5D

Hanford Meteorological Station Computer Codes Volume 7 - The RIVER Computer Code

G. L. Andrews
J. W. Buck

March 1988

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 Battelle Memorial Institute, nor any of their employees, makes any warranty, expressed 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, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, or Battelle Memorial Institute.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE MEMORIAL INSTITUTE
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 22161

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

HANFORD METEOROLOGICAL STATION
COMPUTER CODES

Volume 7 - The RIVER Computer Code

G. L. Andrews
J. W. Buck

March 1988

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RL0 1830

Pacific Northwest Laboratory
Richland, Washington 99352

PREFACE

This volume is the seventh in a series that describes computer programs routinely used at the Hanford Meteorological Station (HMS) (e.g., data entry programs). Each volume, which documents one computer code, consists of a program overview, user's guide, description of input, a detailed program description and flowchart, and a listing of the source code.

The first seven volumes present the data entry programs used at the HMS for archiving and checking various types of meteorological data. All of these programs use a data entry form to facilitate data input, and each code calls the same series of subroutines that control and display the data entry form at the user's terminal. The subroutines were generated from the program FORMEDIT, which was developed by John Wiberg of the Computer Science staff in the Engineering Physics Department at Pacific Northwest Laboratory. An informal user's guide to the FORMEDIT program that describes how to create, modify, and eliminate data entry forms is available from the HMS system manager.

SUMMARY

The RIVER computer code is used to archive Columbia River data measured at the 100N reactor. The data are recorded every other hour starting at 0100 Pacific Standard Time (12 observations in a day), and consist of river elevation (feet), temperature ($^{\circ}\text{C}$), and flow rate (cubic feet per second). The program prompts the user for river data by using a data entry form. After the data have been entered and verified, the program appends each hour of river data to the end of each corresponding surface observation record for the current day. The appended data are then stored in the current month's surface observation file.

ACKNOWLEDGMENTS

The authors would like to thank the staff at the Hanford Meteorological Station for their assistance during the preparation of this document. Special thanks to the HMS Project Manager Dana Hoitink for his continued support and Ora Gifford for his helpful suggestions and ideas.

CONTENTS

PREFACE	iii
SUMMARY	v
ACKNOWLEDGMENTS	vii
INTRODUCTION	1
THE RIVER COMPUTER CODE	3
PROGRAM OVERVIEW	3
USER'S GUIDE FOR THE RIVER COMPUTER CODE	3
DESCRIPTION OF INPUT	6
TROUBLE-SHOOTING THE RIVER COMPUTER CODE	9
TERMINAL PROBLEMS/FAILURES.	9
SYSTEM PROBLEMS/FAILURES	9
PROGRAM PROBLEMS/FAILURES	10
APPENDIX A - PROGRAM DESCRIPTION AND FLOWCHART	A.1
APPENDIX B - RIVER SOURCE CODE LISTING	B.1

INTRODUCTION

The Hanford Meteorological Station (HMS) was established in 1944 on the Hanford Site to collect and archive meteorological data and to provide weather forecasts and related services for Hanford project activities. The HMS is located in the northwest quadrant of the Hanford Site and is operated by the Pacific Northwest Laboratory (PNL).

Columbia River measurements (hereafter referred to as the Columbia River data) are taken every other hour at the 100N reactor, starting at 0100 Pacific Standard Time (PST). Measurements are taken for the following:

- elevation (feet above sea level)
- temperature (°C)
- flow rate [cubic feet per second (CFS)]

These observations are archived daily throughout the year at the HMS using the RIVER program. The data are maintained in disk files on the HMS computer system [Digital Equipment Corporation (DEC) VAX 11/780].

The following sections in this volume describe the implementation and operation of the RIVER computer code. Appendix A contains a description and flowchart of the RIVER computer code and Appendix B contains a source listing of the RIVER computer code.

THE RIVER COMPUTER CODE

The RIVER computer program provides a method for archiving data obtained from the Columbia River at the 100N reactor. The program has replaced manual entry of data via computer cards and a keypunch. The RIVER program was written, compiled, and linked using the FORTRAN compiler on the DEC VAX 11/750 at the HMS, and is executed once a day during the graveyard shift.

