User's Manual for Geophysical Well Logging Software Programs

**G. M. Petrie
D. Gibson
S. C. Blair**

February 1983

**Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830**

**Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute**

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of 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. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151

NTIS Price Codes
Microfiche A01

Printed Copy

Pages	Price Codes
001-025	A02
026-050	A03
051-075	A04
076-100	A05
101-125	A06
126-150	A07
151-175	A08
176-200	A09
201-225	A010
226-250	A011
251-275	A012
276-300	A013

USER'S MANUAL FOR GEOPHYSICAL WELL LOGGING
SOFTWARE PROGRAMS

G. M. Petrie
D. Gibson
S. C. Blair

February 1983

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

SUMMARY

Since 1958 the Ground-Water Surveillance Program for the Hanford Site has made geophysical logging measurements in most of the 800 wells and deep boreholes that have been drilled on the Hanford Site. In 1980 the Pacific Northwest Laboratory (PNL), which conducts the Ground-Water Surveillance Program, began forming a computerized data base for storing and retrieving geophysical well log data and developing software for quantitative analysis of the well log data. This report, designed to serve as a user's guide, documents the data base system that handles the well log data.

Two programs, DIGLOG1 and LOGIT, are used to manipulate the data. The program DIGLOG1 translates analog paper strip charts into digital format; the program LOGIT is a general utility program that edits, displays, checks, stores, writes, and deletes sets of well log data. These two programs do not provide sophisticated display and analytical capabilities; rather, they provide programs that give the user easy access to powerful standard analytical software.

CONTENTS

SUMMARY	iii
INTRODUCTION	1
OPERATION OF THE PDP 11/55 FOR THE PROGRAM DIGLOG1.	4
REBOOTING PROCEDURES FOR THE PDP 11/55	4
DIGLOG1 EXAMPLE RUN	5
PROGRAM START-UP	5
DIGITIZING A SECTION.	6
CALIBRATION	8
FILE OUTPUT	10
EXITING THE PROGRAM	12
DISMOUNTING AND REMOVING THE DISC	13
WRITING FILES TO MAGNETIC TAPE ON THE PDP 11/55	13
OPERATION OF THE VAX 11/780 FOR THE PROGRAM LOGIT	15
OPERATION OF PROGRAM LOGIT	15
REFERENCES	24
APPENDIX A - WELL LOG DESCRIPTIONS	A.1
APPENDIX B - WELL NUMBERING SYSTEM	B.1
APPENDIX C - EXPLANATION OF ERROR MESSAGES	C.1
APPENDIX D - SYSTEM SOFTWARE	D.1
APPENDIX E - PROGRAM LISTING FOR PROGRAM DIGLOG1	E.1
APPENDIX F - PROGRAM LISTING FOR PROGRAM LOGIT	F.1

INTRODUCTION

Over 800 wells and deep boreholes have been drilled on the Department of Energy's Hanford Site in southeastern Washington. These wells provide information on the subsurface structure and composition of the site and allow groundwater movement to be monitored. Since 1958, properties of the subsurface environment of Hanford have been measured through geophysical logging of most of these boreholes and wells. This work has been done as part of the Ground-Water Surveillance Program for the Hanford Site, which is conducted by the Pacific Northwest Laboratory (PNL) for the U.S. Department of Energy.

These geophysical logging data form a substantial data set on the glacio-fluvial sediments that cover the Hanford Site to depths of nearly 1000 feet. Although the geophysical logs have been used in a few independent studies of wells on the Hanford Site (Gephart, Eddy and Deju 1979; Moore and Straight 1981), until recently no effort was made to consolidate the data set. In 1980, the Ground-Water Surveillance Program began the Geophysical Log Interpretation Task. The objectives of this task were to: 1) form a computerized data base for storing and retrieving geophysical well logging data, and 2) develop software for quantitative analysis of the well logging data.

This report documents the data base system that was formed to handle the geophysical well logging data and is designed to serve as a user's guide for the programs DIGLOG1 and LOGIT, which compose the data base (Figure 1). The report is divided into two sections: 1) a discussion of the operation of the PDP 11/55 for the DIGLOG1 program, and 2) a discussion of the operation of the VAX 11/780 for the LOGIT program. These two programs were not intended, in themselves, to provide sophisticated display and analytical capabilities; rather, they provide programs that allow the user easy access to powerful standard analytical software. For example, with program LOGIT the user can easily create files that MINITAB (see Appendix D) can accept. The user can then use MINITAB to do the statistical analysis. Likewise, the user can use program MAKGRAPH to provide sophisticated graphical output.

FIGURE 1. Schematic Representation of Well Log Software and Data Base Relationship

Appendix A discusses the well log data types most common on the Hanford Site. Appendix B is a guide to well names and locations on the Hanford Site. Appendix C explains error messages received through the programs. Appendix D discusses the software available on the PNL VAX systems. Appendices E and F give program listings for DIGLOG1 and LOGIT, respectively.

OPERATION OF THE PDP 11/55 FOR THE PROGRAM DIGLOG1

Three discs (DK0, DK1, and DK2) are associated with the PDP 11/55. DK0, the system disc, is permanently stored in the 11/55, where systems libraries are loaded. DK1 is a temporary storage disk. The user can initialize DK1 and temporarily store data on it. For the particular operation described in this section, the user assigns DK1 as a File-Q disc so that it is accessible as a temporary storage disc for File-Q files.

DK2 is manually loaded into the 11/55. In this case, load the disc containing the program, DIGLOG1, by inserting it into the DK2 disc drive and switch the button beneath the disk drive to the 'RUN' position. Next, type the following instructions on the terminal to mount the disc, set the User's Identification Code (UIC), and set DK1 as a File-Q disc:

```
>MOU DK2:/OVR
>SET/UIC=[207,1]
>DMO DK1:
>FLQ DK1:/ND
```

Once these instructions are completed, proceed to run the program, DIGLOG1. If the system crashes during operation, the following section can be used to explain rebooting procedures.

REBOOTING PROCEDURES FOR THE PDP 11/55

If the PDP 11/55 crashes during operation of a program, use the following procedure to reboot the system.

- The load address for the PDP 11/55 is 773000. Check to see that switches 9, 10, and 12 through 17 are flipped up on the face of the computer and that all others are down.
- On the hardcopy terminal, type 'DK' and hit the carriage return.

- Wait for a pause in the output from this hardcopy terminal. The computer will ask for the date and time in this format: Hour:minute day-month-year. For example, type on the hard copy terminal:

14:31 22-FEB-82

and hit the carriage return. A three-letter abbreviation is used for the month and the time is always based on the 24-hour clock.

- The computer will print out several lines, which end with @<EOF>. When the computer finishes with this hardcopy printout, it is ready for use. Remember, rebooting always dismounts discs and the system loses the user's UIC. Therefore, remount the DK2 disc and reset the UIC as previously described. DO NOT reinitialize DK1 if you wish to save or edit pre-rebooting files on DK1. Such 'new directing' (i.e., ND) will destroy these files.

DIGLOG1 EXAMPLE RUN

PROGRAM START-UP

To run the program DIGLOG1, type the following instructions on the terminal:

```
>RUN DK2:DIGLOG1
```

A carriage return will cause the following to be displayed on the terminal:

```
SET PRINT OPTION (0-5)
ENTER ZERO FOR NO PRINTER OUTPUT
```

The set print option allows the user to output information to the printer. The option numbers indicate the amount be printed. Users will normally enter option 0. The other options are generally of interest only to programmers. Entering a zero will result in the following display:

SET DIGITIZING INCREMENT (IN)

The digitizing increment determines how closely the digitized points will be spaced. The standard setting of .05 will digitize a point every .05 inches along the log trace. This increment will be satisfactory for most well logs but can be changed to accomodate very complex logs.

DIGITIZING A SECTION

Once the print option and digitizing increment have been set, the user is ready to begin digitizing. Tape the logs securely to the digitizing table, making sure that the section to be digitized does not extend into the unlighted areas on the left and right margins of the table. The program will crash if the cursor is moved into these areas and the user will have to begin again at PROGRAM START-UP. Lifting the cursor off the surface of the table will also crash the program. In this event, the x,y coordinates displayed on the front of the digitizer will be frozen and a yellow light will appear under these coordinates in the 'lock' position. To restore the digitizer to its operating condition, replace the cursor on the table and press the red button. The program cannot be restarted when the yellow light is on.

Execution of the initial run command will display the following menu on the VS-60 screen:

```
DIGITIZE SECTION
CALIBRATE LOG
DISPLAY CHART
RESTART
FILE OUTPUT
STOP
```

The remainder of this example run demonstrates how to use this menu to:
1) digitize and calibrate a section, 2) check the section for errors, and
3) save the information in a file. Begin by pressing the light pen on the

DIGITIZE SECTION option displayed on the VS-60 screen. The following operating instructions will be displayed on the terminal:

TO OPERATE THIS PROGRAM

- (1) - DIGITIZE A SECTION OF A CHART
- (2) - DISPLAY CHART TO CHECK FOR ERRORS
- (3) - CALIBRATE THE LOG OR CHART
- (4) - FILE OUTPUT WRITES A CALIBRATED DATA TO A BINARY FILE

DIGITIZE SECTION

- (1) - START NEW LOG
- (2) - CONTINUE SAME LOG
- (3) - ESCAPE

Option one, START NEW LOG, is used when beginning a new log. Option two, CONTINUE SAME LOG, is used for appending data to an existing file. This may be necessary if the program dies or if the user wishes to temporarily stop the program. The data in the current file are stored on the temporary storage disc, DK1, and remain accessible through this option. The ESCAPE option and those to follow have been added to allow the user to exit the current section if the program does not advance to the next section.

Typing a '1' for START NEW LOG displays the following on the terminal:

DIGITIZE TWO POINTS ON DEPTH AXIS
USE BLUE BUTTON

Digitizing the two points sets up the geometry of the logs. Select any two points on the depth (x) axis; do not enter values for these points. Place the cursor on the depth axis within the section to be digitized. Press the blue button. Move the cursor to another point on the depth axis and again press the blue button. Digitizing the points causes the following to be displayed on the terminal:

TO DIGITIZE STRING
PRESS WHITE BUTTON TO START STRING AND
PRESS BLUE BUTTON TO END STRING

Note the beginning and ending depths (x values) of the section of the log to be digitized. These values will be needed for calibration. To digitize, press the white button and move the cursor along the log and press the blue button at the end of the section. Check the log on the display screen for errors by pressing the light pen on the DISPLAY CHART option on the display screen. If the log is not acceptable, the user may digitize it again by pressing the light pen on DIGITIZE SECTION.

CALIBRATION

After the log has been digitized and checked for errors the user is ready to calibrate the log. Press the light pen on the CALIBRATION option on the display screen. The following will be displayed on the terminal:

TO CALIBRATE CHART
(1) - DO CALIBRATION
(2) - ESCAPE

Type a '1' on the terminal for calibration. The ESCAPE option can be used to exit this section. Typing a '1' causes the following question to be displayed on the terminal:

DO YOU WANT TO CALIBRATE THE LOG IN SECTIONS
Y, N

This option allows the user to digitize the log in several sections and to calibrate each section separately. The user will want to answer yes (typing 'Y') when the log is too large to fit on the digitizing table and must be repositioned several times. In this case, the following calibration steps will be repeated for each section. Answer no (typing 'N') for a log consisting of only one section, and the following will then be displayed:

ENTER MIN AND MAX DEPTHS (X) OF SECTION

These values are the beginning and ending depths of the digitized section. The smaller value is entered first and the two values are separated by a comma. Entering these values results in the following display.

DIGITIZE TWO POINTS ON DEPTH AXIS USE BLUE BUTTON

Choose any two points on the depth axis within the digitized part of the log for which x values can be easily determined. Digitize a small value first and then a larger one. Digitizing these points causes the following to be displayed:

ENTER DEPTHS (X VALUES) OF THESE COORDINATES (AMIN, AMAX)

The smaller x value is entered first and then the larger value. These are separated by a comma. Entering these values results in the following display:

ENTER STARTING DEPTH FOR RECORD

This value represents the starting depth of the digitized section and is the starting depth of the log only when the entire log is being digitized. Enter this value and the following will be displayed:

DIGITIZE TWO POINTS ON Y AXIS (AMIN, AMAX)

Again, use the blue button and choose any two points on the depth axis for which the exact values are known. Digitizing these points results in the following display:

ENTER VALUES FOR THESE POINTS (AMIN, AMAX)

As with the x points, the smaller value is entered first and the two are separated by a comma. After these values are entered on the terminal, the original digitized x,y values will be converted to the calibrated values. When all the points have been converted, the calibration is finished and the following will be displayed on the terminal:

CALIBRATION FINISHED

FILE OUTPUT

The user is now ready to create an output file. Press the light pen on the FILE OUTPUT option on the display screen. The following will be displayed on the terminal:

FILE OUTPUT

TYPE OF CHART IS:

- (1) - BOREHOLE GEOPHYSICS
- (2) - OTHER
- (3) - ESCAPE

For any chart other than a geophysical well log, use the OTHER option. In this case the user will be asked only for a file name and file storage location. These two requests will be discussed in the following file output example. Type a '1' on the terminal for geophysical well logs and the following will be displayed on the terminal:

OUTPUT FILE

ENTER WELL DESIGNATION (AREA-COORD1-COORD2)

See Appendix B for an explanation of well names. The names must be entered exactly as they appear in Table B.1. After the well designation is entered, the following will be displayed:

ENTER TYPE OF LOG

- (1) - GAMMA GAMMA (DENSITY)
- (2) - NATURAL GAMMA
- (3) - NEUTRON EPITHERMAL NEUTRON
- (4) - CALIPER
- (5) - SONIC
- (6) - SP & R
- (7) - NEUTRON GAMMA
- (8) - SP
- (9) - TEMP
- (10) - FLOWMETER
- (11) - OTHER

If the digitized log is not listed, the OTHER option will be selected. The user will then be asked to enter a three-letter identifying code. This code can be any three letters the user chooses, but because it will be used to identify the log type the code should be meaningful. Entering a log type will result in the following display:

ENTER DATE LOGGED (MO-DAY-YR)

This date is taken from the well log header. It represents the date the log was made, not the date it is digitized. Enter the date and the following will be displayed:

LOG DIGITIZED BY

The user's initials are required here with a maximum of three characters allowed. Enter these and the following will be displayed:

OUTPUT FILE

- (1) - DB0
- (2) - DK2

The file will be written to one of the above discs. It can be written to the same disc that contains the program (DK2) or can be put on DB0 for transfer to some other device. Select one of the options and the following will be displayed:

FILE NAME (28A1)

The file name must consist of 1 to 9 characters followed by a period and then 3 more characters, for example, WLOG.SP1. The first character cannot be a number. After entering a name the file will be written to the previously selected disc.

At this point, the user may wish to digitize another log. Pressing the light pen on the RESTART option on the VS-60 screen will enable the user to begin the process again, starting with the DIGITIZE SECTION option. If no additional digitizing is to be done, the user is ready to exit the program.

EXITING THE PROGRAM

Press the light pen on the STOP option on the display screen and the following will be displayed on the terminal:

```
TTO -- STOP  
HIT <CR> TO EXIT>
```

Press the carriage return to exit from the program. This option may be used at any time during the operation of the program but none of the data will be saved until the file is written to a disc.

DISMOUNTING AND REMOVING THE DISC

After exiting from the program, type the following on the terminal to dismount the user's DK2 disc:

```
>DMO DK2:
```

The user is now ready to remove the disc. Push the button under the DK2 disc drive to the 'LOAD' position. Wait until the disc drive clicks (about 2 minutes). Open the disc drive door and remove the disc.

WRITING FILES TO MAGNETIC TAPE ON THE PDP 11/55

After creating a number of files, the user will transfer these files from their temporary storage location on either the DK2 or DBO disc to magnetic tape for transfer to the VAX. This is accomplished through the steps below:

1. Manually load the tape onto the tape drive and thread it onto the empty take-up spool.
2. Press the buttons on the tape drive in the following order:

LOAD
ON LINE

3. Mount the disc containing the files to be transferred according to the instructions given on page 4.
4. Mount the tape by typing 'MOU MT0:/OVR' on the terminal.
5. To transfer a file, type 'PIP MT0:=DK2:XXX.XXX', where the x's represent the name of the file. A number of files can be transferred at the same time through the use of a wild card (i.e., replace text with a *). For example, if the user types 'PIP MT0:=DK2:TEST.*', all files beginning with TEST will be transferred. 'PIP MT0:=DK2:*.LOG' will transfer all files ending with LOG.
6. After the transfer is complete, type a 'z' while at the same time holding down the CTRL key. This will take the computer out of the PIP mode. Now the user may dismount the tape by typing 'DMO MT0:' on the terminal.
7. DK2 must also be dismounted by typing 'DMO DK2:'.
8. To remove the tape from the tape drive, push the rewind button on the tape drive. The tape will then be rewound and ready to be removed.

OPERATION OF THE VAX 11/780 FOR THE PROGRAM LOGIT

Begin a session on the VAX by typing 'C' on the terminal keyboard while also holding down the CTRL key. The computer will respond with a prompt of 'Username', to which the user must answer by typing in the word 'LOGS'. The computer will then ask for the password by typing 'Password', to which the user responds by typing in the word 'NEW'. At this stage, the user can run LOGIT by typing the phrase 'RUN LOGIT':

On occasion, the user will need to copy information from a tape to the system disk before running program LOGIT. To do so, the user must follow the steps below.

1. Thread the tape on the tape drive according to the instructions posted on the inside of the tape drive door.
2. Push 'LOAD' and then push the 'ON LINE' button.
3. Return to the terminal and type in 'MOUNT MTO:/OV=ID'.
4. Next, type 'COPY MTO:*. *.*'.
5. When this is done, type 'DISMOUNT MTO:'.
6. Return to the tape drive and push the 'REW/UNLD' button before removing the tape.

After running LOGIT, the last step the user must make is to log off the system. Do this by typing in the phrase 'LO'.

OPERATION OF PROGRAM LOGIT

Program LOGIT is a menu-driven program that allows the user to edit, display, check, store, write, and delete well logs. The user should not have any fear about 'crashing' the system because the program was designed to recognize obvious errors and correct for them (see Appendix C). To further lessen any fears the new user may have of damaging the data base, a special account has been set up to allow the user to 'play' with the software without damaging the real data. The user need only respond to the 'Username' prompt of the computer

with 'LOGIT' instead of 'LOGS'. Using this option the user can experiment with the system while reading the following pages. This approach would normally be the best way to learn the system.

Upon typing 'RUN LOGIT' the user is presented with the following options:

- (0) STOP
- (1) LIST ALL WELLS ON FILE FOR AN AREA
- (2) MAKE A SUMMARY OF ALL WELLS ON THE PRINTER
- (3) ADD A NEW LOG
- (4) ENTER THE NAME OF THE WELL YOU WANT TO WORK ON

By entering the above values, the user can invoke the events below.

- When a '0' is entered, the programs stop.
- When a '1' is entered, the user indicates an interest in getting a well listing, which is currently stored on the system. The computer will respond by asking for the Area the user is interested in. The user must respond with the appropriate number (i.e., 100, 200, 300, 600, 1100, or 3000 Area of the Hanford Site). The computer will then provide a listing of all the wells currently stored on the system for that Area (Figure 2).
- When a '2' is entered, the computer creates a summary of all the wells currently stored on the system on the line printer (see Figure 3).
- When a '3' is entered, the user indicates an interest in entering a new well log into the system. The computer responds by asking for the name of the file where the new data are stored. This file is the same one that was created with program DIGLOG on the 11/55 (Figure 1). If the computer has not already done so, it may ask for the user's name. This allows future users to contact the person who has the most current history for this well log. During the addition of a well log to the system, the computer does some elementary checking. If errors are found, the computer

```

$ RUN LOGIT
PICK ONE OF THE FOLLOWING
(0) STOP
(1) LIST ALL WELLS ON FILE FOR AN AREA
(2) MAKE A SUMMARY OF ALL WELLS ON THE PRINTER
(3) ADD A NEW LOG
(4) ENTER THE WELL NAME YOU WANT TO WORK WITH
1
ENTER THE AREA(100,200,300,600,1100 OR 3000)
600
6-84-33
      1 GAMMA GAMMA (DENSITY)      LOG(S) ON FILE
      1 NATURAL GAMMA             LOG(S) ON FILE
      1 NEUTRON EPITHERMAL NEUTRON LOG(S) ON FILE
      1 CALIPER                   LOG(S) ON FILE
      1 TEMPERATURE               LOG(S) ON FILE
6-10-54B
      1 GAMMA GAMMA (DENSITY)      LOG(S) ON FILE
      1 NATURAL GAMMA             LOG(S) ON FILE
      1 NEUTRON EPITHERMAL NEUTRON LOG(S) ON FILE
      1 CALIPER                   LOG(S) ON FILE
      1 TEMPERATURE               LOG(S) ON FILE
6-15-15G
      1 GAMMA GAMMA (DENSITY)      LOG(S) ON FILE
      1 NEUTRON EPITHERMAL NEUTRON LOG(S) ON FILE
      1 CALIPER                   LOG(S) ON FILE
      1 TEMPERATURE               LOG(S) ON FILE
6-15-15F
      1 GAMMA GAMMA (DENSITY)      LOG(S) ON FILE
      1 NATURAL GAMMA             LOG(S) ON FILE
      1 NEUTRON EPITHERMAL NEUTRON LOG(S) ON FILE
      1 CALIPER                   LOG(S) ON FILE
6-48-48
      1 GAMMA GAMMA (DENSITY)      LOG(S) ON FILE
      1 NATURAL GAMMA             LOG(S) ON FILE
      1 NEUTRON EPITHERMAL NEUTRON LOG(S) ON FILE
      1 CALIPER                   LOG(S) ON FILE
      1 TEMPERATURE               LOG(S) ON FILE
6-48-49
      1 NATURAL GAMMA             LOG(S) ON FILE
      1 NEUTRON EPITHERMAL NEUTRON LOG(S) ON FILE
      1 SONIC                     LOG(S) ON FILE
      1 TEMPERATURE               LOG(S) ON FILE
6-815-E1G
      1 GAMMA GAMMA (DENSITY)      LOG(S) ON FILE
      1 NATURAL GAMMA             LOG(S) ON FILE
      1 NEUTRON EPITHERMAL NEUTRON LOG(S) ON FILE
      1 CALIPER                   LOG(S) ON FILE
      1 TEMPERATURE               LOG(S) ON FILE
6-44-70
      1 NATURAL GAMMA             LOG(S) ON FILE
      1 CALIPER                   LOG(S) ON FILE

```

FIGURE 2. Sample Output Showing Well Logs From the 600 Area Stored in the Data Base

