version of EMACS for the Prime minicomputer were adapted, where appropriate, to the screen-entry portion of the GCBASE program.

GCBASE Program used to input data into the data base and update data that have already been entered into the data base.

GCINPT Program used to input batch data into the data base.

GCRTRV Program used to retrieve and output selected data from the data base.

GCUTIL Program used to run special utilities against the data base.

ID Identification or identifier.

IDA The name chosen for the file structure used in the GCBASE program. It is derived from the term Indexed Direct Access.

Key The part of a data record used to index and retrieve a record. In the IDA file structure, the primary key is unique. Secondary keys are not necessarily unique and are used to provide an index to data records containing similar information.

MIDAS A proprietary data-file format available on Prime minicomputers. It includes built-in indexing and special retrieval software.

Glossary, Continued

NWIS National Water Information System.

P-STAT A statistical package available on Prime and other computer systems. It is a product owned and distributed by P-STAT, Inc.

PRIMOS The name of the operating system used on the Prime 50 series minicomputer.

String A group of consecutive bytes used to hold text data.

X-OFF An ASCII code used by computer systems to stop the transmission of data during data transfer.

X-ON An ASCII code used by computer systems to start the transmission of data during data transfer.