PROGRAM OVERVIEW

The RIVER computer code accesses two input files and one output file. One input file contains the date of the previous river data archive, and the other input file contains 24 hourly surface weather observations from the previous day. The output file consists of hourly surface weather observations and river data for the current month. The convention used in naming files for the monthly surface weather observations is SFCOBSxxx.DAT where the xxx is the three-character abbreviation of the current month (e.g., AUG for August).

The program begins by displaying a data entry form on the terminal screen (Figure 1). The form consists of 37 fields, and the length of each field is indicated by the number of X's adjacent to the field name. The program fills in the date field by accessing an input file to obtain the previous archive's date. You are then prompted to fill in the remaining input fields. You can exit the data entry form after filling in all the input fields. The screen clears, and a prompt to make any needed corrections appears on the screen. Once all the input values are correct, you can exit the program and the program is terminated.

USER'S GUIDE FOR THE RIVER COMPUTER CODE

This section describes how to invoke the RIVER computer code on the HMS DEC VAX 11/750 computer and how to respond to the various prompts associated with the code. We assume that you know how to access the FCST account on the

HMS RIVER INPUT FORM			
HOUR	ELEVATION (FEET)	TEMPERATURE (DEGREES C)	FLOW (CFS)
01	XXXX	XXX	XXXX
03	XXXX	XXX	XXXX
05	XXXX	XXX	XXXX
07	XXXX	XXX	XXXX
09	XXXX	XXX	XXXX
11	XXXX	XXX	XXXX
13	XXXX	XXX	XXXX
15	XXXX	XXX	XXXX
17	XXXX	XXX	XXXX
19	XXXX	XXX	XXXX
21	XXXX	XXX	XXXX
23	XXXX	XXX	XXXX
DATE:xxxxxx			

FIGURE 1. Example of the Data Entry Form Used by the RIVER Computer Code

HMS computer, and that you have some basic knowledge of the VAX operating system. The terminal must be in the DEC VT100 terminal mode before executing the RIVER computer code. To determine the terminal status, simply type SHOW TERMINAL and depress the RETURN key. A list of terminal characteristics appears on the screen. Check to make sure the terminal type is set to VT100. If it is not, type SET TERMINAL/VT100 and depress the RETURN key. Currently, the HMS is equipped with several Tektronix 4107 terminals that can emulate the VT100 terminal.

Basically, there are six steps to executing the RIVER computer code, filling in the data entry form, and terminating the program. These steps are listed below, and each step is discussed in detail in the following paragraphs:

- log in to the HMS computer (need the FCST password)
- type RUN RIVER, then depress the RETURN key
- fill in the data entry form
- exit the form

- answer "yes" or "no" to "need corrections" prompt
- terminate program.

To log in, depress the RETURN key a couple of times. The computer responds with the prompt USERNAME:. Enter FCST and depress the RETURN key. The computer responds with the prompt PASSWORD:. You must know the password to the FCST account before logging in to the HMS computer. Enter the correct password. If you fail to answer either of these prompts correctly, the computer repeats the appropriate prompt. You are given three chances to enter the correct response. By logging in to the FCST account, you are placed in the [FCST] directory and can immediately invoke the RIVER computer code.

The RIVER code is invoked by typing RUN RIVER and depressing the RETURN key. When this action is done, the screen clears and the data entry form appears on the terminal. You can now begin data entry. If problems occur when executing the code, refer to the "Trouble-Shooting" section for help.

The data entry form appears on the screen as shown in Figure 1 with the cursor in field one (ELEVATION for hour 1). When you move the cursor to a new field, a prompt for that field appears on the screen below the data entry form for each field. Included with each prompt is a default value and a valid range or list of values from which the input must be selected. To enter data into a field, simply type in the requested data and depress the RETURN key. Integers are used for all input. The default value may be entered by depressing the ENTER key or the SELECT DEFAULT key, which is found on some Tektronix terminals. This action enters the default value. The input value appears in the field (assuming a valid value or default value was entered) while the cursor moves from left to right into the next field. After data are entered in the rightmost field, the cursor moves down one line to the leftmost field. When values have been entered in all fields, you should check the fields for mistakes. If corrections are needed use the up, down, right, and left arrow keys (↑,↓,→,←) located at the top of the keyboard to move within the form. Once you are satisfied that the entered values are correct, depress the EXIT key to exit the form. You may exit the data entry form from any field.