TYPE	VERSION	REFL	CASE ELEVATION		COORDINATES	DEPTH	RANGE	DATE CREATED
GAMMA GAMMA (DENSITY)	1	6-44-33	343.1	PN	33859.00 W 84489.00	2400.0	2650.1	10-14-81
NATURAL GAMMA	1	6-44-33	343.1	PN	33859.00 W 84489.00	2400.0	2700.8	10-13-81
NEUTRON EPI THERMAL NEUTRON	1	6-44-33	343.1	PN	33859.00 W 84489.00	2400.0	2899.3	10-14-81
CALIPER	1	6-44-33	343.1	PN	33859.00 W 84489.00	2400.0	2699.5	10-8-81
TEMPERATURE	1	6-44-33	343.1	PN	33859.00 W 84489.00	2400.0	2700.7	10-8-81
GAMMA GAMMA (DENSITY)	1	6-10-54b	516.4	PN	53687.00 W 10125.00	2550.0	2799.6	6-5-81
NATURAL GAMMA	1	6-10-54b	516.4	PN	53687.00 W 10125.00	2550.0	2898.5	6-4-81
NEUTRON EPI THERMAL NEUTRON	1	6-10-54b	516.4	PN	53687.00 W 10125.00	2550.0	2800.3	6-2-81
CALIPER	1	6-10-54b	516.4	PN	53687.00 W 10125.00	2550.0	2900.7	6-1-81
TEMPERATURE	1	6-10-54b	516.4	PN	53687.00 W 10125.00	2550.0	2899.7	6-1-81
GAMMA GAMMA (DENSITY)	1	6-15-15G	544.5	PN	14862.00 W 14956.00	2950.0	3198.9	10-27-81
NEUTRON EPI THERMAL NEUTRON	1	6-15-15G	544.5	PN	14862.00 W 14956.00	2950.0	3200.9	10-27-81
CALIPER	1	6-15-15G	544.5	PN	14862.00 W 14956.00	2950.0	3300.1	10-27-81
TEMPERATURE	1	6-15-15G	544.5	PN	14862.00 W 14956.00	2950.0	3299.3	8-28-81
GAMMA GAMMA (DENSITY)	1	6-15-15F	546.2	PN	14831.00 W 14901.00	2950.0	3299.7	11-3-81
NATURAL GAMMA	1	6-15-15F	546.2	PN	14831.00 W 14901.00	2950.0	3300.4	11-3-81
NEUTRON EPI THERMAL NEUTRON	1	6-15-15F	546.2	PN	14831.00 W 14901.00	2950.0	3200.0	11-3-81
CALIPER	1	6-15-15F	546.2	PN	14831.00 W 14901.00	2950.0	3295.9	11-2-81
GAMMA GAMMA (DENSITY)	1	6-48-48	570.0	PN	48253.00 W 47947.00	2300.0	2499.7	10-05-81
NATURAL GAMMA	1	6-48-48	570.0	PN	48253.00 W 47947.00	2300.0	2598.9	10-2-81
NEUTRON EPI THERMAL NEUTRON	1	6-48-48	570.0	PN	48253.00 W 47947.00	2300.0	2795.8	10-5-81
CALIPER	1	6-48-48	570.0	PN	48253.00 W 47947.00	2300.0	2598.2	10-1-81
TEMPERATURE	1	6-48-48	570.0	PN	48253.00 W 47947.00	2300.0	2599.4	10-2-81
NATURAL GAMMA	1	6-48-49	571.5	PN	48253.00 W 47838.00	2300.0	2598.8	4-6-72
NEUTRON EPI THERMAL NEUTRON	1	6-48-49	571.5	PN	48253.00 W 47838.00	2300.0	2598.9	4-6-72
SONIC	1	6-48-49	571.5	PN	48253.00 W 47838.00	2300.0	2599.7	7-19-69
TEMPERATURE	1	6-48-49	571.5	PN	48253.00 W 47838.00	2300.0	2599.6	9-29-81
GAMMA GAMMA (DENSITY)	1	6-S15-E15	404.0		0.00 0.00	2480.0	2700.3	8-25-81
NATURAL GAMMA	1	6-S15-E15	404.0		0.00 0.00	2480.0	2950.0	8-18-81
NEUTRON EPI THERMAL NEUTRON	1	6-S15-E15	404.0		0.00 0.00	2480.0	2700.5	8-25-81
CALIPER	1	6-S15-E15	404.0		0.00 0.00	2480.0	2800.0	6-18-81
TEMPERATURE	1	6-S15-E15	404.0		0.00 0.00	2480.0	2859.5	8-18-81
NATURAL GAMMA	1	6-44-70	733.2	PN	70156.00 W 43801.00	2800.0	3069.0	9-10-77
CALIPER	1	6-44-70	733.2	PN	70156.00 W 43801.00	2800.0	3099.9	9-10-77
NATURAL GAMMA	1	6-17-47	576.8	PN	47316.00 W 17333.00	6.0	314.9	11-7-80
NATURAL GAMMA	1	6-17-5	433.2	PN	4500.00 W 17450.00	4.0	61.7	4-15-81
NATURAL GAMMA	1	6-15-26	523.8	PN	26370.00 W 15602.00	6.0	171.7	5-9-80
NATURAL GAMMA	1	6-9-E2	418.1	PN	2324.00 E 8577.00	4.0	64.0	4-16-81
NATURAL GAMMA	1	6-8-32	554.4	PN	31800.00 W 7875.00	4.0	172.2	9-11-80
NATURAL GAMMA	1	6-H-25	509.3	PN	25003.00 E 7995.00	6.0	158.1	9-11-80
NATURAL GAMMA	1	6-8-17	522.4	PN	17125.00 W 8200.00	6.0	153.6	9-16-80
NATURAL GAMMA	1	6-4-E6	500.0		0.00 0.00	6.0	88.1	4-16-81
NATURAL GAMMA	1	6-3-45	504.5	PN	45007.00 W 3007.00	6.0	124.1	4-10-80
NATURAL GAMMA	1	6-2-33A	536.4	PN	32823.00 W 1813.00	6.0	169.8	9-11-80
NATURAL GAMMA	1	6-2-3	477.1	PN	3325.00 W 1900.00	6.0	115.6	9-11-80
NATURAL GAMMA	1	6-1-18	537.7	PN	17593.00 W 1420.00	6.0	162.4	9-11-80
NATURAL GAMMA	1	6-S3-25	523.5	PS	24501.00 W 2956.00	4.0	166.2	9-10-80
NATURAL GAMMA	1	6-S6-E4J	420.0		0.00 0.00	4.0	75.9	2-27-80
NATURAL GAMMA	1	6-S6-E4H	420.0		0.00 0.00	4.0	62.0	2-27-80
NATURAL GAMMA	1	6-S6-E4G	423.1	PS	4204.00 E 6761.00	6.0	62.0	2-27-80
NATURAL GAMMA	1	6-S6-E4F	418.7	PS	3950.00 E 6595.00	4.0	60.0	2-27-80
NATURAL GAMMA	1	6-S6-E4E	425.7	PS	3727.00 E 6446.00	4.0	66.0	2-27-80
NATURAL GAMMA	1	6-S6-E4C	433.0	PS	3781.00 E 6100.00	6.0	445.0	2-25-80
NATURAL GAMMA	1	6-S6-E4A	445.3	PS	3503.00 E 6102.00	6.0	63.7	9-4-80
NATURAL GAMMA	1	6-S8-19	503.8	PS	18720.00 W 8336.00	4.0	127.3	9-10-80
NATURAL GAMMA	1	6-S12-3	435.5	PS	2700.00 W 12325.00	6.0	104.0	9-10-80
NATURAL GAMMA	1	6-S14-20	492.7	PS	19739.00 W 14491.00	4.0	135.7	2-28-80
NATURAL GAMMA	1	6-S19-E13	394.5	PS	12780.00 E 19434.00	6.0	75.8	9-4-80
NATURAL GAMMA	1	6-S24-19	427.8	PS	18974.00 W 23965.00	6.0	67.9	2-24-80
NEUTRON EPI THERMAL NEUTRON	1	6-S24-19	427.8	PS	18974.00 W 23965.00	6.0	68.3	4-10-80
NATURAL GAMMA	1	6-S27-E14	399.8	PS	15104.00 E 52104.00	6.0	104.0	9-17-80

FIGURE 3. Sample Output Created by Option 2

will post the appropriate error message to the terminal. The proper responses to these messages are given in Appendix C.

- When a '4' is entered, the user indicates an interest in working with a well currently stored on the system. The computer responds by asking for the well name. If there is more than one type of well log associated with this well, the user will be asked which type he wants to work on. Several replicates of a well log may be kept on the system. If this is the case, the user will be asked for the version wanted. Once the exact well is defined, the above menu is supplemented with seven more options. These options are:

- (5) PLOT OUT THE LOG
- (6) DUMP THE LOG TO THE PRINTER
- (7) SAMPLE THE LOG AT EVEN INTERVALS
- (8) DELETE THE LOG
- (9) CREATE A FILE OF RAW X AND Y POINTS
- (10) CREATE A ROCKWELL FORMATTED FILE
- (11) EDIT THE LOG

By entering values 5 through 11, the user may invoke one of the following events:

- By entering a '5', the user indicates that he is interested in plotting out the currently active well log. However, as of September, 1982, the graphics on the VAX 11/780 are not yet on line. Thus, the computer can only respond with an apologetic message. However, the user still has the option of creating a file of the data the user is interested in (see the description for options 6 and 9 below), taking it over to the 11/70, and using the graphics capabilities of that system (see Appendix D). As soon as the VAX 11/780 graphics capabilities come on line, the graphics modules will be linked into the program.
- By entering a '6', the user causes all the information contained in this log to be printed out (Figure 4).

FIGURE 4. Sample of the Information Given About a Specific Well by Using Option 6

NATURAL GAS A OF ELEV. 0-50-440 (VERSION: 1) HAS A ELEVATION OF 423.1											
A L ELEV. RANGE IN 0.0000000 61.967945											
THE LISTED VALUES RANGE OVER 54.0P5114 83.449936											
THE LOG GAS CODE IS 2-27-80											
AND GAS ENTERED IN THE STATE ON 21-SEP-82 BY D.GIBSON											
1	8.00000	63.2141	1	45	26.4391	71.2645	1	89	43.7245	64.5725	1
2	8.95801	62.5749	1	46	26.7984	68.8674	1	90	43.8842	67.0696	1
3	7.37625	61.3704	1	47	27.3572	66.7699	1	91	44.7225	66.5678	1
4	8.14591	58.9792	1	48	27.5169	64.2729	1	92	45.4011	70.5654	1
5	8.92809	60.6772	1	49	27.9560	66.5702	1	93	45.9999	72.5630	1
6	9.07385	63.1742	1	50	28.3952	64.2729	1	94	46.6782	73.7615	1
7	9.55269	65.3716	1	51	28.6746	66.7699	1	95	47.4770	75.7592	1
8	10.0314	63.1742	1	52	28.7944	69.2669	1	96	48.3552	77.0576	1
9	10.5103	64.7723	1	53	29.2335	71.5642	1	97	48.6746	79.4547	1
10	11.0896	62.3752	1	54	29.3932	74.0612	1	98	48.7145	76.9577	1
11	11.4271	59.6782	1	55	29.4730	76.5582	1	99	49.1137	74.6805	1
12	12.3074	61.6707	1	56	29.9520	74.3608	1	100	50.1117	75.1599	1
13	12.7405	63.3740	1	57	30.4311	72.1635	1	101	50.6306	72.9625	1
14	13.6555	61.1700	1	58	31.1896	73.8614	1	102	51.0698	75.2598	1
15	13.6631	58.6796	1	59	31.3892	76.3584	1	103	51.3093	77.7568	1
16	13.7036	61.1766	1	60	31.6287	78.8554	1	104	52.2274	76.7580	1
17	14.1038	63.4739	1	61	32.0678	81.1527	1	105	52.6266	74.4807	1
18	15.1437	65.9709	1	62	32.5069	83.4499	1	106	52.7065	71.9637	1
19	14.3034	65.4679	1	63	32.8662	81.0528	1	107	52.8262	69.4667	1
20	14.7824	66.2705	1	64	32.9660	76.5558	1	108	53.4650	67.4691	1
21	15.2615	64.6731	1	65	33.4251	76.2586	1	109	54.1037	69.4667	1
22	15.7604	65.2705	1	66	34.7445	73.8614	1	110	54.3432	71.9637	1
23	15.9800	63.7735	1	67	33.9041	71.3644	1	111	55.0219	70.0660	1
24	16.2195	61.2705	1	68	34.0239	68.8674	1	112	55.3811	72.4631	1
25	16.5788	63.6735	1	69	34.3033	71.3644	1	113	56.2594	71.2645	1
26	17.3772	65.2717	1	70	34.6626	68.9673	1	114	56.7384	69.0672	1
27	17.9769	67.2693	1	71	34.9421	66.4703	1	115	57.0578	66.6700	1
28	18.2954	64.6722	1	72	35.6606	68.2681	1	116	57.1775	64.1730	1
29	18.4790	62.9745	1	73	35.8203	70.7651	1	117	57.1775	61.5761	1
30	19.2135	60.4775	1	74	36.3393	68.5676	1	118	57.2973	59.0791	1
31	20.0214	61.6700	1	75	36.4590	66.0708	1	119	57.3771	56.5821	1
32	20.3503	63.4739	1	76	37.0179	68.1682	1	120	58.0558	58.4798	1
33	21.0238	65.9709	1	77	37.2973	65.6712	1	121	58.1755	60.9769	1
34	21.6483	67.7667	1	78	37.8163	67.8686	1	122	58.3352	63.4739	1
35	22.0064	65.3697	1	79	37.9361	65.3716	1	123	58.5348	65.9709	1
36	23.3652	64.6722	1	80	38.6147	64.4739	1	124	59.0139	63.7735	1
37	23.5676	62.3752	1	81	39.6127	62.6748	1	125	59.3332	61.3764	1
38	24.2834	64.1730	1	82	40.6107	62.8746	1	126	59.4530	58.6794	1
39	24.7225	61.6758	1	83	41.2095	60.7771	1	127	59.6925	56.3824	1
40	24.9640	59.4768	1	84	42.0478	62.1754	1	128	60.3711	58.2801	1
41	25.3213	61.7759	1	85	42.2873	64.6724	1	129	60.6107	55.7831	1
42	25.4819	64.2729	1	86	42.5668	67.0696	1	130	61.0897	57.9804	1
43	25.6407	66.7699	1	87	43.0059	64.7723	1	131	61.7284	55.9828	1
44	25.8094	69.2669	1	88	43.2853	62.2753	1	132	61.9679	54.0851	1

- By entering a '7', the user indicates an interest in creating a file of the active well log sampled at evenly spaced intervals. The computer responds by asking the user to enter a name for this new file. The name of the new file must start with a letter. The computer then presents a menu from which the user can choose one of three depth origins:

- (1) USE GROUND LEVEL FOR THE ORIGIN
- (2) USE SEA LEVEL FOR THE ORIGIN
- (3) USE 1000 FEET ABOVE SEA LEVEL FOR THE ORIGIN

If the user picks option 1 above, no offset is added to the depth coordinates of the well. If the user picks option 2 above, sea level is taken as the origin, which means that all values above sea level will take on negative values and all values below sea level will take on positive values. This option is important when the user is going to do an analysis where several logs will be used because it removes the effects of topography. Option 3 allows the user to create well logs that will all have a common origin but also have only positive numbers.

The computer next asks the user for the depth where the sampling is to start, the distance between sample points, and the number of points to be created.

- By entering an '8', the user causes the well log to be deleted from the system.
- By entering a '9', the user indicates an interest in creating a file of the raw depth data points. Normally, this option is used to create files to 'feed' into other software (such as MINITAB). The computer responds by asking for a file name and origin in the same manner as described with option 6 above. The major difference is that the data are not sampled at evenly spaced points but are simply written out 'as is'.
- By entering a '10', the user indicates an interest in writing out the well log in a format that can be easily interfaced with the Rockwell well logging system. The computer responds by asking for a file name and the

origin in the same manner as described above. In this case, the well log data are written out in a format that can be interfaced easily with the Rockwell system.

- By entering an '11', the user indicates the desire to edit the log. The computer responds by presenting another menu, which is described below.

Upon invoking option 11, the user is presented with the following options which apply to the 'active' well log chosen with option 4. These include:

```
----->(1) REMOVE POINTS BY DEPTH
----->(2) REMOVE POINTS BY NUMBER
----->(3) CHANGE POINTS BY DEPTH
----->(4) CHANGE POINTS BY NUMBER
----->(5) PLOT THE CURRENT LOG
----->(6) PRINT THE CURRENT LOG
----->(7) REPLACE THE OLD DATA WITH THE NEW
----->(8) ALL DONE
```

- By entering a '1', the user commands the computer to remove all data points that fall within a certain depth range. The computer responds by asking for minimum and maximum depths. The computer then removes these points that fall within this range.
- As shown in Figure 4, each point of a well log has an identification number associated with it. By entering a '2', the user commands the computer to remove all data points that fall between any two of these numbers. The computer responds by asking for the minimum and maximum numbers. All points falling within this range are then removed.
- By entering a '3', the user commands the computer to modify all data points within a certain depth range. The computer responds by asking for the range of depth. The user can now modify all data within this range by first adding a constant value to all the data points and/or multiplying each value by a constant. To do this, the computer asks the user for the amount of offset and the multiplication factor. It then transforms each point, within the given depth range, with the equation:

$\text{New value} = \text{old value} * \text{multiplication factor} + \text{offset}$

- By entering a '4', the user invokes the same effect as defined for option 3 above. The only difference is that the selection is done with the identification numbers.
- As defined above, option 5 plots out the log.
- As defined above, option 6 writes out all the information on this log out to the printer.
- All the changes the user may make with edit options 1 through 4 are only done on a scratch file. Until the user enters a '7', none of these changes are permanent. However, once he does enter a '7', all the changes are made permanent. Thus, if the user makes a mistake he can recover by entering an '8' (see below) without entering a '7' and the computer will forget any editing commands that have been made since the last edit option 7 was invoked.
- By entering an '8', the user commands the computer to leave the editing mode.

REFERENCES

Blair, S. C., L. S. Law and J. W. Lindberg. 1981. A Catalog of Borehole Geophysics on the Hanford Site, 1958 to 1980. PNL-3504, Pacific Northwest Laboratory, Richland, Washington.

Gephart, R. E., P. A. Eddy and R. A. Deju. 1979. Geophysical Logging and Hydrologic Testing of Deep Basalt Flows in the Rattlesnake Hills Well Number One. RHO-BWI-ST-1, Rockwell Hanford Operations, Richland, Washington.

Moore, B. A., and S. R. Straight. 1981. "Geophysical Investigations as Applied to Hydrogeologic Studies on the Hanford Site, Southcentral Washington." EOS 62, No. 45.

APPENDIX A

WELL LOG DESCRIPTIONS

APPENDIX A

WELL LOG DESCRIPTIONS

This appendix presents a brief background discussion for the seven types of geophysical logging data most commonly available for wells on the Hanford Site. Much of this discussion is taken from Keys and McCary (1972) who present a very thorough analysis of the use of geophysical well logs in geohydrology; the reader is referred to their work if more detail is required. The information presented here summarizes 1) what each particular log measures, 2) the appropriate use of the log at Hanford, 3) the radius of investigation of the log, and 4) general comments regarding use of the log. Most logging measurements at Hanford were made by V. L. McGhan at PNL; however, a considerable number of logs are available from other sources, including Washington State University and the major logging companies.

Table A.1 presents a summary of log types commonly available for wells located in each of the major areas at Hanford. Information on availability of a particular log type for a given well is presented by Blair et al. (1980).

Table A.1 shows that four nuclear logs are commonly run at Hanford: Natural Gamma (NAG), Gamma-Gamma (-), Neutron Epithermal Neutron (NEN), and the Neutron Gamma (NG). Each of these logs provides a measurement of fundamental particles emitted from atomic nuclei. Nuclear logs have an advantage in that they can be taken in cased or open holes that are filled with any type of fluid or air.

TABLE A.1. Hanford Area Log Types

<u>Log Type</u>	<u>100</u>	<u>200E</u>	<u>200W</u>	<u>300</u>	<u>600</u>
Gamma Gamma	X	X	X	X	X
Neutron Gamma					X
Neutron Epithermal Neutron	X	X	X	X	X
Natural Gamma	X	X	X	X	X
Caliper					X
Sonic		X			X
Spontaneous Potential and Resistivity					X

NATURAL GAMMA

Type of measurement--This log measures the natural gamma radiation in the borehole.

Use--This log is commonly used for lithologic and stratigraphic correlation. Most Natural Gamma radiation in the borehole environment comes from the potassium isotope ^{40}K ; Natural Gamma probes are often used to identify layers of clay or shale that are rich in potassium. At the Hanford Site they would be used to identify clay layers and detrital deposits.

Radius of Investigation--This is 6 to 12 inches.

Comments--Natural Gamma logs are less sensitive than other nuclear probes to changes in borehole fluid and other near-probe efforts.

NEUTRON GAMMA

Type of Measurement--This type of geophysical log is essentially measurement of in situ neutron activation. Neutrons are introduced into the borehole environment from a neutron source mounted in the probe; these neutrons are moderated and subsequently captured by atomic nuclei of materials in the borehole. The nuclei that capture neutrons then become unstable and emit gamma photons which are detected by a sensor on the probe.

Use--Neutron Gamma logs are very sensitive to the chemistry of fluids in the borehole, especially to increased chlorine content. When compared with Neutron Epithermal Neutron logs, the Neutron Gamma logs can be used to detect increased salinity of borehole fluid. Neutron Gamma logs can also be used to measure clay content, and oxygen, carbon, calcium, and iron under good conditions.

Radius of Investigation--This is 6 inches in high-porosity, water-saturated rocks, and 2 feet in low-porosity, dry rocks.

Comments--Neutron Gamma logs are most commonly used in conjunction with Neutron Epithermal Neutron logs. They are somewhat insensitive to borehole effects and may be used in either cased or open holes which are either wet or dry. Hole diameter has an effect on the logs and the use of a side-collimated

or centered tool is recommended. Data from washed out intervals are unreliable. Although the probe may be used with any type of fluid or mud in the hole, the particular mud or fluid will affect the probe. Changes in the fluid composition will have a more pronounced effect on Neutron Gamma than on Neutron Epithermal Neutron probes.

NEUTRON EPITHERMAL NEUTRON

Type of Measurement--Neutrons from a source on the probe are introduced into the borehole environment. A sensor on the probe measures the amount of moderation of emitted neutrons. Hydrogen is the most effective moderator for neutrons and the epithermal neutrons have the highest percentage of response to hydrogen; thus, the epithermal neutron probe measures the amount of hydrogen moderation of the emitted neutrons.

Use--At depths above the water table, this probe yields data on moisture content vs depth. Below the water table, the NEN log indicates the total porosity of the formation with depth.

Radius of Investigation--This is 6 inches in high-porosity, water-saturated rocks, and 2 feet in low-porosity, dry rocks.

Comments--This probe is somewhat insensitive to borehole effects and may be used in open or cased holes, which are either wet or dry. However, hole diameter has an effect and use of a centered or side-collimated tool is recommended. Also data from washed out intervals are unreliable. Although the probe may be used with any type of fluid or mud in the hole, the particular fluid will affect the probe.

This log is plotted with neutron intensity increasing to the right. Moisture content and saturated porosity show a log response increasing to the left.

GAMMA-GAMMA (DENSITY)

Type of Measurement--This probe uses a gamma ray source and detector. The detector measures radiation that is back-scattered and/or attenuated in the borehole.

Use--This log is used to correlate lithology and to indicate porosity and bulk density. It assists in well maintenance as it can be used to identify cavities in cement grout outside the well casing. The bulk density of material in the borehole is derived from back-scattered electrons because the gamma radiation absorbed is proportional to the electron density of the material penetrated. The electron density is proportional to the bulk density.

Total porosity can be derived using the equation:

$$\text{Porosity} = \frac{P_g - P_b}{P_g - P_f}$$

where P_g = grain density
 P_b = bulk density
 P_f = fluid density.

Radius of Investigation--This is 16 inches.

Comments--Generally, errors are 0.03 to 0.04 g/cc; however, this depends somewhat on hole diameter. Also, fluid in the hole porosity of the rock affects the radius of investigation. Generally, the gamma-gamma log is plotted with radiation and porosity increasing to left; density increases to right.

CALIPER

Type of Measurement--This log measures the average diameter of the borehole as a function of depth. Caliper probes generally use arms or feelers hinged at the upper end of the probe, or bow springs that follow the wall of the hole. Each feeler changes the resistance of a potentiometer. Outputs from the component potentiometers are averaged and transmitted to surface recording equipment. The recorded data (generally calibrated in inches) give the average hole diameter.

Use--The caliper logs:

- locate fractures and cracks in the borehole
- correct the interpretation of other logs for borehole diameter effects
- provide data on well construction and condition
- correlate with other logs to identify lithology and stratigraphy.

Radius of Investigation--The radius is up to 40 inches.

Comments--Caliper logs are generally reliable and repeatable; however, thick mud on walls of the borehole may affect the sensor. This tool may be used in any type of borehole.

SONIC

Type of Measurement--Sonic velocity (acoustic) probes measure the travel time of an acoustic wave between a transmitter and receiver mounted in the probe. The probe is designed so that the quickest path between the two receivers will be for a wave refracted along the borehole wall. Recorded data measure both interval transit time (Δt measured in seconds per foot) and wave amplitude.

Use--Sonic logs are used primarily to measure of rock porosity in open holes (Picket 1960). The porosity measurement is based on Equation A.1 first presented by Wyllie (1963), where

$$\frac{1}{V_r} = \Delta t = \frac{\phi}{V_f} + \frac{1-\phi}{V_m} \quad (A.1)$$

In this equation Δt = transit time, ϕ = porosity, V_f = velocity in pore fluid, V_m = velocity in rock matrix, and V_r = velocity in total rock body.

This is a time-averaging equation which assumes that the rock porosity is uniformly distributed and that computed values should be interpreted as representing effective porosity. Detailed analyses of sonic logs can also be used to evaluate bonding of cement between casing and formation and to approximate engineering properties of rocks at depth.

Radius of Investigation--This is 0.75 ft to 3.5 ft. The radius of investigation depends on frequency and wavelength of the sonic waves, and on the elastic properties of the rock itself. Pirson (1963) shows the radius of investigation to be $r \cong 3\lambda$, where

$$\lambda = V_r/n$$

and V_r = velocity in rock mass, and n = frequency of generated wave.

Comments--Sonic logs must be run in fluid-filled holes and although these logs have potential for providing extensive quantitative information on the subsurface, they are very sensitive to the borehole environment and should be used in concert with other types of porosity logs.

ELECTRIC LOGS

Electric logs use a variety of logging techniques too numerous to pursue in this brief appendix. Table A.2 summarizes the applicability of each log type and the reader is referred to Keys and MacCary (1972) for a more detailed discussion of individual types of electric logs.

REFERENCES

- Blair, S. C., L. S. Law and J. W. Lindberg. 1981. A Catalog of Borehole Geophysics on the Hanford Site, 1958 to 1980. PNL-3504, Pacific Northwest Laboratory, Richland, Washington.
- Keys, W. Scott and L. M. MacCary. 1972. Application of Borehole Geophysics to Water-Resources Investigations. Techniques of Water-Resources Investigations of the U.S. Geol. Survey, United States Government Printing Office.
- Pickett, G. R. 1960. "The Use of Acoustic Logs in the Evaluation of Sandstone Reservoirs." Geophysics 25:1(250-274).
- Pirson, S. J. 1963. Handbook of Well Log Analysis. Prentice-Hall, Inc., Englewood Cliffs, N.J.
- Wyllie, M. R. J. 1963. The Fundamentals of Well Log Interpretation. Academic Press, New York.

TABLE A.2. Hydrologic Applicability of Electric Logs
(Keys and MacCary 1972)

Properties to be Investigated	Type of Electric Log								
	Single Point	Short Normal	Long Normal	Lateral Device	Resistivity (Nonfocused)	Guard and Laterolog	Micro-focused	Induction	SP
Lithologic correlation	X	X	---	---	---	X	---	X	X
Bec thickness	X	X	---	---	X	X	X	X	X
Formation resistivity (low R muds)	---	---	X	X	---	X	---	X	---
Formation resistivity (fresh mud)	---	---	X	X	---	---	---	X	---
Invaded Zone resistivity	---	X	---	---	---	---	---	---	---
Flushed zone resistivity	---	---	---	---	X	---	X	---	---
Mud resistivity ¹ (in place in hole)	---	---	---	---	X	---	---	---	---
Formation water resistivity	---	---	X	X	---	X	---	---	---

¹Use mud kit for pit samples

APPENDIX B

WELL NUMBERING SYSTEM

APPENDIX B

WELL NUMBERING SYSTEM

In the open literature well structures on the Hanford Site are identified by a prefix containing the number 99. The rest of the number that designates a specific well is determined mainly by its location on the Hanford Site. The designation includes three groups of symbols. The first group consists of the number 99 preceded by a number identifying the area in which the well is located, for example, 199, 299, 399, 699, 1199, 3099.

100 AND 200 AREAS

Wells located in the 100 and 200 Areas of the Hanford Site are further identified in the second group of symbols according to the designation of the particular area (E, W, B, D, K, F, H, N), followed by the number of the sheet map encompassing that portion of the area in which the well is located. The sheet maps are shown on official second level maps for each area. For example, the first two-symbol groups for a well located within the area described by Sheet Map 24 of the 200-E Area would be 299-E24-. The 100-K Area and the 100-N Area second level maps are not further divided into sheet maps so the second group of symbols for these areas is simply 'K' and 'N'. The third-symbol group in the well number identifies the specific structure within the sheet map area. In some cases, the numbering system is arbitrary; but where practical, numbers were chosen in accordance with a previous numbering system. For example, the well formerly designated 361-B-6 now has the number 299-E28-6.

Some of the monitoring wells in the 200 Area are dry wells (i.e., wells that do not extend to the water table). These have been differentiated from deeper wells in that all of the shallow wells have third-group numbers greater than 50. In some cases wells have been designated with 100 and 200 Area numbers even though they are actually located outside the area fence. These structures are monitoring wells adjacent to ground disposal facilities that are located outside the area. Monitoring wells adjacent to the BC-Cribs are examples of this situation.

300 AREA

Wells in the 300 Area are designated in a manner similar to that described for the 100 and 200 Areas. The single difference occurs in the second symbol group, which has no area designation, but which has simply the 300 Area sheet map number.

600 AREA

The 600 Area includes all of the Hanford Site outside the limited access areas. Well numbers in this area are prefixed by 699. The second and third groups of numbers of 600 Area wells consist of the north and west plant coordinates, respectively, rounded off to the nearest 1000 feet. For example, a well located at plant coordinates, N25665 and W14554, would be designated 699-26-15. If the well is located south or east of the plant coordinate origin, an S or E is used with the appropriate number. Some 600 Area wells are located within 1000 feet of each other; these have letters (A, B, C, etc.) following the numbers for identification. A few wells located outside the Hanford Site boundaries have also been given 600 Area designations.

EXCEPTIONS USED IN THIS STUDY

To save space and computer time, and to reduce the chance of typing errors, the identification number 99 is left out of the well names for this study. Table B.1 is a list of all well names in this abbreviated form.

TABLE B.1. Hanford Site Well Names

1-B3-1	2-E33-1	2-W15-170	2-W23-68
1-B3-2	2-E33-1	2-W15-171	2-W23-69
1-B4-1	2-E33-1	2-W15-172	2-W23-7
1-B4-2	2-E33-1	2-W15-173	2-W23-70
1-B4-3	2-E33-1	2-W15-174	2-W23-71
1-B4-4	2-E33-1	2-W15-175	2-W23-72
1-B5-1	2-E33-1	2-W15-176	2-W23-73
1-B8-1	2-E33-1	2-W15-177	2-W23-74
1-B8-3	2-E33-1	2-W15-178	2-W23-75
1-B8-4	2-E33-1	2-W15-179	2-W23-77
1-B8-5	2-E33-1	2-W15-180	2-W23-78
1-B9-1	2-E33-1	2-W15-181	2-W23-79
1-D2-5	2-E33-1	2-W15-182	2-W23-8
1-D3-1	2-E33-1	2-W15-184	2-W23-80
1-D5-10	2-E33-1	2-W15-185	2-W23-81
1-D5-11	2-E33-1	2-W15-186	2-W23-82
1-D5-12	2-E33-1	2-W15-188	2-W23-83
1-D8-2	2-E33-1	2-W15-189	2-W23-84
1-D8-3	2-E33-1	2-W15-193	2-W23-85
1-F5-1	2-E33-1	2-W15-194	2-W23-86
1-F5-2	2-E33-1	2-W15-195	2-W23-87
1-F5-3	2-E33-1	2-W15-196	2-W23-88
1-F5-4	2-E33-1	2-W15-197	2-W23-89
1-F5-5	2-E33-1	2-W15-2	2-W23-9
1-F5-6	2-E33-1	2-W15-3	2-W23-90
1-F7-1	2-E33-1	2-W15-4	2-W23-91
1-F8-1	2-E33-1	2-W15-5	2-W23-92
1-F8-2	2-E33-1	2-W15-51	2-W23-93
1-H3-1	2-E33-1	2-W15-52	2-W23-94
1-H4-1	2-E33-1	2-W15-53	2-W23-95
1-H4-2	2-E33-1	2-W15-54	2-W23-96
1-H4-3	2-E33-1	2-W15-55	2-W23-97
1-K-1	2-E33-1	2-W15-56	2-W23-98
1-K-10	2-E33-1	2-W15-57	2-W23-99
1-K-11	2-E33-1	2-W15-58	2-W26-1
1-K-12	2-E33-1	2-W15-59	2-W26-2
1-K-14	2-E33-1	2-W15-6	2-W26-3
1-K-16	2-E33-1	2-W15-60	2-W26-4
1-K-17	2-E33-1	2-W15-61	2-W26-5
1-K-18	2-E33-1	2-W15-67	2-W6-1
1-K-19	2-E33-1	2-W15-68	3-1-2
1-K-2	2-E33-1	2-W15-69	3-1-3
1-K-20	2-E33-1	2-W15-7	3-1-4
1-K-21	2-E33-1	2-W15-70	3-1-5
1-K-22	2-E33-1	2-W15-71	3-1-6
1-K-23	2-E33-1	2-W15-72	3-2-1
1-K-24	2-E33-1	2-W15-73	3-2-2
1-K-25	2-E33-1	2-W15-74	3-2-3
1-K-26	2-E33-1	2-W15-75	3-3-1
1-K-3	2-E33-1	2-W15-76	3-3-10
1-K-4	2-E33-1	2-W15-77	3-3-11
1-K-5	2-E33-1	2-W15-78	3-3-2
1-K-6	2-E33-1	2-W15-79	3-3-3
1-K-7	2-E33-1	2-W15-8	3-3-5
1-K-8	2-E33-1	2-W15-80	3-3-8

TABLE B.1. (contd)

1-K-9	2-E33-182	2-W1 5-81	3-3-9
1-N-1	2-E33-183	2-W1 5-82	3-4-1
1-N-12	2-E33-184	2-W1 5-84	3-4-10
1-N-14	2-E33-185	2-W1 5-85	3-4-2
1-N-15	2-E33-186	2-W1 5-86	3-4-3
1-N-2	2-E33-187	2-W1 5-87	3-4-5
1-N-3	2-E33-188	2-W1 5-88	3-4-7
1-N-4	2-E33-189	2-W1 5-89	3-4-8
1-N-5	2-E33-19	2-W1 5-9	3-4-9
1-N-6	2-E33-190	2-W1 5-90	3-5-1
1-N-7	2-E33-191	2-W1 5-91	3-5-2
1-N-8P	2-E33-192	2-W1 5-92	3-5-3
1-N-8O	2-E33-193	2-W1 5-93	3-6-1
1-N-8R	2-E33-194	2-W1 5-95	3-8-1
1-N-8S	2-E33-195	2-W1 5-96	3-8-2
1-N-8T	2-E33-196	2-W1 5-97	3-8-3
1-N-8V	2-E33-197	2-W1 5-98	30-41-14
1-N-9	2-E33-198	2-W1 5-99	30-42-16
11-12-13	2-E33-199	2-W1 8-1	30-45-18A
11-14-13	2-E33-2	2-W1 8-10	30-45-18B
11-14-14	2-E33-20	2-W1 8-100	30-47-18A
11-15-13C	2-E33-200	2-W1 8-101	6-1-18
11-16-12	2-E33-201	2-W1 8-102	6-10-54
11-16-13D	2-E33-202	2-W1 8-103	6-10-54B
11-17-13A	2-E33-203	2-W1 8-104	6-10-E12
11-17-13B	2-E33-204	2-W1 8-105	6-101-48A
11-18-13	2-E33-206	2-W1 8-107	6-101-48B
11-19-11	2-E33-207	2-W1 8-109	6-101-48C
11-19-13C	2-E33-208	2-W1 8-110	6-102-48
11-20-12	2-E33-209	2-W1 8-113	6-107-79
11-20-13	2-E33-21	2-W1 8-114	6-11-45A
11-20-14	2-E33-210	2-W1 8-115	6-11-45B
11-20-17	2-E33-211	2-W1 8-116	6-111-24
11-21-12	2-E33-212	2-W1 8-117	6-113-38
11-22-11A	2-E33-213	2-W1 8-118	6-114-60
11-22-11B	2-E33-214	2-W1 8-119	6-12-1
11-22-13A	2-E33-215	2-W1 8-120	6-13-64
11-22-13B	2-E33-216	2-W1 8-121	6-14-38
11-22-13C	2-E33-217	2-W1 8-122	6-14-47
11-22-13D	2-E33-218	2-W1 8-123	6-14-E6P
11-22-16A	2-E33-219	2-W1 8-124	6-14-E6Q
11-22-16B	2-E33-22	2-W1 8-125	6-14-E6R
11-22-16C	2-E33-220	2-W1 8-126	6-14-E6S
11-29-12B	2-E33-221	2-W1 8-127	6-14-E6T
11-30-14	2-E33-222	2-W1 8-128	6-15-15A
11-31-19	2-E33-223	2-W1 8-129	6-15-15B
11-32-17	2-E33-224	2-W1 8-13	6-15-15C
11-33-13A	2-E33-225	2-W1 8-130	6-15-15F
11-33-13B	2-E33-226	2-W1 8-131	6-15-15G
11-33-18B	2-E33-227	2-W1 8-132	6-15-26
11-33-18C	2-E33-228	2-W1 8-133	6-15-52
11-33-18D	2-E33-229	2-W1 8-134	6-15-E13
11-33-18E	2-E33-23	2-W1 8-135	6-17-47
11-33-18F	2-E33-230	2-W1 8-136	6-17-5
11-34-13	2-E33-231	2-W1 8-137	6-17-70

TABLE B.1. (contd)

11-34-15	2-E33-232	2-W18-138	6-17-93
11-34-19	2-E33-233	2-W18-139	6-19-47A
11-38-11	2-E33-234	2-W18-14	6-19-51
11-39-10A	2-E33-235	2-W18-140	6-19-58
11-39-10B	2-E33-236	2-W18-141	6-19-88
11-39-16A	2-E33-237	2-W18-142	6-2-3
11-39-16B	2-E33-24	2-W18-143	6-2-33A
11-39-16C	2-E33-240	2-W18-144	6-2-E14
11-39-16D	2-E33-241	2-W18-145	6-2-E19
11-40-15	2-E33-242	2-W18-146	6-20-20
11-40-16A	2-E33-243	2-W18-147	6-20-39
11-40-16B	2-E33-244	2-W18-148	6-20-E19
11-41-13A	2-E33-245	2-W18-152	6-20-E5P
11-41-13B	2-E33-246	2-W18-153	6-20-E5Q
11-41-13C	2-E33-248	2-W18-154	6-20-E5R
11-41-15	2-E33-249	2-W18-155	6-20-E5S
11-45-12	2-E33-25	2-W18-156	6-20-E5T
2-E13-1	2-E33-250	2-W18-157	6-22-55
2-E13-10	2-E33-251	2-W18-2	6-22-70
2-E13-11	2-E33-252	2-W18-3	6-24-1P
2-E13-12	2-E33-253	2-W18-4	6-24-1Q
2-E13-13	2-E33-254	2-W18-5	6-24-1R
2-E13-14	2-E33-255	2-W18-51	6-24-1S
2-E13-15	2-E33-256	2-W18-52	6-24-1T
2-E13-16	2-E33-257	2-W18-53	6-24-33
2-E13-17	2-E33-258	2-W18-54	6-24-46
2-E13-18	2-E33-259	2-W18-55	6-25-51
2-E13-19	2-E33-26	2-W18-56	6-25-55
2-E13-2	2-E33-260	2-W18-57	6-25-70
2-E13-20	2-E33-261	2-W18-58	6-25-80
2-E13-21	2-E33-262	2-W18-59	6-26-15
2-E13-3	2-E33-263	2-W18-6	6-26-89
2-E13-4	2-E33-264	2-W18-60	6-27-47
2-E13-5	2-E33-273	2-W18-61	6-27-8
2-E13-51	2-E33-274	2-W18-62	6-28-40
2-E13-52	2-E33-277	2-W18-63	6-28-49
2-E13-6	2-E33-3	2-W18-64	6-28-52
2-E13-7	2-E33-4	2-W18-65	6-28-55
2-E13-8	2-E33-5	2-W18-66	6-29-78
2-E13-9	2-E33-51	2-W18-69	6-3-45
2-E16-1	2-E33-52	2-W18-7	6-30-47
2-E16-2	2-E33-53	2-W18-8	6-30-51
2-E17-1	2-E33-54	2-W18-84	6-30-55
2-E17-10	2-E33-55	2-W18-85	6-31-31
2-E17-2	2-E33-56	2-W18-86	6-31-53A
2-E17-3	2-E33-57	2-W18-87	6-31-53B
2-E17-4	2-E33-58	2-W18-88	6-31-65
2-E17-5	2-E33-59	2-W18-89	6-32-22
2-E17-50	2-E33-6	2-W18-9	6-32-42
2-E17-6	2-E33-60	2-W18-90	6-32-43
2-E17-7	2-E33-61	2-W18-91	6-32-49A
2-E17-8	2-E33-62	2-W18-92	6-32-62
2-E17-9	2-E33-63	2-W18-93	6-32-70
2-E19-1	2-E33-64	2-W18-94	6-32-72
2-E23-1	2-E33-65	2-W18-95	6-32-77

TABLE B.1. (contd)

2-E23-2	2-E33-66	2-W18-96	6-33-38
2-E24-1	2-E33-67	2-W18-97	6-33-42
2-E24-10	2-E33-68	2-W18-98	6-33-56
2-E24-11	2-E33-69	2-W18-99	6-34-39A
2-E24-12	2-E33-7	2-W19-1	6-34-39B
2-E24-13	2-E33-70	2-W19-10	6-34-41
2-E24-2	2-E33-71	2-W19-2	6-34-42
2-E24-3	2-E33-72	2-W19-3	6-34-51
2-E24-4	2-E33-73	2-W19-4	6-34-88
2-E24-5	2-E33-74	2-W19-5	6-35-27
2-E24-51	2-E33-75	2-W19-53	6-35-66
2-E24-52	2-E33-76	2-W19-54	6-35-70
2-E24-53	2-E33-77	2-W19-55	6-35-78
2-E24-54	2-E33-78	2-W19-56	6-35-9
2-E24-55	2-E33-79	2-W19-57	6-36-46P
2-E24-56	2-E33-8	2-W19-58	6-36-46Q
2-E24-57	2-E33-80	2-W19-59	6-36-46R
2-E24-58	2-E33-81	2-W19-60	6-36-46S
2-E24-59	2-E33-82	2-W19-61	6-36-61A
2-E24-6	2-E33-83	2-W19-62	6-36-61B
2-E24-60	2-E33-84	2-W19-63	6-36-93
2-E24-64	2-E33-85	2-W19-64	6-37-43
2-E24-68	2-E33-86	2-W19-65	6-37-82A
2-E24-8	2-E33-87	2-W19-66	6-37-82B
2-E24-9	2-E33-88	2-W19-67	6-38-65
2-E25-1	2-E33-89	2-W19-68	6-38-70
2-E25-10	2-E33-9	2-W19-69	6-39-39
2-E25-100	2-E33-90	2-W19-7	6-39-79
2-E25-101	2-E33-92	2-W19-70	6-4-E6
2-E25-102	2-E33-93	2-W19-71	6-40-1
2-E25-103	2-E33-94	2-W19-72	6-40-33A
2-E25-104	2-E33-95	2-W19-73	6-40-62
2-E25-105	2-E33-96	2-W19-74	6-41-23
2-E25-106	2-E33-97	2-W19-75	6-42-12A
2-E25-107	2-E33-98	2-W19-76	6-42-42
2-E25-108	2-E33-99	2-W19-8	6-43-104
2-E25-109	2-E34-1	2-W21-1	6-43-42
2-E25-11	2-W10-1	2-W21-51	6-43-88
2-E25-110	2-W10-10	2-W21-52	6-43-89
2-E25-111	2-W10-101	2-W21-53	6-44-64
2-E25-112	2-W10-102	2-W21-54	6-44-70
2-E25-113	2-W10-103	2-W21-55	6-45-42
2-E25-114	2-W10-104	2-W21-56	6-45-69
2-E25-115	2-W10-105	2-W21-57	6-46-21
2-E25-116	2-W10-106	2-W21-58	6-47-35A
2-E25-117	2-W10-107	2-W21-59	6-47-35B
2-E25-118	2-W10-108	2-W21-60	6-47-46
2-E25-119	2-W10-109	2-W21-61	6-47-51
2-E25-12	2-W10-11	2-W21-62	6-47-60
2-E25-121	2-W10-110	2-W21-63	6-48-48
2-E25-122	2-W10-111	2-W21-64	6-48-71
2-E25-123	2-W10-112	2-W21-65	6-49-13
2-E25-124	2-W10-113	2-W21-66	6-49-28
2-E25-13	2-W10-114	2-W21-67	6-49-48
2-E25-14	2-W10-115	2-W21-68	6-49-55

TABLE B.1. (contd)

2-E25-16	2-W10-116	2-W21-69	6-49-57
2-E25-17	2-W10-117	2-W21-70	6-49-79
2-E25-18	2-W10-118	2-W21-71	6-49-95
2-E25-19	2-W10-119	2-W21-72	6-49100B
2-E25-2	2-W10-12	2-W21-73	6-49100C
2-E25-20	2-W10-120	2-W21-74	6-50-18
2-E25-3	2-W10-121	2-W21-75	6-50-19
2-E25-4	2-W10-122	2-W21-76	6-50-28B
2-E25-5	2-W10-123	2-W21-77	6-50-30
2-E25-51	2-W10-124	2-W21-78	6-50-42
2-E25-53	2-W10-125	2-W21-79	6-50-53
2-E25-54	2-W10-126	2-W21-80	6-50-85
2-E25-55	2-W10-127	2-W21-81	6-51-36
2-E25-56	2-W10-128	2-W21-82	6-51-63
2-E25-57	2-W10-129	2-W21-83	6-51-75
2-E25-6	2-W10-130	2-W22-1	6-52-52
2-E25-61	2-W10-133	2-W22-10	6-53-103
2-E25-62	2-W10-134	2-W22-11	6-53-35
2-E25-63	2-W10-135	2-W22-12	6-53-47
2-E25-66	2-W10-136	2-W22-13	6-53-55A
2-E25-67	2-W10-137	2-W22-14	6-53-55B
2-E25-68	2-W10-138	2-W22-15	6-54-17B
2-E25-69	2-W10-139	2-W22-16	6-54-17C
2-E25-7	2-W10-140	2-W22-17	6-54-18
2-E25-70	2-W10-141	2-W22-18	6-54-19
2-E25-71	2-W10-142	2-W22-19	6-54-34
2-E25-72	2-W10-143	2-W22-2	6-54-42
2-E25-73	2-W10-144	2-W22-20	6-54-45
2-E25-74	2-W10-145	2-W22-21	6-54-57
2-E25-75	2-W10-146	2-W22-22	6-55-40
2-E25-76	2-W10-147	2-W22-23	6-55-44
2-E25-77	2-W10-148	2-W22-24	6-55-50A
2-E25-78	2-W10-149	2-W22-25	6-55-50B
2-E25-79	2-W10-150	2-W22-26	6-55-50C
2-E25-8	2-W10-151	2-W22-27	6-55-50D
2-E25-80	2-W10-152	2-W22-28	6-55-57
2-E25-81	2-W10-153	2-W22-29	6-55-60A
2-E25-82	2-W10-154	2-W22-3	6-55-60B
2-E25-83	2-W10-161	2-W22-30	6-55-63
2-E25-84	2-W10-168	2-W22-31	6-55-65A
2-E25-85	2-W10-176	2-W22-32	6-55-65B
2-E25-86	2-W10-2	2-W22-33	6-55-70
2-E25-87	2-W10-3	2-W22-34	6-55-76
2-E25-88	2-W10-4	2-W22-35	6-55-89
2-E25-89	2-W10-5	2-W22-36	6-55-95
2-E25-9	2-W10-51	2-W22-37	6-56-26A
2-E25-90	2-W10-52	2-W22-38	6-56-43
2-E25-91	2-W10-53	2-W22-4	6-57-25A
2-E25-92	2-W10-54	2-W22-5	6-57-25B
2-E25-97	2-W10-55	2-W22-51	6-57-29B
2-E25-98	2-W10-56	2-W22-52	6-57-83
2-E25-99	2-W10-57	2-W22-53	6-58-24
2-E26-1	2-W10-58	2-W22-54	6-59-32
2-E26-2	2-W10-59	2-W22-55	6-59-58
2-E26-3	2-W10-60	2-W22-56	6-59-80A

TABLE B.1. (contd)