The RIVER program provides for two phases of quality assurance to the input data. The first phase is applied as you enter the input data for each field. An error message appears at the bottom of the form if the input value is not within the valid range or valid list of matching numbers, and you are then given the opportunity to enter another value for that field. The second phase occurs when you exit the form and the "need corrections" prompt appears below the information entered in the form. You can check the input values for validity and make necessary corrections.

After exiting the form, the program displays the information entered into the data entry form on the terminal in the format in which it is appended to the monthly surface observation file (e.g., SFCOBSxxx.DAT). A prompt appears just below the information asking if you would care to make any corrections. If no corrections are needed, enter an N or n and the program appends the data to the monthly file. If corrections are needed, enter a Y or y and the program returns to the data entry form mode. At this time, the previously entered values appear in their respective fields. Incorrect value(s) can be corrected by using the arrow keys to move the cursor to the appropriate field(s) and revising the incorrect entry. When all corrections have been made, you can exit the form by depressing the EXIT key. The data entry form may be exited from any field; it is not necessary to move to the last field.

The RIVER program is terminated by answering n or N to the "need corrections" prompt. The entered data are appended to the monthly file and the program terminates with a FORTRAN STOP message written to your terminal. The monthly data file for the daily surface weather and river observations are located in the [FCST.DAT] directory. Each file contains all of the hourly surface weather observations and Columbia River data for any single month.

DESCRIPTION OF INPUT

The data entry form consists of 37 fields; numeric (integer) values must be provided in each of the fields. The RIVER program provides the value for the date field. When you move the cursor to one of the other 37 input

fields, the code provides a prompt related to the required input. The prompt consists of a brief description of the field, the valid range or list of matching numbers from which the input value must be selected, and a default value. (Refer to the section, "User's Guide for the RIVER Code," for information on how to input the default value.) Table 1 indicates the unit of measure and range or list of valid values, along with the default value, for the Columbia River elevation, temperature, flow rate, and the date.

TABLE 1. Reference Guide to Input Fields

<u>Input Field</u>	<u>Units of Measure</u>	<u>Range or List</u>
ELEVATION	tenths of feet	3000 - 6000/
RIVER TEMP	°C in tenths	-20 - 250/
RIVER FLOW	hundreds of cubic feet/second	200 - 3000/
DATE	YYMMDD	860101-990101/999999

The following list describes each input field, the associated range or list of valid values from which the input must be selected, and the default value. All variables refer to the current day's river measurements.

- 1) ELEVATION - The input value for this field is the elevation above sea level of the Columbia River at the 100N reactor. The units of measure are in tenths of feet. The range of this value is from 3000 to 6000 tenths of feet. For example, an entry of 3510 means a river elevation of 351.0 feet above sea level. The default value is 9999, indicating a missing elevation measurement.
- 2) RIVER TEMP - The input value for this field is the temperature of the Columbia River at the 100N reactor. The units are in tenths of degrees Celsius (°C). The range of this value is from -20 to +250 tenths of °C. For example, a value of 134 indicates a river temperature of 13.4°C. The default value is 999, indicating a missing river temperature.
- 3) RIVER FLOW - The input value for this field is the flow rate of the Columbia River at the 100N reactor. The units of measure are

in hundreds of CFS. For example, a value of 1001 indicates a flow rate of 100.1 CFS. The default value is 9999, indicating a missing flow rate.

- 4) DATE - The date format is YYMMDD: where YY is the last two digits of the current year (e.g., 87 for 1987), MM is the numerical equivalent of the month (e.g., 04 represents April), and DD is the day of the month. For example, 870421 is April 21, 1987. The range of the date is 860101 (i.e., January 1, 1986) through 990101 (i.e., January 1, 1999). The default value for the date is 999999. This value indicates that the date is missing.

TROUBLE-SHOOTING THE RIVER COMPUTER CODE

This section provides information that may help resolve problems that occur when operating the RIVER computer code. We assume that you are familiar with the VAX EDT editor and the VAX operating system commands. Basically, there are three types of failure or problem areas: terminal (i.e., CRT), system, and/or program. The following sections will discuss problems or failures that may occur when using the RIVER computer code and the suggested recovery procedures. If a series of recovery actions are suggested, you should try them in sequence until the problems or failure are resolved. Remember that the date is incremented each time you execute the RIVER program. Therefore, you may need to edit the [FCST.PREVIOUS_DATE] RIVER_DATE.DAT file to reflect the desired date after you have tried the recovery procedures.