2-E26-4	2-W10-61	2-W22-57	6-59-80B
2-E26-5	2-W10-62	2-W22-58	6-60-32
2-E26-6	2-W10-63	2-W22-59	6-60-57
2-E26-7	2-W10-64	2-W22-6	6-60-60
2-E27-1	2-W10-65	2-W22-60	6-61-16A
2-E27-100	2-W10-66	2-W22-61	6-61-26A
2-E27-101	2-W10-67	2-W22-62	6-61-37
2-E27-102	2-W10-68	2-W22-63	6-61-41
2-E27-103	2-W10-69	2-W22-65	6-61-55
2-E27-104	2-W10-70	2-W22-66	6-61-62
2-E27-105	2-W10-71	2-W22-67	6-61-66
2-E27-106	2-W10-72	2-W22-7	6-61-84
2-E27-107	2-W10-73	2-W22-8	6-62-31
2-E27-108	2-W10-74	2-W22-9	6-62-32
2-E27-109	2-W10-75	2-W23-1	6-62-43A
2-E27-115	2-W10-76	2-W23-10	6-62-43B
2-E27-116	2-W10-8	2-W23-100	6-62-43C
2-E27-117	2-W10-80	2-W23-101	6-62-43D
2-E27-118	2-W10-82	2-W23-102	6-62-43E
2-E27-119	2-W10-83	2-W23-103	6-62-43F
2-E27-120	2-W10-84	2-W23-104	6-62-43H
2-E27-121	2-W10-85	2-W23-105	6-62-43I
2-E27-122	2-W10-86	2-W23-106	6-62-43J
2-E27-124	2-W10-87	2-W23-107	6-62-43K
2-E27-2	2-W10-88	2-W23-108	6-62-43L
2-E27-3	2-W10-89	2-W23-109	6-62-43M
2-E27-5	2-W10-9	2-W23-11	6-62-43N
2-E27-51	2-W10-90	2-W23-110	6-63-25A
2-E27-52	2-W10-91	2-W23-111	6-63-25B
2-E27-53	2-W10-92	2-W23-112	6-63-51
2-E27-54	2-W10-93	2-W23-113	6-63-55
2-E27-55	2-W10-94	2-W23-114	6-63-58
2-E27-56	2-W10-95	2-W23-115	6-63-89
2-E27-57	2-W10-96	2-W23-116	6-63-90
2-E27-6	2-W10-97	2-W23-117	6-63-92
2-E27-65	2-W10-98	2-W23-118	6-64-27
2-E27-66	2-W10-99	2-W23-119	6-64-62
2-E27-67	2-W11-1	2-W23-120	6-65-50
2-E27-68	2-W11-10	2-W23-121	6-65-59A
2-E27-69	2-W11-11	2-W23-122	6-65-59B
2-E27-70	2-W11-12	2-W23-123	6-65-59C
2-E27-71	2-W11-13	2-W23-124	6-65-83
2-E27-72	2-W11-14	2-W23-125	6-65-95
2-E27-73	2-W11-15	2-W23-132	6-66-103
2-E27-74	2-W11-16	2-W23-133	6-66-23
2-E27-75	2-W11-17	2-W23-134	6-66-38
2-E27-76	2-W11-18	2-W23-135	6-66-39
2-E27-77	2-W11-19	2-W23-136	6-66-58
2-E27-78	2-W11-2	2-W23-137	6-66-64
2-E27-79	2-W11-20	2-W23-138	6-66-91
2-E27-80	2-W11-21	2-W23-139	6-67-51
2-E27-81	2-W11-23	2-W23-140	6-67-86
2-E27-82	2-W11-24	2-W23-141	6-67-98
2-E27-83	2-W11-26	2-W23-142	6-68-105
2-E27-84	2-W11-3	2-W23-143	6-69-45

TABLE B.1. (contd)

2-E27-85	2-W11-4	2-W23-144	6-70-68
2-E27-86	2-W11-5	2-W23-145	6-71-30
2-E27-87	2-W11-51	2-W23-146	6-71-52
2-E27-88	2-W11-52	2-W23-147	6-71-77
2-E27-89	2-W11-53	2-W23-148	6-72-73
2-E27-90	2-W11-54	2-W23-149	6-72-92
2-E27-91	2-W11-55	2-W23-150	6-74-23
2-E27-92	2-W11-56	2-W23-151	6-74-44
2-E27-93	2-W11-57	2-W23-152	6-74-48
2-E27-94	2-W11-58	2-W23-153	6-74-60
2-E27-95	2-W11-59	2-W23-154	6-75-23A
2-E27-96	2-W11-6	2-W23-155	6-75-23B
2-E27-97	2-W11-60	2-W23-156	6-77-36
2-E27-98	2-W11-61	2-W23-157	6-77-54
2-E27-99	2-W11-62	2-W23-158	6-78-62
2-E28-1	2-W11-63	2-W23-159	6-79-104
2-E28-10	2-W11-64	2-W23-160	6-8-17
2-E28-11	2-W11-65	2-W23-161	6-8-25
2-E28-15	2-W11-66	2-W23-162	6-8-32
2-E28-16	2-W11-67	2-W23-163	6-80-39B
2-E28-17	2-W11-68	2-W23-164	6-80-43P
2-E28-18	2-W11-69	2-W23-165	6-80-43Q
2-E28-19	2-W11-7	2-W23-166	6-80-43R
2-E28-2	2-W11-70	2-W23-167	6-80-43S
2-E28-20	2-W11-8	2-W23-168	6-81-58
2-E28-21	2-W11-9	2-W23-169	6-83-47
2-E28-3	2-W12-1	2-W23-170	6-84-33
2-E28-4	2-W14-1	2-W23-171	6-84-35A
2-E28-5	2-W14-2	2-W23-172	6-86-60
2-E28-51	2-W14-3	2-W23-173	6-87-55
2-E28-52	2-W14-4	2-W23-174	6-89-35
2-E28-53	2-W14-5	2-W23-175	6-9-E2
2-E28-54	2-W14-53	2-W23-176	6-92-14
2-E28-55	2-W14-6	2-W23-178	6-93-93
2-E28-56	2-W14-7	2-W23-179	6-96-49
2-E28-57	2-W14-8	2-W23-180	6-97-43
2-E28-58	2-W14-8A	2-W23-181	6-HAN-1
2-E28-59	2-W15-1	2-W23-182	6-HAN-10
2-E28-6	2-W15-10	2-W23-183	6-HAN-11
2-E28-60	2-W15-100	2-W23-184	6-HAN-12
2-E28-61	2-W15-101	2-W23-185	6-HAN-13
2-E28-62	2-W15-102	2-W23-186	6-HAN-14
2-E28-63	2-W15-11	2-W23-187	6-HAN-16
2-E28-64	2-W15-12	2-W23-188	6-HAN-17
2-E28-65	2-W15-121	2-W23-189	6-HAN-18
2-E28-66	2-W15-122	2-W23-190	6-HAN-2
2-E28-69	2-W15-123	2-W23-191	6-HAN-20
2-E28-7	2-W15-124	2-W23-192	6-HAN-21
2-E28-70	2-W15-126	2-W23-193	6-HAN-22
2-E28-71	2-W15-127	2-W23-194	6-HAN-23
2-E28-8	2-W15-128	2-W23-195	6-HAN-25
2-E28-9	2-W15-129	2-W23-196	6-HAN-26
2-E29-1	2-W15-13	2-W23-197	6-HAN-3
2-E32-1	2-W15-130	2-W23-198	6-HAN-31
2-E33-1	2-W15-131	2-W23-199	6-HAN-4

TABLE B.1. (contd)

2-E33-10	2-W15-132	2-W23-2	6-HAN-7
2-E33-100	2-W15-133	2-W23-200	6-S11E12A
2-E33-101	2-W15-134	2-W23-201	6-S11E12B
2-E33-102	2-W15-135	2-W23-202	6-S12-29
2-E33-103	2-W15-136	2-W23-203	6-S12-3
2-E33-104	2-W15-137	2-W23-204	6-S14-20
2-E33-105	2-W15-138	2-W23-205	6-S18-E2A
2-E33-106	2-W15-139	2-W23-206	6-S19-11
2-E33-107	2-W15-14	2-W23-207	6-S19-E13
2-E33-108	2-W15-140	2-W23-208	6-S23-26
2-E33-109	2-W15-141	2-W23-210	6-S24-19
2-E33-110	2-W15-142	2-W23-212	6-S27-E14
2-E33-111	2-W15-143	2-W23-225	6-S29-E12
2-E33-112	2-W15-144	2-W23-226	6-S3-25
2-E33-113	2-W15-145	2-W23-227	6-S3-E12
2-E33-114	2-W15-146	2-W23-3	6-S30E15A
2-E33-115	2-W15-147	2-W23-4	6-S30E15C
2-E33-116	2-W15-148	2-W23-5	6-S31-1
2-E33-117	2-W15-149	2-W23-51	6-S4-E16
2-E33-118	2-W15-153	2-W23-52	6-S51-2
2-E33-119	2-W15-154	2-W23-53	6-S6-E4A
2-E33-12	2-W15-155	2-W23-54	6-S6-E4B
2-E33-120	2-W15-156	2-W23-55	6-S6-E4C
2-E33-121	2-W15-158	2-W23-56	6-S6-E4D
2-E33-122	2-W15-159	2-W23-57	6-S6-E4E
2-E33-123	2-W15-160	2-W23-58	6-S6-E4F
2-E33-124	2-W15-161	2-W23-59	6-S6-E4G
2-E33-125	2-W15-162	2-W23-6	6-S6-E4H
2-E33-126	2-W15-163	2-W23-60	6-S6-E4J
2-E33-127	2-W15-164	2-W23-61	6-S6E14A
2-E33-128	2-W15-165	2-W23-62	6-S6E14B
2-E33-129	2-W15-166	2-W23-64	6-S6E15A
2-E33-13	2-W15-167	2-W23-65	6-S6E15B
2-E33-130	2-W15-168	2-W23-66	6-S7-34
2-E33-131	2-W15-169	2-W23-67	6-S8-19
			6-S9-63

APPENDIX C

EXPLANATION OF ERROR MESSAGES

APPENDIX C

EXPLANATION OF ERROR MESSAGES

<u>Message</u>	<u>Explanation</u>
Bad Well Name	The name of the well is invalid. See Table B.1.
Depth Increment too large	The depth increment is so large that the second point to be interpolated falls outside the range of values stored for this log.
Error in Deepest Point	This is a non-fatal system check. The person responsible for program DIGLOG1 should be called in.
Error in High Log Datum	This is a non-fatal system check. The person responsible for program DIGLOG1 should be called in.
Error in Low Log Datum	This is a non-fatal system check. The person responsible for program DIGLOG1 should be called in.
Error Input Starting Depth	This is a non-fatal system check. The person responsible for program DIGLOG1 should be called in.
Error is Base 36 Calculation	This is system error. The person responsible for the system should be called in.
Internal Check Error	This is a system error. The person responsible for the system should be called in.
Internal Check Warning on Data Pairs	This is a system message. Call in the person responsible for the system.
Invalid Area, Try Again	The user must enter a 100, 200, 300, 1100, or 3000; all other values are invalid.
Invalid Depth Increment	The depth increment must be a value greater than 0.
Invalid Log type	This is a system error. The person who is responsible for the system should be called in.

Invalid Number	The number entered is outside the range of acceptable values.
Invalid Order	Values must be entered in increasing value.
Invalid Starting Depth	The sampling process must start with the range of stored values for this log.
No Match Found for Well	All the well names are kept on a master file (Table B.1). If a new well is added to the system, it must find its match on this master file. If it does not, the name is flagged as wrong and the user is given the chance to enter the correct name.
Outside of Range (High Side)	The user attempted to edit out nonexistent points at a depth below the well log. The value given with the message is the largest depth this log reaches.
Outside of Range (Low Side)	The user attempted to edit out nonexistent points above where the log starts. The value given with the message is the depth where the log starts.
Storage Overflow	The program does not have enough room to store all the points for this well. The person responsible for the system should be called in.
The Name is Too Long	The name of the well cannot contain more than 12 characters.
Too Many Replicates of Well _____ on File	Too many replicates of this well are stored out on the disk file. You will have to store some of the replicates out on tape to make room.
Well Name Too Long	This is a system message. Call the person responsible for the system.
Well Not Found	This well has not yet been added to the system.
Would Remove All Data	The user attempted to remove all the data associated with this well.

APPENDIX D

SYSTEM SOFTWARE

APPENDIX D

SYSTEM SOFTWARE

Part of the intent of the well logging system was to allow the user access to 'off the shelf' analytical software. A brief description is given of some of the software currently available on the PNL VAX systems.

ALDS is a system of software modules developed in conjunction with the Analysis of Large Data Sets research program. Features of the system include a menu-driven command language interpreter, an interactive data editor (ADE), a graphics-enhanced version of the Minitab statistical package, and support of SDB (Self Describing Binary files).

CIRMIS is a data base system for processing ground-water data, other than well logs, for the Hanford Site. Whereas the system itself will not be used, 'hooks' have been built into the well logging system to allow information be extracted from the CIRMIS data base for use in the analysis of well logs.

GLIM is a statistical package that provides a framework for the fitting of generalized linear models to data. GLIM can be used as a calculator, operating on vectors and scalars with general arithmetic expressions including many standard functions. Results may be displayed either graphically, in scatter diagrams or by lists. Data may be entered at the terminal or read from another previously prepared file; the numbers are in fixed or free format. GLIM can be used for data exploration, for example, by trying various transformations to produce approximate linearity of relationship between two variates, to subdivide the data and to average over the subsets, to check data visually for the presence of outliers and to calculate simple statistics. In addition, GLIM has a facility to do iterative weighted-least squares, which can be used to program any general model or other regression procedures that can be stored as macros for future use.

MAKGRAPH is a software package developed by C. R. Cole of PNL to produce classical two dimensional graphs. Its usefulness is evidenced by a wide variety of options available to the user in one package, the ease in which

these options can be invoked, and the ease in which the software itself can be changed to meet special needs of the user.

MINITAB is an easy-to-use, general purpose statistical computing system. It is a flexible and powerful tool that has been designed specifically for researchers who have had no previous experience with computers. MINITAB is designed to allow users to 'speak' to the computer in commands similar to English sentences.

RS/1 is an integrated software system designed specifically to meet the unique data management requirements of the research scientist. Capabilities include the ability to make tables, graphs and to do statistical analysis with either an English-like command language or a built-in, PL/1-like programming language.

RUMMAGE is designed to perform analyses of complex linear models for non-orthogonal data. Possible analyses include multiple regression, analysis of variance, analysis of covariance, and the analysis of categorical data. The RUMMAGE command structure uses easy-to-learn, English-like phrases and words and is very similar to MINITAB.

SPSS (Statistical Package for the Social Sciences) is an integrated system of computer programs designed for the analysis of social comprehensive packages. It enables the user to perform many different types of data analysis in a simple and convenient manner. SPSS allows a great deal of flexibility in the format of data. It provides the user with a comprehensive set of procedures for data transformation and file manipulation, and it offers the researcher a large number of statistical routines commonly used in the social sciences.

APPENDIX E

PROGRAM LISTING FOR PROGRAM DIGLOG1

```
-----
00001 C PROGRAM DIGLOG
00002 C FUNCTION: GENERALIZED DIGITIZING PROGRAM FOR
00003 C WELL LOGS AND GRAPHS
00004 C
00005 C AUTHOR: S.C.BLAIR, OCT.1980
00006 C FLO FILE 'SCRATCH' CONTAINS GEOMETRY OF CHART
00007 C FLO FILE 'LOGDATA' CONTAINS CALIBRATED VALUES
00008 C
00009 C BINARY OUTPUT FILES ARE OF TWO TYPES:
00010 C (1)- A FILE FOR GEOPHYSICAL WELL LOGS
00011 C WITH A HEADER CONTAINING WELDES,
00012 C LGTYP,DATE,AND OTHER FILE DESCRIPTORS.
00013 C (2) - A FILE FOR GENERAL CHARTS AND GRAPHS.
00014 C THIS FILE CONTAINS ONLY THE FILE
00015 C DESCRIPTORS (IE NBLK,NLAST,DSTRT,DMAX
00016 C YMN1,YMX1)
00017 C
00018 C DEFINITION OF VARIABLES:
00019 C NBLK=NO. OF BLOCKS IN A FILE-Q FILE
00020 C EACH BLOCK CONTAINS 64 XY PAIRS
00021 C EXCEPT THE LAST BLOCK, NBLK IS ALSO THE NUMBER
00022 C OF BINARY LINES CONTAINING XY PAIRS WRITTEN TO
00023 C OUTPUT,
00024 C NLAST=NO. OF POINTS CONTAINED IN THE LAST BLOCK OR LINE
00025 C OF A FILE,
00026 C DSTRT=STARTING DEPTH OF A WELL LOG OR STARTING VALUE OF
00027 C A CHART (X AXIS),
00028 C DMAX=MAX. DEPTH OR MAX. X VALUE OF A CHART OR LOG,
00029 C YMN1 =MIN. Y VALUE
00030 C YMX1 =MAX. Y VALUE
00031 C YMN AND YMX ARE MIN AND MAX Y VALUES IN DIGITIZER COORDS.
00032 C TLNTH=TOTAL LENGTH OF DIGITIZED CHART IN INCHES.
00033 C
00034 C WELDES = WELL DESIGNATION, IT SHOULD BE THE SAME AS IS
00035 C USED FOR THE CIRMIS SYSTEM,
00036 C ISW = SWITCH FOR VS-60 DISPLAY (1=OFF,2=ON)
00037 C
00038 C
00039 C *****
00040 C BYTE WELDES(12),TCHN(28),DATE(8),IA,LGTYP(3)
00041 C BYTE FNAME(28)
00042 C REAL BUFF(2,64)
00043 C INTEGER MAIN(50),MENU(100),DSPY(3800)
00044 C DIMENSION HDBUF1(20),HDBUF2(20)
00045 C COMMON/BK1/CT,ST
00046 C IHTN=0
00047 C NPTS=0
00048 C NBLK=0
00049 C STRT=0.
00050 C NSECT=0
00051 C TLNTH=0.0
00052 C ZBLK=0.
00053 C YMN=0.
```

```

00054      YMX=0.
00055      ISW=0.
00056      HDSIZ=40.
00057  C      INITIALIZE DIGITIZER, VS-60
00058      CALL BXINIT(2,1)
00059      CALL MAINDF(MAIN,50,1,1)
00060      CALL CLKOFF
00061      CALL DPFILE(3,'SCRATCH',500.,NE,2)
00062      CALL DPFILE(3,'LOGDATA',500.,NC,2)
00063  C
00064  C      DISPLAY MENU
00065      CALL SUBDF(MENU,100,0)
00066      CALL AREA(1)
00067      CALL LPHIT(1)
00068      CALL APOS(1,900)
00069      CALL NAME(1)
00070      CALL TEXT('DIGITIZE SECTION',-40)
00071      CALL NAME(2)
00072      CALL TEXT('CALIBRATE LOG',-40)
00073      CALL NAME(3)
00074      CALL TEXT('DISPLAY CHART',-40)
00075      CALL NAME(4)
00076      CALL TEXT('RESTART',0)
00077      CALL APOS(1,200)
00078      CALL NAME(10)
00079      CALL TEXT('FILE OUTPUT',0)
00080      CALL APOS(1,100)
00081      CALL NAME(40)
00082      CALL TEXT('STOP',0)
00083      CALL ENDSB
00084  C
00085  C      SET PRINT OPTION
00086      WRITE(6,*)'SET PRINT OPTION (0 - 5)'
00087      WRITE(6,*)'ENTER ZERO FOR NO PRINTER OUTPUT'
00088      READ(6,107)NPRINT
00089      WRITE(6,*)'SET DIGITIZING INCREMENT (IN)'
00090      READ(6,104)DELTA
00091  C
00092  C
00093      WRITE(6,*)'*****'
00094      WRITE(6,*)'*          TO OPERATE THIS PROGRAM          *'
00095      WRITE(6,*)'* (1) - DIGITIZE A SECTION OF A CHART          *'
00096      WRITE(6,*)'* (2) - DISPLAY CHART TO CHECK FOR ERRORS          *'
00097      WRITE(6,*)'* (3) - CALIBRATE THE LOG OR CHART          *'
00098      WRITE(6,*)'* (4) - FILE OUTPUT WRITES A CALIBRATED DATA          *'
00099      WRITE(6,*)'*          TO A BINARY FILE,          *'
00100      WRITE(6,*)'*****'
00101  C      GET LIGHT PEN INTERRUPT
00102  9991 IEND=0
00103      WHILE(IEND.EQ.0)
00104      . CALL GRATN(0,ITYP,'LP')
00105      . IF(ITYP.EQ.'LP')
00106      . . CALL LTPEN(NAM,ITIP,IXP,IYP)
00107      . . IF(ITIP.EQ.1)
00108      . . . SELECT(NAM)
00109      . . . (1)

```

```

00110      . . . . . WRITE(6,*)'***** DIGITIZE SECTION *****'
00111      . . . . . WRITE(6,*)'(1) - START NEW LOG'
00112      . . . . . WRITE(6,*)'(2) - CONTINUE SAME LOG'
00113      . . . . . WRITE(6,*)'(3) - ESCAPE'
00114      . . . . . READ(6,107)IAA
00115      . . . . . SELECT(IAA)
00116      . . . . . (1)
00117      . . . . . NBLK=0
00118      . . . . . SIRT=0.
00119      . . . . . NPTS=0
00120      . . . . . NSECT=0
00121      . . . . . TLNTH=0.
00122      . . . . . YMN=0.
00123      . . . . . YMX=0.
00124      . . . . . ISW=0.
00125      . . . . . ...FIN
00126      . . . . . (2)
00127      C . . . . .
00128      C . . . . . READ SCRATCH FILE AND DISPLAY CHART
00129      . . . . . DISPLAY=CHART
00130      . . . . . STRT=(NBLK*1.0)-1.0
00131      . . . . . NBLK=NBLK+1
00132      . . . . . NPTS=NLAST
00133      . . . . . ...FIN
00134      . . . . . (3)
00135      . . . . . GO TO 9991
00136      . . . . . ...FIN
00137      . . . . . ...FIN
00138      . . . . . GET-ROTATION=ANGLE
00139      . . . . . WRITE(6,*)'*****TO DIGITIZE STRING *****'
00140      . . . . . WRITE(6,*)' '
00141      . . . . . WRITE(6,*)'PRESS WHITE BUTTON TO START STRING AND'
00142      . . . . . WRITE(6,*)'PRESS BLUE BUTTON TO END STRING'
00143      . . . . . DIGITIZE-POINTS
00144      C . . . . .
00145      C . . . . .
00146      . . . . . NAM=0
00147      . . . . . ITYP=' '
00148      . . . . . NSECT=NSECT+1
00149      . . . . . ...FIN
00150      . . . . . (2)
00151      . . . . . REPEAT UNTIL(ICALB.LE.2)
00152      . . . . . WRITE(6,*)' '
00153      . . . . . WRITE(6,*)'***** TO CALIBRATE CHART *****'
00154      . . . . . WRITE(6,*)' (1) - DO CALIBRATION'
00155      . . . . . WRITE(6,*)' (2) - ESCAPE'
00156      . . . . . READ(6,*)ICALB
00157      . . . . . ...FIN
00158      . . . . . IF(ICALB.EQ.2)GO TO 9991
00159      C . . . . .
00160      C . . . . .
00161      . . . . . CALIBRATE=XY-AXES
00162      . . . . . CALIBRATE=SCRATCH-FILE
00163      C . . . . .
00164      C . . . . .
00165      . . . . . ...FIN

```

```

00166      . . . . (3)
00167      . . . . . DISPLAY=CHART
00168      C . . . . .
00169      C . . . . .
00170      . . . . . ...FIN
00171      . . . . . (4)
00172      C . . . . . RESTART PROGRAM
00173      C . . . . . RESET SCRATCH FILE
00174      . . . . . CALL DPINQ(3,'SCRATCH',ISTAT,SZ)
00175      . . . . . IF(ISTAT,EQ,-1)
00176      . . . . . . WRITE(6,*)'FLQ FILE DOES NOT EXIST'
00177      . . . . . . GO TO 9991
00178      . . . . . ...FIN
00179      . . . . . DISPLAY=CHART
00180      . . . . . STRT=(NBLK*1.0)-1.0
00181      . . . . . NBLK=NBLK-1
00182      . . . . . NPTS=NLAST
00183      . . . . . ...FIN
00184      . . . . . (10)
00185      . . . . . WRITE(6,*)'***** FILE OUTPUT *****'
00186      . . . . . REPEAT UNTIL(ICHRT,LE.3)
00187      . . . . . . WRITE(6,*)'TYPE OF CHART IS:'
00188      . . . . . . WRITE(6,*)'(1) - BOREHOLE GEOPHYSICS'
00189      . . . . . . WRITE(6,*)'(2) - OTHER'
00190      . . . . . . WRITE(6,*)'(3) - ESCAPE'
00191      . . . . . . READ(6,107)ICHRT
00192      . . . . . ...FIN
00193      C . . . . .
00194      C . . . . . SELECT(ICHRT)
00195      . . . . . . (1)
00196      . . . . . . . GET=WELL-DATA
00197      . . . . . . . OPEN=BIN-FILE
00198      . . . . . . . WRITE=BIN-FILE
00199      . . . . . . ...FIN
00200      C . . . . .
00201      . . . . . . (2)
00202      . . . . . . . OPEN=BIN-FILE
00203      C . . . . . .
00204      . . . . . . . CALL DPINQ(3,'LOGDATA',ISTAT,SZ)
00205      . . . . . . . IF(ISTAT,EQ,-1)
00206      . . . . . . . . WRITE(6,*)'FILE LOGDATA DOES NOT EXIST'
00207      . . . . . . . . GO TO 9991
00208      . . . . . . . ...FIN
00209      . . . . . . . WRITE=BIN-FILE
00210      . . . . . . . ...FIN
00211      . . . . . . (3)
00212      . . . . . . . GO TO 9991
00213      . . . . . . . ...FIN
00214      . . . . . . ...FIN
00215      C . . . . .
00216      C . . . . .
00217      . . . . . . NAM=0
00218      . . . . . . . ITYP=' '
00219      . . . . . . ...FIN
00220      . . . . . (40)
00221      . . . . . . IEND=1

```

```

00222      . . . . .FIN
00223      . . . . .FIN
00224      . . . . .FIN
00225      . . . . .FIN
00226      . . . . .FIN
00227      102  FORMAT(2X,5HIPTS=,1X,I5,2X,5HNBK=,1X,I4,1X,6HNLAST=,1X,I4)
00228      103  FORMAT(1X,6HBLOCK=,1X,I4)
00229      104  FORMAT(F10,3)
00230      105  FORMAT(28A1)
00231      106  FORMAT(2F10,3)
00232      107  FORMAT(I5)
00233      108  FORMAT(2X,6HTLNTH=,F10,3)
00234      109  FORMAT(A2,I1)
00235      110  FORMAT(I4,I4,F7.2,F7.2,F7.2)
00236      111  FORMAT(28O4)
00237      112  FORMAT(I4,I4,F7.2,F7.2,F7.2,F7.2)
00238      113  FORMAT(4X,'WELDES',4X,'TYPE',2X,'DATE',5X,'NBK',1X,'NLAST',
00239      1 3X,'YMN',8X,'YMX',7X,'DSTRT',7X,'DMAX')
00240      114  FORMAT(1X,I2A1,3X,3A1,2X,8A1,1X,I3,3X,I2,2X,F10.3,1X,F10.3,
00241      1 1X,F10.3,1X,F10.3)
00242      STOP

```

```

-----
00243      TO DIGITIZE-POINTS
00244      . IMODE=0
00245      1  . CONTINUE
00246      . IBTN=0
00247      . CALL BX2XY(X1,Y1,IBTN)
00248      . SELECT(IBTN)
00249      . (2)
00250      C *****
00251      C . . . MUST STORE FIRST POINT OF EACH SECTION
00252      C . . . IN ORDER TO MAKE THE DISTANCE CALCULATION
00253      C *****
00254      . . . CALL BXLON
00255      . . . CALL BXPEEP
00256      C . . . SAVE FIRST POINT OF EACH SECTION IN ROTATED COORDS
00257      . . . X1ST=X1
00258      . . . Y1ST=Y1
00259      . . . CALL ROTPT(X1ST,Y1ST)
00260      . . . XFST=X1ST
00261      C . . . STORE VERY FIRST POINT
00262      . . . IF(NSECT.EQ.0)X1A=XFST
00263      . . . STORE-POINT
00264      . . . CALL BXLOFF
00265      . . . IMODE=1
00266      C . . . RESET 1ST POINT FOR DISTANCE CALCULATION
00267      . . . X1=XL
00268      . . . Y1=YL
00269      . . . FIN
00270      . . . (3)
00271      . . . CALL BXLON
00272      . . . CALL BXPEEP
00273      . . . STORE-POINT
00274      . . . TLNTH=X1

```

```

00275      . . . CALL BXLOFF
00276      . . . IMODE=2
00277      . . . FIN
00278      . . . (4)
00279      . . . CALL BXLOFF
00280      . . . IMODE=2
00281      . . . FIN
00282      . . . FIN
00283      C      .
00284      . SELECT(IMODE)
00285      . . (1)
00286      C      . . COMPUTE DISTANCE TO POINT AND STORE POINT
00287      C      . . IF DISTANCE IS GE DELTA,
00288      . . . D=SQRT((X1-XL)**2+(Y1-YL)**2)
00289      . . . IF(D,GE,DELTA)
00290      . . . . CALL BXLON
00291      . . . . STORE-POINT
00292      . . . . CALL BXLOFF
00293      . . . . FIN
00294      . . . FIN
00295      . . (2)
00296      . . . WRITE-BUFFER
00297      . . . IPTS=(NBLK-1)*64+NLAST
00298      C      . .
00299      . . . IF(NPRINT,GE,2)
00300      . . . . IFLG=1
00301      . . . . PRINT-HEADER
00302      . . . . FIN
00303      C      . .
00304      . . . GO TO 999
00305      . . . FIN
00306      . . . FIN
00307      . GO TO 1
00308      999 . CONTINUE
00309      . . . FIN

```

```

-----
00310      TO STORE-POINT
00311      . NPTS=NPTS+1
00312      . XL=X1
00313      . YL=Y1
00314      . CALL ROTPT(X1,Y1)
00315      C      . *****
00316      C      . TRANSLATE AXES WHERE X1=TOTAL LENGTH OF CHART
00317      C      . AND Y1=DISTANCE FROM CALIB. LINE.
00318      C      . *****
00319      . X1=(X1-XFST)+TLNTH
00320      . Y1=Y1-YP1
00321      C      .
00322      . BUFF(1,NPTS)=X1
00323      . BUFF(2,NPTS)=Y1
00324      . YMX=MAX(YMX,Y1)
00325      . YMN=MIN(YMN,Y1)
00326      C      .
00327      . IF(NPRINT,GE,4)

```

```

00328      . . IF(NPTS,EQ,1)
00329      . . . WRITE(5,106)XL,YL
00330      . . . WRITE(5,106)X1,Y1
00331      . . ...FIN
00332 C      . .
00333      . . ...FIN
00334      . IF(NPTS,GE,64)WRITE-BUFFER
00335      . ...FIN

```

```

00336      TO WRITE-BUFFER
00337 C      . WRITE POINTS IN BUFFER
00338      . WORDS=NPTS*4
00339      . STRT=STRT+1.
00340      . NBLK=NBLK+1
00341      . NLAST=NPTS
00342      . CALL DPW(NE,STRT,BUFF,WORDS)
00343 C      .
00344      . IF(NPRINT,GE,4)
00345      . . WRITE(5,*)'YMN , YMX'
00346      . . WRITE(5,106)YMN,YMX
00347      . ...FIN
00348 C      . *****
00349 C      . UPDATE HEADER BLOCK
00350      . HDBUF1(1)=NBLK
00351      . HDBUF1(2)=NLAST
00352      . HDBUF1(3)=TLNTH
00353      . HDBUF1(4)=YMN
00354      . HDBUF1(5)=YMX
00355      . CALL DPW(NE,ZBLK,HDBUF1,HDSIZ)
00356 C      . *****
00357      . IF(NPRINT,GE,2)
00358      . . IFLG=1
00359      . . PRINT-HEADER
00360      . ...FIN
00361 C      .
00362      . NPTS=0
00363      . ...FIN

```

```

00364      TO WRITE-BIN-FILE
00365      . STRT=0.
00366      . NPTS=64
00367      . WORDS=256.
00368 C      . READ HEADER
00369      . IFLG=2
00370      . READ-FLQ-HEADER
00371 C      .
00372      . IF(NPRINT,GE,2)
00373      . . IFLG=2
00374      . . PRINT-HEADER
00375      . ...FIN
00376 C      .
00377 C      .

```



```

00378      . SELECT(ICHRT)
00379      . . (1)
00380      . . . WRITE(4)WELDES,LGTYP,DATE,NBLK,NLAST,YMN1,YMX1,
00381 1. . . . DSTRT,DMAX
00382      . . . IF(NPRINT,GE,2)
00383      . . . . WRITE(5,113)
00384      . . . . WRITE(5,114)WELDES,LGTYP,DATE,NBLK,NLAST,YMN1,YMX1,DSTRT,DMAX
00385      . . . . ...FIN
00386      . . . ...FIN
00387      . . (2)
00388      . . . WRITE(4)NBLK,NLAST,YMN1,YMX1,DSTRT,DMAX
00389      . . . ...FIN
00390      . . . ...FIN
00391 C      . READ AND WRITE DATA
00392      . DO(I=1,NBLK)
00393      . . STRT=STRT+1.
00394 C      . . RESET POINTERS FOR LAST BLOCK
00395      . . IF(I,EQ,NBLK)
00396      . . . NPTS=NLAST
00397      . . . WORDS=NLAST*4.
00398      . . . ...FIN
00399      . . CALL DPR(NC,STRT,BUFF,WORDS)
00400      . . WRITE(4)((BUFF(J,K),J=1,2),K=1,NPTS)
00401 C      . .
00402      . . IF(NPRINT,GE,3)
00403      . . . WRITE(5,*)'CALIBRATED VALUES STARTING BLOCK'
00404      . . . WRITE(5,107)I
00405      . . . WRITE(5,106)BUFF(1,1),BUFF(2,1)
00406      . . . ...FIN
00407 C      . .
00408      . . . ...FIN
00409      . CLOSE(UNIT=4)
00410      . ...FIN

```

```

00411      TO OPEN-BIN-FILE
00412      . IOD=0
00413      . REPEAT UNTIL(IOD,EQ,1,OR,IOD,EQ,2)
00414 811 . . CONTINUE
00415      . . WRITE(6,*)'OUTPUT FILE'
00416      . . WRITE(6,*)'(1) = DB0'
00417      . . WRITE(6,*)'(2) = DK2'
00418      . . READ(6,107,ERR=811)IOD
00419      . ...FIN
00420      . SELECT(IOD)
00421      . . (1)
00422      . . . GET-FILE-NAME
00423      . . . WRITE(5,105)FNAME
00424 C      . . .
00425 C      . . .
00426      . . . CALL ASNLUN(4,'DB',0)
00427      . . . OPEN(UNIT=4,NAME=FNAME,TYPE='NEW',FORM='UNFORMATTED')
00428      . . . ...FIN
00429      . . (2)
00430      . . . GET-FILE-NAME

```

```

00431      . . . WRITE(5,105)FNAME
00432      . . . WRITE(5,111)FNAME
00433      C      . . .
00434      C      . . .
00435      . . . CALL ASNLUN(4,'DK',2)
00436      . . . OPEN(UNIT=4,NAME=FNAME,TYPE='NEW',FORM='UNFORMATTED')
00437      . . . FIN
00438      . . . FIN
00439      . . . FIN

```

```

-----
00440      TO GET=WELL=DATA
00441      555      . WRITE(6,*)'***** OUTPUT FILE *****'
00442      . WRITE(6,*)' '
00443      . WRITE(6,*)' ENTER WELL DESIGNATION (12A1)'
00444      . READ(6,105)WELDES
00445      C      .
00446      . DO(NN=1,12)
00447      . . IF(WELDES(NN).EQ.'=')
00448      . . . WRITE(6,*)'CANNOT HAVE (=) IN NAME'
00449      . . . WRITE(6,*)'TRY AGAIN'
00450      . . . GO TO 555
00451      . . . FIN
00452      . . . FIN
00453      C      .
00454      . REPEAT UNTIL(IDTYP.LE.11)
00455      . . WRITE(6,*)' ENTER TYPE OF LOG'
00456      . . WRITE(6,*)' '
00457      . . WRITE(6,*)'(1) - GAMMA GAMMA (DENSITY)'
00458      . . WRITE(6,*)'(2) - NATURAL GAMMA'
00459      . . WRITE(6,*)'(3) - NEUTRON EPITHERMAL NEUTRON'
00460      . . WRITE(6,*)'(4) - CALIPER'
00461      . . WRITE(6,*)'(5) - SONIC'
00462      . . WRITE(6,*)'(6) - SP & R'
00463      . . WRITE(6,*)'(7) - NEUTRON GAMMA'
00464      . . WRITE(6,*)'(8) - SP'
00465      . . WRITE(6,*)'(9) - TEMP'
00466      . . WRITE(6,*)'(10) - FLOWMETER'
00467      . . WRITE(6,*)'(11) - OTHER'
00468      . . READ(6,107)IDTYP
00469      . . . FIN
00470      C      .
00471      . DENOTE=TYPE
00472      C      .
00473      . WRITE(6,*)'ENTER DATE LOGGED (MO=DAY=YR)'
00474      . READ(6,105)DATE
00475      C      .
00476      . WRITE(6,*)'LOG DIGITIZED BY'
00477      . READ(6,105)TCHN
00478      . . . FIN

```

```

-----
00479      TO DENOTE=TYPE
00480      . SELECT(IDTYP)

```

```
00481      . . (1)
00482      . . . LGTYP(1)='D'
00483      . . . LGTYP(2)='E'
00484      . . . LGTYP(3)='N'
00485      . . ...FIN
00486      . . (2)
00487      . . . LGTYP(1)='N'
00488      . . . LGTYP(2)='A'
00489      . . . LGTYP(3)='G'
00490      . . ...FIN
00491      . . (3)
00492      . . . LGTYP(1)='N'
00493      . . . LGTYP(2)='E'
00494      . . . LGTYP(3)='N'
00495      . . ...FIN
00496      . . (4)
00497      . . . LGTYP(1)='C'
00498      . . . LGTYP(2)='A'
00499      . . . LGTYP(3)='L'
00500      . . ...FIN
00501      . . (5)
00502      . . . LGTYP(1)='S'
00503      . . . LGTYP(2)='O'
00504      . . . LGTYP(3)='N'
00505      . . ...FIN
00506      . . (6)
00507      . . . LGTYP(1)='S'
00508      . . . LGTYP(2)='P'
00509      . . . LGTYP(3)='R'
00510      . . ...FIN
00511      . . (7)
00512      . . . LGTYP(1)='N'
00513      . . . LGTYP(2)='E'
00514      . . . LGTYP(3)='G'
00515      . . ...FIN
00516      . . (8)
00517      . . . LGTYP(1)='S'
00518      . . . LGTYP(2)='P'
00519      . . . LGTYP(3)='O'
00520      . . ...FIN
00521      . . (9)
00522      . . . LGTYP(1)='T'
00523      . . . LGTYP(2)='M'
00524      . . . LGTYP(3)='P'
00525      . . ...FIN
00526      . . (10)
00527      . . . LGTYP(1)='F'
00528      . . . LGTYP(2)='L'
00529      . . . LGTYP(3)='M'
00530      . . ...FIN
00531      . . (11)
00532      . . . WRITE(6,*)'ENTER THREE LETTER CODE FOR LOG TYPE'
00533      . . . READ(6,105)LGTYP
00534      . . ...FIN
00535      . ...FIN
00536      ...FIN
```

```
-----
00537      TO PRINT-HEADER
00538      . SELECT(IFLQ)
00539      . . (1)
00540      . . . WRITE(5,*)'FLQ HEADER DATA (SCRATCHCH)'
00541      . . . WRITE(5,*)'NBLK,NLAST,TLNTH,YMN,YMX'
00542      . . . WRITE(5,110)NBLK,NLAST,TLNTH,YMN,YMX
00543      . . ...FIN
00544      . . (2)
00545      . . . WRITE(5,*)'FLQ HEADER DATA (LOGDATA)'
00546      . . . WRITE(5,*)'NBLK,NLAST,YMN1,YMX1,DSTRT,DMAX'
00547      . . . WRITE(5,112)NBLK,NLAST,YMN1,YMX1,DSTRT,DMAX
00548      . . ...FIN
00549      . ...FIN
00550      ...FIN
-----
```

```
-----
00551      TO GET-FILE-NAME
00552      . WRITE(6,*)'FILE NAME (28A1)'
00553      . READ(6,105)FNAME
00554      . N=ICHR(FNAME)
00555      ...FIN
-----
```

```
-----
00556      TO GET-ROTATION-ANGLE
00557      . WRITE(6,*)' '
00558      . WRITE(6,*)'*****'
00559      . WRITE(6,*)'DIGITIZE TWO POINTS ON DEPTH AXIS'
00560      . WRITE(6,*)'USE BLUE BUTTON'
00561      . GET-CALIBRATION-POINT
00562      . XP1=XP
00563      . YP1=YP
00564      . GET-CALIBRATION-POINT
00565      . XP2=XP
00566      . YP2=YP
00567      . DX=(XP2-XP1)
00568      . DY=(YP2-YP1)
00569      . D=SQRT(DX*DX+DY*DY)
00570      . ST=DY/D
00571      . CT=DX/D
00572      C . ROTATE CALIBRATION POINTS
00573      . CALL ROTPT(XP1,YP1)
00574      . CALL ROTPT(XP2,YP2)
00575      C .
00576      . IF(NPRINT,GE,5)
00577      . . WRITE(5,*)'COS AND SIN'
00578      . . WRITE(5,106)CT,ST
00579      . ...FIN
00580      ...FIN
-----
```

```
00581      TO GET-CALIBRATION-POINT
00582      . CALL BX1XY(XP,YP,IBTN)
00583      . CALL RXLON
00584      . CALL BXBEEP
00585      . CALL BXLOFF
00586      ...FIN
```

```
-----
00587      TO CALIBRATE-XY-AXES
00588      C      . DIGITIZE CALIBRATION POINTS FOR X&Y AXES
00589      C      . X = DEPTH, Y = PARAMETER VALUE
00590      C      .
00591      . GET-ROTATION-ANGLE
00592      . DX=XP2-XP1
00593      . WRITE(6,*)'ENTER DEPTHS (X VALUES) OF THESE COORDS (AMIN,AMAX)'
00594      . WRITE(6,*)'INCLUDE DECIMAL POINT'
00595      . READ(6,106)AMIN,AMAX
00596      . DELD=AMAX-AMIN
00597      . DINC=DELD/DX
00598      . WRITE(6,*)'ENTER STARTING DEPTH FOR RECORD'
00599      . READ(6,104)DSTRT
00600      C      .
00601      . WRITE(6,*)'DIGITIZE TWO POINTS ON Y AXIS (AMIN,AMAX)'
00602      . GET-CALIBRATION-POINT
00603      . CALL ROTPT(XP,YP)
00604      . YCAL1=YP-YP1
00605      . GET-CALIBRATION-POINT
00606      . CALL ROTPT(XP,YP)
00607      . YCAL2=YP-YP1
00608      . DY=YCAL2-YCAL1
00609      . WRITE(6,*)'ENTER VALUES FOR THESE POINTS (AMIN,AMAX)'
00610      . WRITE(6,*)'INCLUDE DECIMAL POINT'
00611      . READ(6,106)YMNC,YMNC
00612      . DYV=YMNC-YMNC
00613      . YINC=DYV/DY
00614      C      .
00615      . IF(NPRINT,GE,4)
00616      . . WRITE(5,*)'INCREMENTS ALONG X&Y'
00617      . . WRITE(5,106)DINC,YINC
00618      . ...FIN
00619      C      .
00620      ...FIN
-----
```

```
00621      TO CALIBRATE-SCRATCH-FILE
00622      . NPTS=64
00623      . WORDS=256,
00624      C      .
00625      . IFLQ=1
00626      . READ=FLQ-HEADER
00627      C      .
00628      . IF(NPRINT,GE,2)
00629      . . WRITE(5,*)'CALIBRATION LOOP'
00630      . . IFLQ=1
```

```

00631      . . PRINT=HEADER
00632      . ...FIN
00633      C      .
00634      C      . READ AND WRITE DATA
00635      . DO(I=1,NBLK)
00636      . . STRT=STRT+1.
00637      C      . . RESET POINTERS FOR LAST BLOCK
00638      . . IF(I,EQ,NBLK)
00639      . . . NPTS=NLAST
00640      . . . WORDS=NLAST*4.
00641      . . ...FIN
00642      . . CALL DPR(NE,STRT,BUFF,WORDS)
00643      C      .
00644      . . IF(NPRINT,GE,4)
00645      . . . WRITE(5,*)'POINTS BEFORE AND AFTER TRANSLATION'
00646      . . . WRITE(5,106)BUFF(1,1),BUFF(2,1)
00647      . . ...FIN
00648      C      .
00649      . . DO(J=1,NPTS)
00650      . . . BUFF(1,J)=DSTRT+(DINC*(BUFF(1,J)))
00651      . . . BUFF(2,J)=YMNC+(YINC*(BUFF(2,J)-YCAL1))
00652      . . ...FIN
00653      . . DMAX=BUFF(1,NPTS)
00654      . . CALL DPW(NC,STRT,BUFF,WORDS)
00655      C      .
00656      . . IF(NPRINT,GE,4)
00657      . . . WRITE(5,106)BUFF(1,1),BUFF(2,1)
00658      . . ...FIN
00659      C      .
00660      . ...FIN
00661      C      .
00662      C      ***** FILL HDBUF2 *****
00663      . HDBUF2(1)=NBLK
00664      . HDBUF2(2)=NLAST
00665      . HDBUF2(3)=YMNC+(YINC*(YMN-YCAL1))
00666      . HDBUF2(4)=YMXC+(YINC*(YMX-YCAL1))
00667      . HDBUF2(5)=DSTRT
00668      . HDBUF2(6)=DMAX
00669      . CALL DPW(NC,0,,HDBUF2,HDSIZ)
00670      . ...FIN

```

```

00671      TO READ=FLQ-HEADER
00672      . STRT=0.
00673      . IF(IFLQ,EQ,1)
00674      . . CALL DPR(NE,0,,HDBUF1,HDSIZ)
00675      . . NBLK=HDBUF1(1)
00676      . . NLAST=HDBUF1(2)
00677      . . TLNTH=HDBUF1(3)
00678      . . YMN=HDBUF1(4)
00679      . . YMX=HDBUF1(5)
00680      . . ...FIN
00681      . IF(IFLQ,EQ,2)
00682      . . CALL DPR(NC,0,,HDBUF2,HDSIZ)
00683      . . NBLK=HDBUF2(1)

```

```

00684      . . NLAST=HDBUF2(2)
00685      . . YMN1=HDBUF2(3)
00686      . . YMX1=HDBUF2(4)
00687      . . DSTR1=HDBUF2(5)
00688      . . DMAX=HDBUF2(6)
00689      . ...FIN
00690      ...FIN
00691

```

```

-----
00692      TO DISPLAY=CHART
00693      . WORDS=256,
00694      . NPTS=64
00695      . IFLQ=1
00696      . READ-FLQ=HEADER
00697      . IF(NPRINT,GE,2)
00698      . . IFLQ=1
00699      . . PRINT=HEADER
00700      . ...FIN
00701 C *****
00702 C      . TO SCALE PLOT: HINC AND WINC ARE INCREMENTS ALONG
00703 C      . X AND Y AXES. 870 IS WIDTH AND HEIGHT OF WINDOW
00704 C      . USED ON THE VS-60 SCOPE.
00705 C *****
00706      . HIGH=YMX-YMN
00707      . HINC=(870./HIGH)
00708      . WDINC=870./TLNTH
00709 C      .
00710      . IF(NPRINT,GE,4)
00711      . . WRITE(5,*)'WDINC , HINC'
00712      . . WRITE(5,106)WDINC,HINC
00713      . ...FIN
00714 C      .
00715      . CALL SUBDF(DSPY,3800,ISW)
00716      . ISW=2
00717      . CALL APOS(450,20)
00718      . CALL TEXT('X = AXIS')
00719      . CALL APOS(1,500)
00720      . CALL TEXT('Y = AXIS')
00721 C      .
00722 C      . PLOT DATA
00723      . DO(I=1,NBLK)
00724      . . STRT=STRT+1.
00725 C      . . RESET POINTERS FOR LAST BLOCK
00726      . . IF(1,EQ,NBLK)
00727      . . . NPTS=NLAST
00728      . . . WORDS=NLAST*4,0
00729      . . ...FIN
00730      . . CALL DPR(NE,STRT,BUFF,WORDS)
00731 C      .
00732 C      . LOCATE FIRST POINT
00733      . IF(1,EQ,1)
00734      . . . XP=BUFF(1,1)
00735      . . . YP=BUFF(2,1)
00736      . . . IXP=77+(WDINC*XP)