TERMINAL PROBLEMS/FAILURES

Problem: Terminal does not respond.

Recovery: Determine if terminal is on. On Tektronix terminals the on/off switch is located in the front and to the right.

Determine if the dimmer switch is turned all the way down. On Tektronix terminals the dimmer switch is located in the front and to the right. Turn the switch clockwise to increase the light intensity on the screen.

If the above actions fail to get a response from the terminal, ask the instrument specialist (the HMS Room 125) to check the terminal.

SYSTEM PROBLEMS/FAILURES

Problem: System does not respond to operating system commands or carriage returns.

Recovery: Check with the Emergency Meteorological System (EMS) computer manager (Eric Stephan, 376-1107, or Ora Gifford, 373-3216) to see if the HMS computer system is down for maintenance or

repair.

If the EMS computer manager indicates the HMS computer system is operating, turn the terminal off, wait approximately 15 to 20 seconds, and turn the terminal back on. Depress the RETURN key several times to get attention of the computer system.

If the above actions fail ask an instrument specialist to examine the computer and notify the EMS computer manager.

PROGRAM PROBLEMS/FAILURES

Problem: Terminal not correctly displaying data entry form.

Recovery: Depress the CTRL key and the Y key simultaneously to get to the dollar sign prompt. Type SET TERMINAL/VT100 and depress RETURN. Try running the program again.

Depress the SETUP key at the top of the keyboard. The terminal should respond with an asterisk. Type CODE ANSI and depress RETURN, then depress the SETUP key. Try running the program again.

Problem: Terminal displaying garbled data.

Recovery: Depress the SETUP key at the top of the keyboard. The terminal should respond with an asterisk. Type CODE ANSI, depress RETURN, then depress the SETUP key.

Problem: Program RIVER.EXE cannot be found or program has been "trashed" in some way.

Recovery: Copy backup version to the FCST directory. Type
COPY [FCST.DATA_ENTRY]RIVER.EXE [FCST]RIVER.EXE. Try running the RIVER program again.

Problem: Input file ([FCST.PREVIOUS_DATE]RIVER_DATE.DAT) or output file ([FCST.DAT]SFCOBSxxx.DAT where xxx is the three-character abbreviation of the current month) cannot be found.

Recovery: Check the [FCST.PREVIOUS_DATE] directory for the input file or the [FCST.DAT] directory for the output file.

If file(s) is intact, try copying the backup version of RIVER.EXE to the [FCST] directory (see above for directions). Try running the code again.

Problem: RIVER program prints error message and terminates because:

1) there are not 24 records in DAILYSF COBS.DAT, 2) the records in DAILYSF COBS.DAT are not in time sequence, 3) the time or date in DAILYSF COBS.DAT and RIVER_DATE.DAT do not match, or 4) there is an error in the data entry subroutine.

Recovery: Check the DAILYSF COBS.DAT and RIVER_DATE.DAT files for errors. If these files are good, check with Ora Gifford or Gregg Andrews (376-0138).

APPENDIX A

PROGRAM DESCRIPTION AND FLOWCHART

APPENDIX A

PROGRAM DESCRIPTION AND FLOWCHART

This discussion describes, in detail, the operations and logical flow of the RIVER computer code. A flow chart indicating the general operations and logical sequence of the RIVER code is shown at the end of this appendix. The flowchart is divided into three sections by two dashed lines. The top and bottom sections are transparent to the user and do not require input from you. The middle section involves the data entry form, which requires input from you. The following three paragraphs describe the operations within each section. For a more in-depth review of the operations and logical sequence of the program, refer to the source listing in Appendix B.

The RIVER computer program is invoked once a day during the graveyard shift after all the previous day's surface observations have been recorded (Andrews and Buck 1987). Its purpose is to allow you to enter Columbia River data, and then append that data to the end of the hourly surface weather observation data for the same day.