```

```
00737      . . . IYP=77+(HINC*YP)
00738      . . . CALL APOS(IXP,IYP)
00739      . . . IXPL=IXP
00740      . . . IYPL=IYP
00741      . . . ...FIN
00742 C      . . .
00743 C      . . . PLOT BUFFER
00744      . . . DO(J=1,NPTS)
00745      . . . IF(NPRINT,GE.4)
00746      . . . IF(J.EQ.1)
00747      . . . . . WRITE(5,*)'X , Y'
00748      . . . . . WRITE(5,106)BUFF(1,1),BUFF(2,1)
00749      . . . . . ...FIN
00750      . . . ...FIN
00751      . . . IXP=77+(WDINC*(BUFF(1,J)))
00752      . . . IYP=77+(HINC*(BUFF(2,J)))
00753 C      . . .
00754 C *****
00755 C      . . . PLOT ONLY POINTS THAT ARE 5 INCREMENTS APART ON THE
00756 C      . . . SCREEN, PLOTTING TOO MANY POINTS WILL OVERLOAD THE
00757 C      . . . DISPLAY BUFFER (DSPY).
00758 C *****
00759      . . . IF(ABS(IXP-IXPL).GE.5.OR,ABS(IYP-IYPL).GE.5)
00760      . . . . . CALL VEC(IXP,IYP)
00761 C      . . . . . RESET IXPL FOR SPACING ON DISPLAY SCREEN
00762      . . . . . IXPL=IXP
00763      . . . . . IYPL=IYP
00764      . . . . . ...FIN
00765      . . . ...FIN
00766      . . . ...FIN
00767      . . . CALL ENDSB(ISZ)
00768      . . . ...FIN
00769      . . . END
```

PROCEDURE CROSS-REFERENCE TABLE

```
00243 DIGITIZE-POINTS
00143

00556 GET-ROTATION-ANGLE
00138 00591

00587 CALIBRATE-XY-AXES
00161

00621 CALIBRATE-SCRATCH-FILE
00162

00581 GET-CALIBRATION-POINT
00561 00564 00602 00605

00537 PRINT-HEADER
00301 00359 00374 00631 00699
```


00336 WRITE-BUFFER
00296 00334

00440 GET-WELL-DATA
00196

00411 OPEN-BIN-FILE
00197 00202

00479 DENOTE-TYPE
00471

00310 STORE-POINT
00263 00273 00291

00364 WRITE-BIN-FILE
00198 00209

00551 GET-FILE-NAME
00422 00430

00692 DISPLAY-CHART
00129 00167 00179

00671 READ-FLQ-HEADER
00370 00626 00696

(FLECS VERSION 22.46)

00770 SUBROUTINE ROTPT(X,Y)
00771 C ROTATES POINTS BY AN ANGLE THETA
00772 COMMON/BK1/CT,ST
00773 XR=X*CT+Y*ST
00774 YR=Y*CT-X*ST
00775 X=XR
00776 Y=YR
00777 RETURN
00778 END

(FLECS VERSION 22.46)

APPENDIX F

PROGRAM LISTING FOR PROGRAM LOGIT

```

-----
00001 C    PROGRAM LOGIT.FLX
00002 C    UTILITY PROGRAM FOR THE BORE HOLE GEOPHYSICAL DATA BASE
00003 C    THE PROGRAM DOES THE FOLLOWING FOLLOWING:
00004 C          ADDS NEW LOGS TO THE SYSTEM
00005 C          ALLOWS THE USER TO EDIT A LOG
00006 C          EXTRACTS INFORMATION FOR OTHER SYSTEMS
00007 C          PLOTS A LOG
00008 C    LOGICAL ALL_DONE,FOUND_THE_WELL,OUT_OF_DATA,WELL_IN_CIRMS,
00009 C    +      ASKED_FOR_NAME,OK
00010 C    BYTE TITLE_OF_AN_AREA(28,30),TEMP_STORAGE(200),
00011 C    +      A_NEW_FILE(30),LOG_TYPE(3),TOTAL_LOG_HEADER(156),
00012 C    +      WHO_CAN_WE_BLAZE(40),A_WELL_NAME(12),CONTROL
00013 C    INTEGER*4 THE_TOTAL_RECORD(128),IPICK(20)
00014 C    DIMENSION NUMBER_ON_FILE(127),Z(5000),Y(5000)
00015 C    CHARACTER*26 WELL_LOG_TYPE_NAME(10)
00016 C    CHARACTER*8 AN_AREA_FILE
00017 C    INCLUDE 'WHEAD.CMM'
00018 C    INCLUDE 'LOGHEAD.CMM'
00019 C    EQUIVALENCE(THE_TOTAL_RECORD,NUMBER_OF_LOG_TYPES_IN_USE)
00020 C    EQUIVALENCE(TOTAL_LOG_HEADER,NUMBER_OF_PAIRS)
00021 C    DATA WELL_LOG_TYPE_NAME/
00022 C    +      'GAMMA GAMMA (DENSITY)',
00023 C    +      'NATURAL GAMMA',
00024 C    +      'NEUTRON EPITHERMAL NEUTRON',
00025 C    +      'CALIPER',
00026 C    +      'SONIC',
00027 C    +      'SP AND R',
00028 C    +      'NEUTRON GAMMA',
00029 C    +      'SP',
00030 C    +      'TEMPERATURE',
00031 C    +      'FLOW METERR'/
00032 C
00033 C
00034 C    INITIALIZE-FOR-A-RUN
00035 C
00036 C
00037 C    REPEAT UNTIL(ALL_DONE)
00038 C    . GIVE-OPTIONS
00039 C    . IMPLIMENT-A-OPTION
00040 C    ...FIN
00041 C
00042 C 999 CONTINUE
00043 C    CLOSE(UNIT=2)
00044 C    STOP

```

```

-----
00045 C    TO INITIALIZE-FOR-A-RUN
00046 C    . IPPOINT=0
00047 C    . ALL_DONE=.FALSE.
00048 C    . ASKED_FOR_NAME=.FALSE.
00049 C    . OPEN(UNIT=2,NAME='MASTER.LOG',ACCESS='DIRECT',RECL=128,
00050 C    +      TYPE='OLD')

```

```
00051      . OPEN(UNIT=7,NAME='WNAME.DAT',ACCESS='KEYED',
00052      +.      TYPE='OLD',FORM='FORMATTED',KEY=(2:13),
00053      +.      ORGANIZATION='INDEXED',RECL=79,READONLY)
00054      ...FIN
```

```
-----
00055      TO GIVE-OPTIONS
00056      . IMAX=4
00057      200 . TYPE *, 'PICK ONE OF THE FOLLOWING'
00058      . TYPE *, '(0) STOP'
00059      . TYPE *, '(1) LIST ALL WELLS ON FILE FOR AN AREA'
00060      . TYPE *, '(2) MAKE A SUMMARY OF ALL WELLS ON THE PRINTER'
00061      . TYPE *, '(3) ADD A NEW LOG'
00062      . TYPE *, '(4) ENTER THE NAME OF THE WELL YOU WANT TO WORK ON'
00063      . IF(IPOINT.GT.0)
00064      . . TYPE 210, WELL-LOG-TYPE-NAME(ITYPE),WELL-NAME
00065      210 . . FORMAT(' FOR A ',A26,' LOG OF WELL ',12A1,' YOU CAN:')
00066      . . IMAX=11
00067      . . TYPE *, ' (5) PLOT OUT THE LOG'
00068      . . TYPE *, ' (6) DUMP THE LOG TO THE PRINTER'
00069      . . TYPE *, ' (7) SAMPLE THE AT EVEN INTERVALS'
00070      . . TYPE *, ' (8) DELETE THE LOG'
00071      . . TYPE *, ' (9) CREATE A FILE OF RAW X AND Y POINT'
00072      . . TYPE *, ' (10) CREATE A ROCKWELL FORMATTED FILE'
00073      . . TYPE *, ' (11) EDIT THE LOG'
00074      . ...FIN
00075      . READ (5,*,ERR=200,END=999) IOPTION
00076      . IF(IOPTION.EQ.0)
00077      . . CLOSE(UNIT=2)
00078      . . STOP 'I QUIT'
00079      . ...FIN
00080      . IF(IOPTION.LT.1.OR,IOPTION.GT.IMAX)
00081      . . WRITE(5,*) 'INVALID NUMBER'
00082      . . GO TO 200
00083      . ...FIN
00084      ...FIN
```

```
-----
00085      TO IMPLIMENT-A-OPTION
00086      . SELECT (IOPTION)
00087      . . (1) LIST-WELLS-BY-AREA
00088      . . (2) MAKE-A-SUMMARY-LIST-OF-ALL-WELLS
00089      . . (3) ADD-A-NEW-LOG
00090      . . (4) ENTER-A-WELL-NAME-TO-WORK-WITH-AND-READ-IN-THE-DATA
00091      . . (5) PLOT-OUT-A-LOG
00092      . . (6) DUMP-A-LOG-TO-THE-PRINTER
00093      . . (7) SAMPLE-A-LOG-FOR-OUTPUT
00094      . . (8) DELETE-A-LOG
00095      . . (9) CREATE-A-FILE-OF-RAW-X-AND-Y-POINTS
00096      . . (10) CREATE-A-ROCKWELL-FORMATTED-FILE
00097      . . (11)EDIT-A-LOG
00098      . ...FIN
00099      ...FIN
```

```

-----
00100      TO LIST-WELLS-BY-AREA
00101      100 . TYPE *, 'ENTER THE AREA(100,200,300,600,1100 OR 3000)'
00102          . READ (5,*,ERR=100,END=999) IAREA
00103          . OK=.FALSE.
00104          . SELECT(IAREA)
00105          . . (100)
00106          . . . OK=.TRUE.
00107          . . . AN_AREA_FILE='0100.ARA'
00108          . . . ...FIN
00109          . . (200)
00110          . . . OK=.TRUE.
00111          . . . AN_AREA_FILE='0200.ARA'
00112          . . . ...FIN
00113          . . (300)
00114          . . . AN_AREA_FILE='0300.ARA'
00115          . . . OK=.TRUE.
00116          . . . ...FIN
00117          . . (600)
00118          . . . AN_AREA_FILE='0600.ARA'
00119          . . . OK=.TRUE.
00120          . . . ...FIN
00121          . . (1100)
00122          . . . AN_AREA_FILE='1100.ARA'
00123          . . . OK=.TRUE.
00124          . . . ...FIN
00125          . . (3000)
00126          . . . AN_AREA_FILE='3000.ARA'
00127          . . . OK=.TRUE.
00128          . . . ...FIN
00129          . . . ...FIN
00130          . IF(.NOT.OK)
00131          . . TYPE *, 'INVALID AREA TRY AGAIN'
00132          . . GO TO 100
00133          . . ...FIN
00134          . OPEN(UNIT=1,NAME=AN_AREA_FILE,TYPE='OLD',READONLY)
00135          . READ(1,110,END=120) A_WELL_NAME,NPOINT
00136      110 . FORMAT(1X,12A1,17)
00137          . REWIND(1)
00138          . GO TO 130
00139      120 . CONTINUE
00140          . TYPE *, 'NO WELLS ON FILE FOR THIS AREA'
00141          . GO TO 160
00142      130 . CONTINUE
00143          . REPEAT UNTIL(I,NE,I)
00144          . . READ(1,110,END=160) A_WELL_NAME,NPOINT
00145          . . TYPE 140, A_WELL_NAME
00146      140 . . FORMAT (1X,12A1)
00147          . . READ(2,NPOINT) NUMBER_OF_TYPES,NUMBER_ON_FILE
00148          . . WHEN(NUMBER_OF_TYPES.GT.0)
00149          . . . DO(ITYP=1,10)
00150          . . . . IF(NUMBER_ON_FILE(ITYP).GT.0)
00151          . . . . . TYPE 150, NUMBER_ON_FILE(ITYP),WELL_LOG_TYPE_NAME(ITYP)
00152      150 . . . . . FORMAT(6X,14,1X,A26,' LOG(S) ON FILE')
00153          . . . . . ...FIN

```

```

00154      . . . FIN
00155      . . . FIN
00156      . . ELSE TYPE *, ' ALL LOGS FOR THIS WELL HAVE BEEN DELETED'
00157      . . . FIN
00158      160 . CONTINUE
00159      . CLOSE(UNIT=1)
00160      . . . FIN

```

```

-----
00161      TO MAKE-A-SUMMARY-LIST-OF-ALL-WELLS
00162      . READ(2,1)NUMBER_OF_WELLS_ON_FILE
00163      . WRITE(6,300)
00164      300 . FORMAT(1X,T2,'TYPE',T29,'VERSIONS',T44,'WELL',T61,
00165      +. 'CASE ELEVATION',T96,'COORDINATES')
00166      . DO(IWELL=2,NUMBER_OF_WELLS_ON_FILE+1)
00167      . . READ(2,IWELL)THE_TOTAL_RECORD
00168      C . . DUMP OUT A RECORD
00169      . . ICOUNT=0
00170      . . DO(ITYP=1,10)
00171      . . . IF(NUMBER_OF_REPLICATES_ON_DISK(ITYP).GT.0)
00172      . . . . ICOUNT=ICOUNT+1
00173      . . . . WRITE(6,320) WELL_LOG_TYPE_NAME(ITYP),
00174      +. . . . . NUMBER_OF_REPLICATES_ON_DISK(ITYP),
00175      +. . . . . WELL_NAME,CASE_ELEVATION,PLANT_OR_RICHLAND_CODE,
00176      +. . . . . NORTH_OR_SOUTH_CODE,REAL_Y_COORD_VALUE,
00177      +. . . . . EAST_OR_WEST_CODE,REAL_X_COORD_VALUE
00178      320 . . . . . FORMAT(1X,A26,1X,I4,10X,12A1,F12.1,22X,A1,A1,
00179      +. . . . . F11.2,1X,A1,F11.2)
00180      . . . . . FIN
00181      . . . . . FIN
00182      . . IF(ICOUNT.EQ.0)
00183      . . . WRITE(6,330) WELL_NAME
00184      330 . . . FORMAT(' THIS WELL HAS NO LOGS ON FILE',T42,12A1)
00185      . . . . . FIN
00186      . . . . . FIN
00187      . CLOSE(UNIT=6)
00188      . . . FIN

```

```

-----
00189      TO ENTER-A-WELL-NAME-TO-WORK-WITH-AND-READ-IN-THE-DATA
00190      350 . TYPE *, 'ENTER THE NAME OF THE WELL'
00191      . DO(I=1,12)WELL_NAME(I)= ' '
00192      . READ(5,410)NQ,(TEMP_STORAGE(L),L=1,NQ)
00193      . IF(NQ.GT.12)
00194      . . TYPE *, 'THE NAME IS TOO LONG'
00195      . . IF(NQ.GT.200) STOP 'I QUIT'
00196      . . GO TO 350
00197      . . . FIN
00198      . DO(I=1,NQ)WELL_NAME(I)=TEMP_STORAGE(I)
00199      . IPOINT=INDEX_FOR_A_WELL(WELL_NAME,IAREA)
00200      . IF(IAREA.NE.0)
00201      . . WHEN(IPOINT.EQ.0)
00202      . . . TYPE 360,WELL_NAME
00203      360 . . . . . FORMAT(1X,12A1,' WELL NOT FOUND')

```

```

00204      . . ...FIN
00205      . . ELSE
00206      . . . TYPE *, 'THE WELL WAS FOUND IN THE ', IAREA, ' AREA'
00207      . . . READ(2, IPOINT) THE_TOTAL_RECORD
00208      . . . WHEN (NUMBER_OF_LOG_TYPES_IN_USE, GT, 0)
00209      . . . . IFOUND=0
00210      . . . . DO (ITYPE=1, 10)
00211      . . . . . IF (NUMBER_OF_REPLICATES_ON_DISK (ITYPE), GT, 0)
00212      . . . . . . IFOUND=IFOUND+1
00213      . . . . . . IPICK (IFOUND)=ITYPE
00214      . . . . . ...FIN
00215      . . . . ...FIN
00216      . . . . IF (IFOUND, NE, NUMBER_OF_LOG_TYPES_IN_USE)
00217      . . . . . TYPE *, 'INTERNAL CHECK BAD'
00218      . . . . . WRITE(6, 380) WELL_NAME, IFOUND, NUMBER_OF_LOG_TYPES_IN_USE
00219      . . . . . FORMAT(1X, 'BAD WELL =', I2A1, 'TYPES FOUND=', I3,
380      . . . . . 'TYPES RECORDED =', I3)
00220      . . . . . +. . . . . STOP 'I QUIT'
00221      . . . . . ...FIN
00222      . . . . . WHEN (IFOUND, GT, 1)
00223      . . . . . . TYPE *, 'PICK ONE OF THE FOLLOWING TYPES'
00224      . . . . . . DO (I=1, IFOUND)
00225      . . . . . . . ITYP=IPICK(I)
00226      . . . . . . . TYPE 395, I, WELL_LOG_TYPE_NAME (ITYP)
00227      . . . . . . . FORMAT(1X, I2, 1X, A26)
00228      . . . . . . . ...FIN
00229      . . . . . . . READ (5, *, ERR=390, END=999) IWANT
00230      . . . . . . . IF (IWANT, LT, 1, OR, IWANT, GT, IFOUND)
00231      . . . . . . . . TYPE *, 'INVALID NUMBER'
00232      . . . . . . . . GO TO 390
00233      . . . . . . . ...FIN
00234      . . . . . . . ITYPE=IPICK (IWANT)
00235      . . . . . . . ...FIN
00236      . . . . . . . ELSE ITYPE=IPICK (1)
00237      . . . . . . . WHEN (NUMBER_OF_REPLICATES_ON_DISK (ITYPE), GT, 1)
00238      . . . . . . . . TYPE *, 'ENTER WHICH VERSION OF THE LOG YOU NEED'
00239      . . . . . . . . TYPE *, 'THERE ARE ', NUMBER_OF_REPLICATES_ON_DISK (ITYPE),
396      . . . . . . . . 'ON FILE'
00240      . . . . . . . . READ (5, *, ERR=396, END=999) IREP
00241      . . . . . . . . IF (IREP, LT, 1, OR, IREP, GT, NUMBER_OF_REPLICATES_ON_DISK (ITYPE))
00242      . . . . . . . . . TYPE *, 'INVALID NUMBER'
00243      . . . . . . . . . GO TO 396
00244      . . . . . . . . . ...FIN
00245      . . . . . . . . . ...FIN
00246      . . . . . . . . . ELSE IREP=1
00247      . . . . . . . . . READ=IN=ALL=DATA=FOR=THIS=LOG
00248      . . . . . . . . . ...FIN
00249      . . . . . . . . . ELSE
00250      . . . . . . . . . . TYPE 370, WELL_NAME
00251      . . . . . . . . . . FORMAT(1X, 'ALL WELL LOGS DELETED FROM THIS WELL', I2A1)
00252      . . . . . . . . . . IPOINT=0
00253      . . . . . . . . . . ...FIN
00254      . . . . . . . . . . ...FIN
00255      . . . . . . . . . . ...FIN
00256      . . . . . . . . . . ...FIN
00257      . . . . . . . . . . ...FIN
00258      . . . . . . . . . . ...FIN

```

```

-----
00259      TO READ-IN-ALL-DATA-FOR-THIS-LOG
00260      .   CALL OPEN,_AN,_OLD,_FILE,_ON,_UNIT3(WELL_NAME,
00261      +,      LOG_TYPES_ID(1,ITYPE),IREP)
00262      .   READ(3) TOTAL_LOG_HEADER
00263      .   ITOTAL=0
00264      .   DO(IPAIRS=1,NUMBER_OF_PAIRS)
00265      .     .   WHEN(IPAIRS,GT,5000)
00266      .     .     .   TYPE *,'STORAGE OVER FLOW'
00267      .     .     .   ...FIN
00268      .     .   ELSE READ(3,END=389) Z(IPAIRS),Y(IPAIRS)
00269      .     .   ...FIN
00270      389 .   CLOSE(UNIT=3)
00271      .   ...FIN

```

```

-----
00272      TO ADD-A-NEW-LOG
00273      C      .
00274      400 .   TYPE *, 'ENTER THE NEW FILES NAME'
00275      .   READ (5,410,ERR=400,END=999) NQ,(A_NEW_FILE(L),L=1,NQ)
00276      410 .   FORMAT(0,79A1)
00277      .   A_NEW_FILE(NQ+1)=0
00278      .   OPEN(UNIT=3,NAME=A_NEW_FILE,TYPE='OLD',READONLY,
00279      +,      FORM='UNFORMATTED',ERR=400,RECL=500)
00280      .   READ(3)WELL_NAME,LOG_TYPE,DATE_LOG_WAS_MADE,
00281      +,      NUMBER_OF_XY_BLOCKS,
00282      +,      NUMBER_OF_PAIRS_IN_LAST_BLOCK,SMALLEST_DATUM,
00283      +,      HIGHEST_DATAUM,
00284      +,      STARTING_DEPTH,DEEPEST_MEASUREMENT
00285      C      .
00286      .   TYPE 420,WELL_NAME
00287      420 .   FORMAT(' YOU HAVE READ IN DATA FOR WELL ',12A1)
00288      C      .
00289      .   IPOINT=INDEX_FOR_A_WELL(WELL_NAME,IAREA)
00290      .   ITYPE=WELL_TYPE(LOG_TYPE)
00291      .   IF(ITYPE,GT,0)
00292      .     .   TYPE 430,WELL_LOG_TYPE_NAME(ITYPE)
00293      430 .     .   FORMAT(' WITH A ',A26,' LOG')
00294      .     .   ...FIN
00295      .   IF(IAREA,NE,0,AND,ITYPE,NE,0)
00296      C      .     .   AT LEAST THE WELL NAME MAY BE RIGHT
00297      .     .   WHEN(IPOINT,EQ,0)
00298      .     .     .   TYPE *,'THIS WELL IS NOT YET ON FILE'
00299      .     .     .   TYPE *,'GIVE ME A MINUTE TO CHECK IT OUT'
00300      C      .     .     .   SEE IF THE WELL NAME IS ON FILE WITH CIRMIS(I,E,GOOD)
00301      .     .     .   CALL CIRMIS(IERROR,IAREA,I_CHANGED_THE_NAME) IFILL HEADER
00302      .     .     .   IF(IERROR,EQ,0)
00303      .     .     .     .   IF(I_CHANGED_THE_NAME,EQ,1)
00304      .     .     .     .     .   ICHECK=IAREA
00305      .     .     .     .     .   IPOINT=INDEX_FOR_A_WELL(WELL_NAME,IAREA)
00306      .     .     .     .     .   IF(IAREA,NE,ICHECK)
00307      .     .     .     .     .   TYPE *,'INTERNAL CHECK ERROR',ICHECK,IAREA
00308      .     .     .     .     .   STOP 'I QUIT'
00309      .     .     .     .     .   ...FIN

```



```

00310      . . . . . FIN
00311      . . . . . WHEN(IPOINT,EQ,0)
00312      C      . . . . . CONTINUE WITH THE PROCESS
00313      . . . . . NUMBER_OF_LOG_TYPES_IN_USE=0
00314      . . . . . DO(ITYP=1,20)
00315      . . . . . . NUMBER_OF_REPLICATES_ON_DISK(ITYP)=0
00316      . . . . . . NUMBER_OF_REPLICATES_ON_TAPE(ITYP)=0
00317      . . . . . . DO(K=1,3)LOG_TYPES_ID(K,ITYP)= ' '
00318      . . . . . . FIN
00319      . . . . . . DELETION_KEY=' '
00320      . . . . . . FIN
00321      . . . . . ELSE READ(2'IPOINT)THE_TOTAL_RECORD
00322      . . . . . FIN
00323      . . . . . FIN
00324      . . . . . ELSE
00325      . . . . . IERROR=0
00326      . . . . . READ(2'IPOINT)THE_TOTAL_RECORD
00327      . . . . . FIN
00328      . . . . . IF(IERROR,EQ,0)
00329      . . . . . WHEN(NUMBER_OF_REPLICATES_ON_DISK(ITYPE),EQ,0)
00330      . . . . . . NUMBER_OF_REPLICATES_ON_DISK(ITYPE)=1
00331      . . . . . . DO(K=1,3)LOG_TYPES_ID(K,ITYPE)=LOG_TYPE(K)
00332      . . . . . . NUMBER_OF_LOG_TYPES_IN_USE=NUMBER_OF_LOG_TYPES_IN_USE+1
00333      . . . . . . FIN
00334      . . . . . ELSE
00335      . . . . . . NUMBER_OF_REPLICATES_ON_DISK(ITYPE)=
00336      +. . . . . . NUMBER_OF_REPLICATES_ON_DISK(ITYPE)+1
00337      . . . . . . FIN
00338      . . . . . REPEAT UNTIL(NQ.LE,79)
00339      . . . . . . TYPE *,'ENTER A COMMENT HERE(79A1) '
00340      . . . . . . DO(L=1,79)A_COMMENT(L)= ' '
00341      . . . . . . READ (5,410) NQ,(TEMP_STORAGE(L),L=1,NQ)
00342      . . . . . . IF(NQ.GT,79)
00343      . . . . . . . TYPE *,'COMMENT TOO LONG'
00344      . . . . . . . IF(NQ.GT,200)STOP 'I QUIT'
00345      . . . . . . FIN
00346      . . . . . . FIN
00347      . . . . . DO(I=1,NQ)A_COMMENT(I)=TEMP_STORAGE(I)
00348      . . . . . IF(.NOT.ASKED_FOR_NAME)ASK-FOR-A-NAME
00349      . . . . . CALL DATE(DATE_OF_LAST_UPDATE)
00350      . . . . . CALL DATE(DATE_IT_WAS_ENTERED)
00351      . . . . . DO(I=1,40)
00352      . . . . . . PERSON_WHO_MADE_THE_LAST_CHANGE(I)=WHO_CAN_WE_BLAAME(I)
00353      . . . . . . PERSON_WHO_ENTERED_THIS_LOG(I)=WHO_CAN_WE_BLAAME(I)
00354      . . . . . . FIN
00355      . . . . . WHEN(IPOINT,EQ,0)
00356      C      . . . . . A BRAND NEW WELL LOG
00357      . . . . . READ(2'1)NUMBER_OF_WELLS_ON_FILE
00358      . . . . . . NUMBER_OF_WELLS_ON_FILE=NUMBER_OF_WELLS_ON_FILE+1
00359      . . . . . . IPOINT=NUMBER_OF_WELLS_ON_FILE+1
00360      . . . . . . CALL ADD_THIS_LOG_TO_THE_SYSTEM(IAREA,IPOINT,ITYPE,1,Z,Y)
00361      . . . . . . WRITE(2'1)NUMBER_OF_WELLS_ON_FILE
00362      . . . . . . FIN
00363      . . . . . ELSE CALL ADD_THIS_LOG_TO_THE_SYSTEM(IAREA,IPOINT,ITYPE,0,Z,Y)
00364      . . . . . WRITE(2'IPOINT)THE_TOTAL_RECORD
00365      . . . . . FIN

```

```

00366      . ...FIN
00367      . CLOSE(UNIT=3)
00368      . IPOINT=0
00369      ...FIN

```

```

-----
00370      TO ASK-FOR-A-NAME
00371      . ASKED_FOR_NAME=.TRUE.
00372      . DO(I=1,40) WHO_CAN_WE_BLAAME(I)= ' '
00373      . TYPE *, ' ENTER YOUR NAME HERE'
00374      . READ (5,500) NQ,(WHO_CAN_WE_BLAAME(I),I=1,NQ)
00375      500 . FORMAT(Q,40A1)
00376      ...FIN

```

```

-----
00377      TO EDIT-A-LOG
00378      . REPEAT UNTIL(IWANT,EQ,8)
00379      . . REPEAT UNTIL(IWANT,GE,1,AND,IWANT,LE,8)
00380      600 . . . TYPE *, 'PICK ONE OF THE FOLLOWING OPTIONS'
00381      . . . TYPE *, '----->(1) REMOVE POINTS BY DEPTH'
00382      . . . TYPE *, '----->(2) REMOVE POINTS BY NUMBER'
00383      . . . TYPE *, '----->(3) CHANGE POINTS BY DEPTH'
00384      . . . TYPE *, '----->(4) CHANGE POINTS BY NUMBER'
00385      . . . TYPE *, '----->(5) PLOT THE CURRENT LOG'
00386      . . . TYPE *, '----->(6) PRINT THE CURRENT LOG'
00387      . . . TYPE *, '----->(7) REPLACE THE OLD DATA WITH THE NEW'
00388      . . . TYPE *, '----->(8) ALL DONE'
00389      . . . READ (5,*,ERR=600,END=999) IWANT
00390      . . . IF(IWANT.LT.1,OR,IWANT.GT.8) TYPE *, 'INVALID NUMBER'
00391      . . . ...FIN
00392      . . . IF(IWANT.LT.8)
00393      . . . . IF(IWANT,EQ,1,OR,IWANT,EQ,3)
00394      . . . . . REPEAT UNTIL(OK)
00395      610 . . . . . TYPE *, 'ENTER DEPTH RANGE'
00396      . . . . . READ (5,*,END=999,ERR=610)DEPTH1,DEPTH2
00397      . . . . . OK=DEPTH1,LE,DEPTH2
00398      . . . . . IF(DEPTH1,GT,HEIGHEST_DATUM)
00399      . . . . . . TYPE *, 'OUT SIDE OF RANGE(HIGH SIDE)',HEIGHEST_DATUM
00400      . . . . . . OK=.FALSE.
00401      . . . . . . ...FIN
00402      . . . . . IF(DEPTH2,LT,SMALLEST_DATUM)
00403      . . . . . . TYPE *, 'OUT SIDE OF RANGE(LOW SIDE)',SMALLEST_DATUM
00404      . . . . . . OK=.FALSE.
00405      . . . . . . ...FIN
00406      . . . . . IF(.NOT.OK)TYPE *, 'INVALIDE NUMBERS'
00407      . . . . . . ...FIN
00408      . . . . . ...FIN
00409      . . . . IF(IWANT,EQ,2,OR,IWANT,EQ,4)
00410      . . . . . REPEAT UNTIL(OK)
00411      620 . . . . . TYPE *, 'ENTER NUMBER RANGE(LOW,HIGH)'
00412      . . . . . READ (5,*,END=999,ERR=620) NUMBER1,NUMBER2
00413      . . . . . OK=NUMBER1,LE,NUMBER2,AND,NUMBER1,GT,0
00414      . . . . . IF(.NOT,OK) TYPE *, 'INVALIDE ORDER'
00415      . . . . . ...FIN

```

```

00416      . . . ...FIN
00417      . . . IF(IWANT, EQ, 3, OR, IWANT, EQ, 4)
00418 630      . . . TYPE *, 'ENTER OFFSET AND MULTIPLICATION FACTOR'
00419      . . . READ (5, *, END=999, ERR=630) OFFSET, FACTOR_TO_MULTIPLY_BY
00420      . . . DO(I=1, NUMBER_OF_PAIRS)
00421      . . . WHEN(IWANT, EQ, 3)
00422      . . . IF(Z(I), GE, DEPTH1, AND, Z(I), LE, DEPTH2)
00423      . . . Y(I)=Y(I)*FACTOR_TO_MULTIPLY_BY+OFFSET
00424      . . . ...FIN
00425      . . . ...FIN
00426      . . . ELSE
00427      . . . IF(I, GE, NUMBER1, AND, I, LE, NUMBER2)
00428      . . . Y(I)=Y(I)*FACTOR_TO_MULTIPLY_BY +OFFSET
00429      . . . ...FIN
00430      . . . ...FIN
00431      . . . ...FIN
00432      . . . ...FIN
00433      . . . IF(IWANT, EQ, 1, OR, IWANT, EQ, 2)
00434      . . . IF(IWANT, EQ, 1)
00435      . . . NUMBER2=1
00436      . . . NUMBER1=NUMBER_OF_PAIRS
00437      . . . DO(I=1, NUMBER_OF_PAIRS)
00438      . . . IF(Z(I), GE, DEPTH1, AND, Z(I), LE, DEPTH2)
00439      . . . NUMBER1=MIN(I, NUMBER1)
00440      . . . NUMBER2=MAX(I, NUMBER2)
00441      . . . ...FIN
00442      . . . ...FIN
00443      . . . ...FIN
00444      . . . IOFFSET=NUMBER2-NUMBER1+1
00445      . . . WHEN(IOFFSET, LT, NUMBER_OF_PAIRS)
00446      . . . NUMBER_OF_PAIRS=NUMBER_OF_PAIRS-IOFFSET
00447      . . . DO(I=NUMBER1, NUMBER_OF_PAIRS)
00448      . . . Z(I)=Z(I+IOFFSET)
00449      . . . Y(I)=Y(I+IOFFSET)
00450      . . . ...FIN
00451      . . . SMALLEST_DATUM=Y(1)
00452      . . . HIGHEST_DATUM=Y(1)
00453      . . . STARTING_DEPTH=Z(1)
00454      . . . DEEPEST_MEASUREMENT=Z(NUMBER_OF_PAIRS)
00455      . . . DO(I=1, NUMBER_OF_PAIRS)
00456      . . . SMALLEST_DATUM=MIN(SMALLEST_DATUM, Y(I))
00457      . . . HIGHEST_DATUM=MAX(HIGHEST_DATUM, Y(I))
00458      . . . ...FIN
00459      . . . ...FIN
00460      . . . ELSE
00461      . . . TYPE *, ' WOULD WIPE OUT ALL THE DATA'
00462      . . . TYPE *, ' USE THE DELETE OPTION TO DOE THIS'
00463      . . . TYPE *, ' NO ACTION TAKE HERE'
00464      . . . ...FIN
00465      . . . ...FIN
00466      . . . SELECT(IWANT)
00467      . . . (5) PLOT-OUT-A-LOG
00468      . . . (6) DUMP-A-LOG-TO-THE-PRINTER
00469      . . . (7)
00470      . . . CALL MAKE_A_NEW_VERSION_OF_THIS_LOG(ITYPE, IREP, Z, Y)
00471      . . . ...FIN

```

```
00472      . . . ...FIN
00473      . . ...FIN
00474      C      .
00475      C      .
00476      . ...FIN
00477      . IPOINT=0
00478      ...FIN
```

```
00479      TO PLOT-OUT-A-LOG
00480      . TYPE *, 'GRAPHICS EQUIPMENT NOT YET ON LINE'
00481      ...FIN
```

```
00482      TO SAMPLE-A-LOG-FOR-OUTPUT
00483      . OPEN-A-NEW-FILE
00484      700 . TYPE *, 'THE RANGE OF DEPTH = ', STARTING_DEPTH+OFFSET,
00485      +.   DEEPEST_MEASUREMENT+OFFSET
00486      . TYPE *, 'ENTER STARTING DEPTH, INTERVAL AND MAX NUMBER OF POINTS'
00487      . READ (5, *, ERR=700, END=999) STARTING_DEPTH, DEPTH_INCREMENT,
00488      +.   NUMBER_OF_NEW_POINTS
00489      . IF (STARTING_DEPTH.LT.Z(1)+OFFSET)
00490      . . TYPE *, 'INVALID STARTING DEPTH'
00491      . . GO TO 700
00492      . ...FIN
00493      . IF (DEPTH_INCREMENT.LE.0,0)
00494      . . TYPE *, 'INVALID DEPTH INCREMENT'
00495      . . GO TO 700
00496      . ...FIN
00497      . IF (NUMBER_OF_NEW_POINTS.LE.0)
00498      . . TYPE *, 'INVALID NUMBER OF POINTS'
00499      . . GO TO 700
00500      . ...FIN
00501      . POINT2=STARTING_DEPTH+DEPTH_INCREMENT
00502      . IF (POINT2.GT.Z(NUMBER_OF_PAIRS))
00503      . . TYPE *, 'DEPTH INCREMENT TOO LARGE'
00504      . . GO TO 700
00505      . ...FIN
00506      . CALL EVEN(Z,Y,NUMBER_OF_PAIRS,STARTING_DEPTH,NUMBER_OF_NEW_POINTS,
00507      +.   DEPTH_INCREMENT,OFFSET)
00508      . CLOSE(UNIT=1)
00509      ...FIN
```

```
00510      TO DUMP-A-LOG-TO-THE-PRINTER
00511      . WRITE(6,800) WELL_LOG_TYPE_NAME(ITYPE),WELL_NAME,IREP,
00512      +.   CASE_ELEVATION,
00513      +.   STARTING_DEPTH,DEEPEST_MEASUREMENT,
00514      +.   SMALLEST_DATUM,HEIGHEST_DATUM,
00515      +.   DATE_LOG_WAS_MADE,DATE_IT_WAS_ENTERED,
00516      +.   PERSON_WHO_ENTERED_THIS_LOG
00517      800 . FORMAT(1X,A26,' OF WELL ',I2A1,'(VERSION:',I3,')',
00518      +.   ' HAS A ELEVATION OF ',F10.1,/,
```

```

00519      +.      ' A DEPTH RANGE OF ',2G20.8,,
00520      +.      ' THE LOGGED VALUES RANGE OVER ',2G20.8,,
00521      +.      ' THE LOG WAS MADE ON ',8A1,,
00522      +.      ' AND WAS ENTERED ON THE SYSTEM ON ',9A1,
00523      +.      ' BY ',40A1)
00524      .      OK=,TRUE.
00525      .      DO(I=1,79)IF(A_COMMENT(I).NE,' ')OK=,FALSE.
00526      .      IF(,NOT,OK)WRITE(6,810) A_COMMENT
00527      810      .      FORMAT(' WITH A COMMENT OF:',79A1)
00528      C      .
00529      .      LINES=NUMBER_OF_PAIRS/3
00530      .      DO(LINE=1,LINES)
00531      .      .      IOFF=2
00532      .      .      IOVER=LINE
00533      .      .      CONTROL=' '
00534      .      .      DO(I=1,3)
00535      .      .      .      WRITE(6,820)CONTROL,IOVER,Z(IOVER),Y(IOVER)
00536      820      .      .      .      FORMAT(A1,T<IOFF>,I6,2(1X,G16,6),'!')
00537      .      .      .      CONTROL='+'
00538      .      .      .      IOFF=IOFF+44
00539      .      .      .      IOVER=IOVER+LINES
00540      .      .      ...FIN
00541      .      ...FIN
00542      C      .      NOW DO THE LAST PART
00543      .      IF(MOD(NUMBER_OF_PAIRS,3).NE,0)
00544      .      .      DO(IOVER=LINES*3+1,NUMBER_OF_PAIRS)
00545      .      .      .      WRITE(6,830) IOVER,Z(IOVER),Y(IOVER)
00546      830      .      .      .      FORMAT(T86'!',T90,I6,2(1X,G16,6),'!')
00547      .      .      ...FIN
00548      .      ...FIN
00549      .      CLOSE(UNIT=6)
00550      .      ...FIN

```

```

-----
00551      TO DELETE-A-LOG
00552      .      WHEN(IREP,EQ,NUMBER_OF_REPLICATES_ON_DISK(ITYPE))
00553      .      .      CALL DELETE(IREP,ITYPE)
00554      .      .      NUMBER_OF_REPLICATES_ON_DISK(ITYPE)=NUMBER_OF_REPLICATES_ON_DISK
00555      +.      .      (ITYPE)-1
00556      .      .      IF(NUMBER_OF_REPLICATES_ON_DISK(ITYPE).EQ.0)
00557      .      .      .      NUMBER_OF_LOG_TYPES_IN_USE=0
00558      .      .      ...FIN
00559      .      ...FIN
00560      .      ELSE
00561      .      .      ISAVE=IREP
00562      .      .      DO(IMOVE=ISAVE+1,NUMBER_OF_REPLICATES_ON_DISK(ITYPE))
00563      .      .      .      IREP=IMOVE
00564      .      .      .      READ-IN-ALL-DATA-FOR-THIS-LOG
00565      .      .      .      CALL MAKE_A_NEW_VERSION_OF_THIS_LOG(ITYPE,IMOVE-1,Z,Y)
00566      .      .      ...FIN
00567      .      .      CALL DELETE(NUMBER_OF_REPLICATES_ON_DISK(ITYPE),ITYPE)
00568      .      .      NUMBER_OF_REPLICATES_ON_DISK(ITYPE)=NUMBER_OF_REPLICATES_ON_DISK
00569      +.      .      (ITYPE)-1
00570      .      .      IF(NUMBER_OF_REPLICATES_ON_DISK(ITYPE).EQ.0)
00571      .      .      .      NUMBER_OF_LOG_TYPES_IN_USE=0

```

```

00572      . . ...FIN
00573      . ...FIN
00574      . WRITE(2,'IPOINT')THE_TOTAL_RECORD
00575      . IPOINT=0
00576      ...FIN

```

```

-----
00577      TO CREATE-A-FILE-OF-RAW-X-AND-Y-POINTS
00578      . OPEN-A-NEW-FILE
00579      . DO(I=1,NUMBER_OF_PAIRS)
00580      . . WRITE(1,900) Z(I)+OFFSET,Y(I)
00581 900    . . FORMAT(1X,2G20.8)
00582      . ...FIN
00583      . CLOSE(UNIT=1)
00584      ...FIN

```

```

-----
00585      TO CREATE-A-ROCKWELL-FORMATTED-FILE
00586      . OPEN-A-NEW-FILE
00587      . WRITE(1,920) WELL_NAME,CASE_ELEVATION,REAL_X_COORD_VALUE,
00588      +. REAL_Y_COORD_VALUE,PLANT_OR_RICHLAND_CODE,
00589      +. NORTH_OR_SOUTH_CODE,EAST_OR_WEST_CODE,
00590      +. NUMBER_OF_PAIRS,ITYPE,
00591      +. WELL_LOG_TYPE_NAME(ITYPE),A_COMMENT
00592 920    . FORMAT(1X,12A1,3G20.8,3(1X,A1),2I7,1X,26A1,/,1X,A79)
00593      . DO(I=1,NUMBER_OF_PAIRS)
00594      . . WRITE(1,900) Z(I)+OFFSET,Y(I)
00595      . ...FIN
00596      . CLOSE(UNIT=1)
00597      ...FIN

```

```

-----
00598      TO OPEN-A-NEW-FILE
00599      . REPEAT UNTIL(NQ,LE.14)
00600      . . TYPE *, 'ENTER THE NAME OF THE NEW FILE'
00601      . . READ (5,410,END=999) NQ,(TEMP_STORAGE(L),L=1,NQ)
00602      . . IF(NQ.GT.14)
00603      . . . TYPE *, 'NAME TOO LONG'
00604      . . . IF(NQ.GT.200) STOP 'I QUIT'
00605      . . ...FIN
00606      . ...FIN
00607      . TEMP_STORAGE(L+1)=0
00608      . OPEN(UNIT=1,NAME=TEMP_STORAGE,TYPE='NEW')
00609      . REPEAT UNTIL(IWANT.GT.0.AND.IWANT.LT.4)
00610 1000    . . TYPE *, 'PICK ONE OF THE FOLLOWING'
00611      . . . TYPE *, ' (1) USE GROUND LEVEL FOR THE ORIGIN'
00612      . . . TYPE *, ' (2) USE SEA LEVEL FOR THE ORIGIN'
00613      . . . TYPE *, ' (3) USE 1000 FEET ABOVE SEA LEVEL FOR THE ORIGIN'
00614      . . READ (5,*,ERR=1000,END=999) IWANT
00615      . . IF(IWANT.LT.1.OR.IWANT.GT.3)TYPE *, 'INVALID NUMBER'
00616      . . ...FIN
00617      . SELECT(IWANT)
00618      . . (1) OFFSET=0.0

```

```
00619      . . (2) OFFSET=-CASE_ELEVATION
00620      . . (3) OFFSET=1000.0-CASE_ELEVATION
00621      . ...FIN
00622      ...FIN
00623      END
```

PROCEDURE CROSS-REFERENCE TABLE

```
00272 ADD-A-NEW-LOG
00089

00100 LIST-WELLS-BY-AREA
00087

00370 ASK-FOR-A-NAME
00348

00161 MAKE-A-SUMMARY-LIST-OF-ALL-WELLS
00088

00551 DELETE-A-LOG
00094

00510 DUMP-A-LOG-TO-THE-PRINTER
00092 00468

00598 OPEN-A-NEW-FILE
00483 00578 00586

00085 IMPLIMENT-A-OPTION
00039

00189 ENTER-A-WELL-NAME-TO-WORK-WITH-AND-READ-IN-THE-DATA
00090

00482 SAMPLE-A-LOG-FOR-OUTPUT
00093

00565 CREATE-A-ROCKWELL-FORMATTED-FILE
00096

00377 EDIT-A-LOG
00097

00045 INITIALIZE-FOR-A-RUN
00034

00259 READ-IN-ALL-DATA-FOR-THIS-LOG
00249 00564

00055 GIVE-OPTIONS
00038

00577 CREATE-A-FILE-OF-RAW-X-AND-Y-POINTS
```

00095

00479 PLOT-OUT-A-LOG
00091 00467

(FLECS VERSION 22.46)

```
-----
00624      SUBROUTINE EVEN (DEPTH,DATUM,NUMBER_OF_RAW_POINTS,STARTING_DEPTH1,
00625      +                NUMBER_OF_NEW_POINTS,DEPTH_INCREMENT,OFFSET)
00626      DIMENSION DEPTH(NUMBER_OF_RAW_POINTS),DATUM(NUMBER_OF_RAW_POINTS)
00627      BYTE A_NEW_FILE(30)
00628      LOGICAL OUT_OF_DATA,CREATED_ALL_THE_NEW_POINTS,
00629      +FOUND_THE_RIGHT_INTERVAL
00630      INTEGER RAW_DATA_INDEX
00631      C
00632      C
00633      C
00634      RAW_DATA_INDEX=1
00635      STARTING_DEPTH=STARTING_DEPTH1-OFFSET
00636      NUMBER_OUT=0
00637      REPEAT UNTIL(OUT_OF_DATA .OR. CREATED_ALL_THE_NEW_POINTS)
00638      .   A_NEW_DEPTH=NUMBER_OUT*DEPTH_INCREMENT+STARTING_DEPTH
00639      .   FIND-THE-RIGHT-INTERVAL
00640      .   WRITE-OUT-THE-INTERPOLATED-VALUE
00641      .   CREATED_ALL_THE_NEW_POINTS=NUMBER_OUT.GE.NUMBER_OF_NEW_POINTS
00642      ...FIN
00643      C
00644      RETURN
-----
```

```
00645      TO FIND-THE-RIGHT-INTERVAL
00646      .   REPEAT UNTIL(OUT_OF_DATA.OR.FOUND_THE_RIGHT_INTERVAL)
00647      .   .   OUT_OF_DATA=RAW_DATA_INDEX.GE.NUMBER_OF_RAW_POINTS
00648      .   .   IF(.NOT.OUT_OF_DATA)
00649      .   .   .   FOUND_THE_RIGHT_INTERVAL=A_NEW_DEPTH.GE.DEPTH(RAW_DATA_INDEX)
00650      +.   .   .   .AND.
00651      +.   .   .   A_NEW_DEPTH.LE.DEPTH(RAW_DATA_INDEX+1)
00652      .   .   ...FIN
00653      .   .   IF(.NOT.FOUND_THE_RIGHT_INTERVAL)RAW_DATA_INDEX=RAW_DATA_INDEX+1
00654      .   ...FIN
00655      ...FIN
-----
```

```
00656      TO WRITE-OUT-THE-INTERPOLATED-VALUE
00657      .   IF(.NOT.OUT_OF_DATA)
00658      .   .   RISE=DATUM(RAW_DATA_INDEX+1)-DATUM(RAW_DATA_INDEX)
00659      .   .   RUN =DEPTH(RAW_DATA_INDEX+1)-DEPTH(RAW_DATA_INDEX)
00660      .   .   A_NEW_VALUE=RISE*( (A_NEW_DEPTH-DEPTH(RAW_DATA_INDEX)) /RUN)
00661      +.   .   + DATUM(RAW_DATA_INDEX)+OFFSET
00662      .   .   WRITE(1,100) A_NEW_DEPTH,A_NEW_VALUE
00663      100 .   .   FORMAT(1X,2G15.6)
```



```
00664      . .  NUMBER_OUT=NUMBER_OUT+1
00665      . . .FIN
00666      ...FIN
00667      END
```

PROCEDURE CROSS-REFERENCE TABLE

00656 WRITE-OUT-THE-INTERPOLATED-VALUE
00640

00645 FIND-THE-RIGHT-INTERVAL
00639

(FLECS VERSION 22,46)

```
00668      FUNCTION WELL_TYPE(LOG_TYPE_IN)
00669      C      RETURNS CORESPNDING POINTER
00670      CHARACTER*3 LOG_TYPE
00671      BYTE LOG_TYPE_IN(3),LOG_TYPE_CHAR(3)
00672      CHARACTER*3 LOG_TYPE_IN_USE(10)
00673      EQUIVALENCE(LOG_TYPE_CHAR,LOG_TYPE)
00674      DATA LOG_TYPE_IN_USE/
00675      +      'DEN','NAG','NEW','CAL','SON',
00676      +      'SPR','NEG','SPO','TEM','FLM'/
00677      WELL_TYPE=0
00678      DO(I=1,3)LOG_TYPE_CHAR(I)=LOG_TYPE_IN(I)
00679      DO(I=1,10)
00680      .      IF(LOG_TYPE_IN_USE(I),EQ,LOG_TYPE)WELL_TYPE=I
00681      ...FIN
00682      IF(WELL_TYPE,EQ,0) TYPE 100, LOG_TYPE
00683      100  FORMAT(' INVALID LOG TYPE ',A3)
00684      RETURN
00685      END
```

(FLECS VERSION 22,46)

```
00686      SUBROUTINE DELETE(IREP,ITYPE)
00687      C      DELECT A FILE
00688      CHARACTER*8 FILE
00689      REAL BUFFER(128)
00690      BYTE TOTAL_LOG_HEADER(156)
00691      INCLUDE 'WHEAD.CMM'
00692      INCLUDE 'LOGHEAD.CMM'
00693      EQUIVALENCE (TOTAL_LOG_HEADER,NUMBER_OF_PAIRS)
00694      C
00695      C
00696      C
00697      CALL OPEN_A_FILE_FOR_DEPTH_ON_UNIT_1(WELL_NAME,
00698      +      LOG_TYPES_ID(1,ITYPE),IREP)
```

```
00699      CLOSE(UNIT=1,DISP='DELETE')
00700      RETURN
00701      END
```

(FLECS VERSION 22,46)

```
00702      SUBROUTINE OPEN_A_FILE_FOR_DEPTH_ON_UNIT_1(
00703      +      WELL_NAME,LOG_TYPE,NUMBER_ON_FILE)
00704      BYTE WELL_NAME(12),FILE_NAME(20),LOG_TYPE(3)
00705  C
00706      LENGTH=0
00707      DO(I=1,12)
00708      .   IF(WELL_NAME(I).NE.' ' .AND.WELL_NAME(I).NE.'-')
00709      .   .   LENGTH=LENGTH+1
00710      .   .   FILE_NAME(LENGTH)=WELL_NAME(I)
00711      .   .   FIN
00712      .   .   FIN
00713      IF(LENGTH.GT.8)
00714      .   TYPE 100,WELL_NAME,LENGTH,(FILE_NAME(L),L=1,LENGTH)
00715 100 .   FORMAT(' WELL NAME TOO LONG ',12A1,I7,1X,20A1)
00716      .   STOP 'I QUIT'
00717      .   .   FIN
00718      MAX=1296
00719      IF(LENGTH.EQ.8)MAX=35
00720      IF(NUMBER_ON_FILE.GT.MAX)
00721      .   TYPE 200,WELL_NAME,LOG_TYPE
00722 200 .   FORMAT(' YOU WILL HAVE TO DUMP SOME REPLICATES OF WELL ',
00723      +      12A1,'(TYPE =',3A1,') TO TAPE IN ORDER TO MAKE ROOM')
00724      .   STOP 'I QUIT'
00725      .   .   FIN
00726      CALL BASE36(NUMBER_ON_FILE,FILE_NAME(LENGTH+1))
00727      IF(NUMBER_ON_FILE.LT.36)
00728      .   FILE_NAME(LENGTH+1)=FILE_NAME(LENGTH+2)
00729      .   LENGTH=LENGTH-1
00730      .   .   FIN
00731      FILE_NAME(LENGTH+3)=' '
00732      L=LENGTH+4
00733      DO(I=1,3)
00734      .   FILE_NAME(L)=LOG_TYPE(I)
00735      .   L=L+1
00736      .   .   FIN
00737      FILE_NAME(L)=0
00738      OPEN(UNIT=1,FILE=FILE_NAME,TYPE='OLD',FORM='UNFORMATTED')
00739      RETURN
00740      END
```

(FLECS VERSION 22,46)

```
00741      SUBROUTINE MAKE_A_NEW_VERSION_OF_THIS_LOG(ITYPE,IREP,Z,Y)
00742  C      AND WRITE OUT A FILE WITH ALL THE DATA
00743      CHARACTER*8 FILE
00744      REAL Z(NUMBER_OF_PAIRS),Y(NUMBER_OF_PAIRS)
```

```
00745      BYTE TOTAL_LOG_HEADER(156)
00746      INCLUDE 'WHEAD.CMM'
00747      INCLUDE 'LOGHEAD.CMM'
00748      EQUIVALENCE (TOTAL_LOG_HEADER,NUMBER_OF_PAIRS)
00749      C
00750      C
00751      C
00752      CALL OPEN_A_LOGFILE_ON_UNIT_1(WELL_NAME,
00753      + LOG_TYPES_ID(1,ITYPE),IREP)
00754      WRITE(1) TOTAL_LOG_HEADER
00755      DO(I=1,NUMBER_OF_PAIRS)
00756      . WRITE(1) Z(I),Y(I)
00757      ...FIN
00758      CLOSE(UNIT=1)
00759      RETURN
00760      END
```

(FLECS VERSION 22,46)

```
00761      SUBROUTINE ADD_THIS_LOG_TO_THE_SYSTEM(IAREA,IPOINT,ITYPE,IADD,Z,Y)
00762      C      ADD THE THIS WELL NAME TO THE AREA FILE
00763      C      AND WRITE OUT A FILE WITH ALL THE DATA
00764      CHARACTER*8 FILE
00765      REAL BUFFER(128),Z(5000),Y(5000)
00766      BYTE TOTAL_LOG_HEADER(156)
00767      INCLUDE 'WHEAD.CMM'
00768      INCLUDE 'LOGHEAD.CMM'
00769      EQUIVALENCE (TOTAL_LOG_HEADER,NUMBER_OF_PAIRS)
00770      C
00771      C
00772      SELECT(IAREA)
00773      . (100) FILE='0100,ARA'
00774      . (200) FILE='0200,ARA'
00775      . (300) FILE='0300,ARA'
00776      . (600) FILE='0600,ARA'
00777      . (1100) FILE='1100,ARA'
00778      . (3000) FILE='3000,ARA'
00779      ...FIN
00780      C
00781      CALL OPEN_A_LOGFILE_ON_UNIT_1(WELL_NAME,LOG_TYPES_ID(1,ITYPE),
00782      + NUMBER_OF_REPLICATES_ON_DISK(ITYPE))
00783      NUMBER_OF_PAIRS=(NUMBER_OF_XY_BLOCKS-1)*64+
00784      + NUMBER_OF_PAIRS_IN_LAST_BLOCK
00785      READ-IN-DATA=PAIRS
00786      CHECK-MAX-AND-MIN=VALUES
00787      IF(ITOTAL.NE.NUMBER_OF_PAIRS)
00788      . TYPE *, 'INTERNAL CHECK WARNING ON DATA PAIRS',ITOTAL,
00789      + NUMBER_OF_PAIRS
00790      . NUMBER_OF_PAIRS=ITOTAL
00791      ...FIN
00792      WRITE(1) TOTAL_LOG_HEADER
00793      DO(I=1,NUMBER_OF_PAIRS)
00794      . WRITE(1) Z(I),Y(I)
00795      ...FIN
```

```

00796      CLOSE(UNIT=1)
00797      IF(IADD,EQ,1)
00798      .   OPEN(UNIT=1,NAME=FILE,TYPE='OLD',ACCESS='APPEND')
00799      .   WRITE(1,100) WELL_NAME,IPOINT
00800      100 .   FORMAT(1X,12A1,17)
00801      .   CLOSE(UNIT=1)
00802      ...FIN
00803      RETURN

```

```

-----
00804      TO READ-IN-DATA-PAIRS
00805      .   ITOTAL=0
00806      .   DO(I=1,NUMBER_OF_XY_BLOCKS-1)
00807      .   .   READ(3) BUFFER
00808      .   .   L=1
00809      .   .   DO(II=1,64)MOVE-IT
00810      .   ...FIN
00811      .   LAST_SET_OF_DATA=2*NUMBER_OF_PAIRS_IN_LAST_BLOCK
00812      .   READ(3)(BUFFER(L),L=1, LAST_SET_OF_DATA)
00813      .   L=1
00814      .   DO(IPR=1,NUMBER_OF_PAIRS_IN_LAST_BLOCK) MOVE-IT
00815      .   ...FIN

```

```

-----
00816      TO MOVE-IT
00817      .   ITOTAL=ITOTAL+1
00818      .   IF(ITOTAL,GT,5000)
00819      .   .   TYPE *, 'SYSTEM OUT OF SPACE'
00820      .   .   ITOTAL=5000
00821      .   .   NUMBER_OF_PAIRS=5000
00822      .   ...FIN
00823      .   Z(ITOTAL)=BUFFER(L)
00824      .   Y(ITOTAL)=BUFFER(L+1)
00825      .   L=L+2
00826      .   ...FIN

```

```

-----
00827      TO CHECK-MAX-AND-MIN-VALUES
00828      .   IF(Z(1).NE,STARTING_DEPTH)
00829      .   .   TYPE *, 'ERROR INPUT STARTING DEPTH',Z(1),STARTING_DEPTH
00830      .   .   STARTING_DEPTH=Z(1)
00831      .   ...FIN
00832      .   IF(Z(ITOTAL).NE,DEEPEST_MEASUREMENT)
00833      .   .   TYPE *, 'ERROR IN DEEPEST POINT',Z(ITOTAL),DEEPEST_MEASUREMENT
00834      .   ...FIN
00835      .   DATUM_LOW=Y(1)
00836      .   DATUM_HIGH=Y(1)
00837      .   DO(I=1,ITOTAL)
00838      .   .   DATUM_LOW=MIN(DATUM_LOW,Y(I))
00839      .   .   DATUM_HIGH=MAX(DATUM_HIGH,Y(I))
00840      .   ...FIN
00841      .   IF(DATUM_HIGH.NE,HIGHEST_DATUM)
00842      .   .   TYPE *, 'ERROR IN HIGH LOG DATUM',DATUM_HIGH,HIGHEST_DATUM

```

```
00843      . . HEIGHEST_DATUM=DATUM_HIGH
00844      . ...FIN
00845      . IF(DATUM_LOW,NE,SMALLEST_DATUM)
00846      . . TYPE *,'ERROR IN LOW LOG DATUM',DATUM_LOW,SMALLEST_DATUM
00847      . . SMALLEST_DATUM=DATUM_LOW
00848      . ...FIN
00849      ...FIN
00850      END
```

PROCEDURE CROSS-REFERENCE TABLE

00827 CHECK-MAX-AND-MIN-VALUES
00786

00816 MOVE=IT
00809 00814

00804 READ-IN-DATA-PAIRS
00785

(FLECS VERSION 22.46)

```
00851      SUBROUTINE OPEN_A_LOGFILE_ON_UNIT_1(
00852      +      WELL_NAME,LOG_TYPE,NUMBER_ON_FILE)
00853      BYTE WELL_NAME(12),FILE_NAME(20),LOG_TYPE(3)
00854      C
00855      LENGTH=0
00856      DO(I=1,12)
00857      . IF(WELL_NAME(I),NE,' ',AND,WELL_NAME(I),NE,'-')
00858      . . LENGTH=LENGTH+1
00859      . . FILE_NAME(LENGTH)=WELL_NAME(I)
00860      . ...FIN
00861      ...FIN
00862      IF(LENGTH.GT.8)
00863      . TYPE 100,WELL_NAME,LENGTH,(FILE_NAME(L),L=1,LENGTH)
00864      100 . FORMAT(' WELL NAME TOO LONG ',12A1,I7,1X,20A1)
00865      . STOP 'I QUIT'
00866      ...FIN
00867      MAX=1296
00868      IF(LENGTH.EQ.8)MAX=35
00869      IF(NUMBER_ON_FILE.GT.MAX)
00870      . TYPE 200,WELL_NAME,LOG_TYPE
00871      200 . FORMAT(' YOU WILL HAVE TO DUMP SOME REPLICATES OF WELL ',
00872      +      12A1,'(TYPE =',3A1,') TO TAPE IN ORDER TO MAKE ROOM')
00873      . STOP 'I QUIT'
00874      ...FIN
00875      CALL BASE36(NUMBER_ON_FILE,FILE_NAME(LENGTH+1))
00876      IF(NUMBER_ON_FILE.LT.36)
00877      . FILE_NAME(LENGTH+1)=FILE_NAME(LENGTH+2)
00878      . LENGTH=LENGTH-1
00879      ...FIN
```

```
00880      FILE_NAME(LENGTH+3)=' '
00881      L=LENGTH+4
00882      DO(I=1,3)
00883      .   FILE_NAME(L)=LOG_TYPE(I)
00884      .   L=L+1
00885      ...FIN
00886      FILE_NAME(L)=0
00887      OPEN(UNIT=1,FILE=FILE_NAME,TYPE='NEW',FORM='UNFORMATTED')
00888      RETURN
00889      END
```

(FLECS VERSION 22,46)

```
-----
00890      SUBROUTINE BASE36(NUMBER,DIGIT36)
00891      C   TRANSLATE A BASE 10 NUMBER TO BASE 36
00892      C   ASSUMES NO NUMBER LARGER THAN 36*36 AND
00893      C   THAT ALL ARE POSITIVE
00894      BYTE DIGIT36(2),TEST(0:35)
00895      INTEGER DIGIT(2)
00896      DATA TEST/'0','1','2','3','4','5','6','7','8','9','A','B',
00897      +          'C','D','E','F','G','H','I','J','K','L','M','N',
00898      +          'O','P','Q','R','S','T','U','V','W','X','Y','Z'/
00899      C
00900      DIGIT(1)=NUMBER/36
00901      IF(DIGIT(1).GT.35)
00902      .   TYPE *, 'ERROR IN BASE 36 CALCULATIONS ',DIGIT(1),NUMBER
00903      .   STOP 'I QUIT'
00904      ...FIN
00905      DIGIT(2)=MOD(NUMBER,36)
00906      DO(I=1,2)DIGIT36(I)=TEST(DIGIT(I))
00907      RETURN
00908      END
```

(FLECS VERSION 22,46)

```
-----
00909      SUBROUTINE OPEN_AN_OLD_FILE_ON_UNIT3(
00910      +      WELL_NAME,LOG_TYPE,IREF)
00911      BYTE WELL_NAME(12),FILE_NAME(20),LOG_TYPE(3)
00912      C
00913      LENGTH=0
00914      DO(I=1,12)
00915      .   IF(WELL_NAME(I).NE.' ' .AND. WELL_NAME(I).NE.'-')
00916      .   .   LENGTH=LENGTH+1
00917      .   .   FILE_NAME(LENGTH)=WELL_NAME(I)
00918      .   ...FIN
00919      ...FIN
00920      CALL BASE36(IREF,FILE_NAME(LENGTH+1))
00921      IF(NUMBER_ON_FILE.LT.36)
00922      .   FILE_NAME(LENGTH+1)=FILE_NAME(LENGTH+2)
00923      .   LENGTH=LENGTH-1
00924      ...FIN
00925      FILE_NAME(LENGTH+3)=' '

```

```
00926      L=LENGTH+4
00927      DO(I=1,3)
00928      .   FILE_NAME(L)=LOG_TYPE(I)
00929      .   L=L+1
00930      ...FIN
00931      FILE_NAME(L)=0
00932      OPEN(UNIT=3,FILE=FILE_NAME,TYPE='OLD',FORM='UNFORMATTED')
00933      RETURN
00934      END
```

(FLECS VERSION 22,46)

```
00935      FUNCTION INDEX_FOR_A_WELL (WELL_NAME,IAREA)
00936      C      TAKES THE WELL NAME AND RETURNS THE INDEX FOR
00937      C      THE WELL IF IT EXISTS
00938      C      ELSE IT RETURNS A ZERO
00939      C      IT ALSO RETURNS THE AREA WHERE THE WELL IS LOCATED
00940      BYTE WELL_NAME(12),FILE_NAME(9),A_POSSIBLE_MATCH(12)
00941      LOGICAL A_MATCH
00942      CHARACTER*12 A_WELL
00943      EQUIVALENCE (A_POSSIBLE_MATCH,A_WELL)
00944      EQUIVALENCE (AREA,FILE_NAME)
00945      DATA FILE_NAME/' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '
00946      C
00947      IAREA=0
00948      SELECT(WELL_NAME(1))
00949      .   ('1')
00950      .   .   AREA='0100'
00951      .   .   IAREA=100
00952      .   ...FIN
00953      .   ('2')
00954      .   .   AREA='0200'
00955      .   .   IAREA=200
00956      .   ...FIN
00957      .   ('3')
00958      .   .   AREA='0300'
00959      .   .   IAREA=300
00960      .   ...FIN
00961      .   ('6')
00962      .   .   AREA='0600'
00963      .   .   IAREA=600
00964      .   ...FIN
00965      ...FIN
00966      IF(WELL_NAME(1).EQ.'1'.AND.WELL_NAME(2).EQ.'1')
00967      .   AREA='1100'
00968      .   IAREA=1100
00969      ...FIN
00970      IF(WELL_NAME(1).EQ.'3'.AND.WELL_NAME(2).EQ.'0')
00971      .   AREA='3000'
00972      .   IAREA=3000
00973      ...FIN
00974      WHEN(IAREA,EQ,0)
00975      .   TYPE *,'BAD WELL NAME'
00976      .   INDEX=0
```

```
00977      ...FIN
00978      ELSE
00979      . OPEN(UNIT=1,NAME=FILE_NAME,TYPE='OLD',READONLY)
00980      . A_MATCH=,FALSE. !IN CASE OF NO DATA
00981      . REPEAT UNTIL(A_MATCH)
00982      . . READ(1,100,END=999) A_POSSIBLE_MATCH,IPOINT
00983      100 . . FORMAT(1X,12A1,I7)
00984      . . CALL SAME_WELL(A_POSSIBLE_MATCH,WELL_NAME,A_MATCH)
00985      . ...FIN
00986      999 . CONTINUE
00987      . WHEN(A_MATCH)INDEX_FOR_A_WELL=IPOINT
00988      . ELSE INDEX_FOR_A_WELL=0
00989      . CLOSE(UNIT=1)
00990      ...FIN
00991      RETURN
00992      END
```

(FLECS VERSION 22.46)

```
-----
00993      SUBROUTINE CIRMIS (IERROR,NEW_AREA_ID,I_CHANGED_THE_NAME)
00994      LOGICAL A_MATCH
00995      BYTE A_POSSIBLE_MATCH(12)
00996      INCLUDE 'WHEAD.CMM'
00997      C
00998      IERROR=0
00999      I_CHANGED_THE_NAME=0
01000      A_MATCH=,FALSE.
01001      REPEAT UNTIL ( A_MATCH)
01002      . FIND=A-MATCH
01003      . IF(,NOT,A_MATCH)
01004      . . TYPE 100, WELL_NAME
01005      100 . . FORMAT (' NO MACTH FOUND FOR WELL ',12A1,/,
01006      +. . ' ENTER A NEW NAME OR',/, ' SKIP THIS WELL UP DATE')
01007      . . READ (5,200,ERR=300,END=300)NQ, WELL_NAME
01008      200 . . FORMAT(Q,12A1)
01009      . . IF(NQ,EQ,0) GO TO 300
01010      . . I_CHANGED_THE_NAME=1
01011      . . GO TO 400
01012      300 . . CONTINUE
01013      . . IERROR=1
01014      . . A_MATCH=,TRUE.
01015      400 . . CONTINUE
01016      . ...FIN
01017      ...FIN
01018      CLOSE(UNIT=1)
01019      RETURN
```

```
-----
01020      TO FIND=A-MATCH
01021      . READ (7,500,ERR=999,KEY=WELL_NAME) A_POSSIBLE_MATCH,
01022      +. . Y_MODEL_COORDINATE,
01023      +. . X_MODEL_COORDINATE,
01024      +. . CASE_ELEVATION,PLANT_OR_RICHLAND_CODE,
```


(FLECS VERSION 22,46) 14-OCT-82 15:18:15 PAGE 00023

```
01025      +.      NORTH_OR_SOUTH_CODE,REAL_X_COORD_VALUE,
01026      +.      EAST_OR_WEST_CODE, REAL_Y_COORD_VALUE,
01027      +.      INITIAL_CIRMIS_BLOCK_ADDRESS,
01028      +.      NUMBER_OF_WORDS_IN_A_CIRMIS_REC,
01029      +.      NEW_AREA_ID
01030  500      .      FORMAT(1X,12A1,1X,2F10.0,F7.2,1X,2A1,F8.0,1X,A1,F8.0,2I6,1X,I4)
01031      .      A_MATCH=.TRUE.
01032  999      .      CONTINUE
01033      ...FIN
01034      END
```

PROCEDURE CROSS-REFERENCE TABLE

01020 FIND-A-MATCH
01002

(FLECS VERSION 22,46)

```
01035      SUBROUTINE SAME_WELL(NAME1,NAME2,LOGICAL_RESULT)
01036      LOGICAL LOGICAL_RESULT
01037      BYTE NAME1(12),NAME2(12)
01038      LOGICAL_RESULT=.TRUE.
01039      DO(I=1,12)
01040      .      IF(NAME1(I).NE.NAME2(I))LOGICAL_RESULT=.FALSE.
01041      ...FIN
01042      RETURN
01043      END
```

(FLECS VERSION 22,46)

DISTRIBUTION

<u>No. of Copies</u>		<u>No. of Copies</u>	
<u>OFFSITE</u>			
27	<u>DOE Technical Information Center</u> D. M. Johnson University of Washington Seattle, WA 98105 L. S. Law Colorado School of Mines Golden, CO 80401 J. W. Crosby Washington State University Pullman, WA 99163		K. K. Kover A. G. Law R. K. Ledgersood D. J. Moak J. V. Panesko W. W. Pidcoe W. H. Price A. E. Rowlette F. A. Spane S. R. Strait A. M. Tallman G. L. Wagenaar C. T. Webster T. J. Wood <u>Hanford Environmental Health Foundation</u>
<u>ONSITE</u>			
7	<u>DOE Richland Operations Office</u> D. R. Elle D. E. Gerton A. G. Lassila H. E. Tiernan M. G. White (2)		L. J. Maas <u>Vitro Engineering Corporation</u> S. S. Compton
26	<u>Rockwell Hanford Operations</u> M. K. Addition M. Babad R. L. Biggerstaff L. E. Bruns R. W. Bryce D. L. Chase K. R. Fecht D. L. Gephart M. J. Graham M. H. Hall G. L. Hanson R. L. Jackson	3	<u>Westinghouse Hanford Company</u> G. D. Carpenter R. B. Hall K. D. Hayden
		2	<u>United Nuclear Incorporated</u> T. E. Dabrowski E. A. Weakly
		50	<u>Pacific Northwest Laboratory</u> S. C. Blair (5) P. J. Bramson D. B. Cearlock C. S. Cline

No. of
Copies

J. P. Corley
D. W. Damschen
D. W. Dragnich
P. A. Eddy
D. Gibson
P. J. Gutknecht
E. L. Hilty
J. R. Houston
T. L. Liikala
V. L. McGhan
D. A. Myers
R. G. Parkhurst

No. of
Copies

G. M. Petrie (5)
L. S. Prater (2)
K. R. Price
M. R. Quaders
J. R. Raymond
J. T. Rieger
J. A. Stottlemire
C. M. Unruh
Publishing Coordination BE(2)
Technical Information (5)
Geosciences Research and
Engineering Library (5)