The RIVER computer program begins by opening the [FCST.DAT]DAILYSF COBS.DAT file. This file should contain 24 hourly surface weather observations recorded from the previous day. The program counts the number of hourly surface weather observations as the data is read in. If there are less than 24 records in the DAILYSF COBS.DAT file, the RIVER program sends an error message to the user's terminal and the program is terminated. The error message includes a course of action needed to correct the error. If there are 24 records in the DAILYSF COBS.DAT file, the RIVER program checks the file for correct time sequence (i.e., hours numbered from 1 through 24 in sequence). If the hours are not in sequence, the RIVER program prints out an error message and terminates. Again, a course of action is suggested. If the hours are in correct time sequence, the program opens the [FCST.PREVIOUS_DATE]RIVER_DATE.DAT file to obtain the hour (always 0100 PST) and date from the previous day. The program then compares that date and hour to the first date and hour of the DAILYSF COBS.DAT file, which should be hour 0100 PST and the previous day's date. If there is a mismatch

in time or date the RIVER program prints an error message and terminates. If the times and dates match, the RIVER program opens the current month's surface weather observation file (e.g., SFCOBSxxx.DAT where the xxx are the three-letter abbreviation of the current month). The SFCOBSxxx.DAT file contains the hourly surface weather observations and river data for the current month. A new SFCOBSxxx.DAT file is opened if the data are from the first of the new month. At this point the code calls several subroutines that control and display a data entry form at your terminal. If an error occurs while invoking those subroutines the program sends an error message to the screen and terminates the program.

The RIVER code uses a data entry form to facilitate data entry. The form consists of several input fields (see Fig. 1 in the main text), each of which represents a Columbia River measurement for the specified hour. Each field is displayed in reverse video to highlight the field length. The fields are initially blank. You are prompted from the bottom of the form for an input value. The prompt consists of a brief description of the field, the valid range or list of numbers from which input must be selected, and a default value. A valid entry must be made before the cursor moves to the next field. If the numeric value is valid, the number is displayed in the appropriate field and the cursor moves to the next field when the ENTER key is depressed. If the input value is not valid, an error message occurs at the bottom of the form while the cursor remains in that field. When this occurs, the prompt reappears. After all fields have been filled in you exit the data entry form (see main text for exiting the form) and the program appends the data to the surface weather observation data form (the DAILY SFCOBS.DAT file). The river data is appended to the corresponding surface weather observation data. For example, the 0100 PST river data is appended to the end of the 0100 PST surface weather observation data. This process is repeated for all the river data (0100 PST through 2300 PST). After the river data are appended to the surface weather observation data, the new data records are appended to the end of the current monthly SFCOBSxx.DAT file. A message is sent to your screen indicating the data have been appended to the SFCOBSxxx.DAT file.

At this point, the program checks to see if it is the end of a month.

If so, a new SFCOBSxxx.DAT files is created and the previous month's file is closed. If it is not the end of the month, the current SFCOBSxxx.DAT files are closed. Next, a new DAILY SFCOBS.DAT file is created for the next day's surface weather observations. The date from the RIVER_DATE.DAT file is incremented and written to RIVER_DATE.DAT. The RIVER program terminates after the new date is written to the RIVER_DATE.DAT file. A FORTRAN STOP is sent to your screen.

REFERENCE

Andrews, G. L. and J. W. Buck. 1987. Hanford Meteorological Station Computer Codes - Volume 6: The SFC Computer Code. PNL-6279 Vol. 6, Pacific Northwest Laboratory, Richland, Washington.

FIGURE A.1. Logical Sequence of the RIVER Code

FIGURE A.1. (Cont'd.)

APPENDIX B

RIVER SOURCE CODE LISTING

```

0001 C *****
0002 C
0003 C     PROGRAM: RIVER.FOR
0004 C
0005 C     PROGRAMMER: GREGG ANDREWS
0006 C
0007 C     FUNCTION: TO READ RIVER DATA FROM A FORM INPUT MODE
0008 C               THAN APPEND THE DATA TO THE DAILYSECOBS.DAT
0009 C               FILE AND WRITE IT TO A MONTHLY FILE.
0010 C
0011 C     DATE: 1/20/88
0012 C
0013 C     VERSION: 1.0
0014 C     VERSION: 2.0 GREGG ANDREWS REASON - TO PREVENT THE DATA FROM
0015 C               WRITING TO OUTPUT FILE WHEN ERROR
0016 C               IN NUMBER OF RECORDS IN
0017 C               DAILYSECOBS FILE.
0018 C
0019 C *****
0020
0021
0022     INCLUDE 'ERDS\INCLUDE\FORMFLAGS.INC'
0023 Byte      Form_Interface_Record( 148)
0024 Integer*4  Iflags( 37)      ! Form flags
0025 C
0026 C          Type      Name      Field Type      Field Prompt String
0027 C
0028 Integer*4  I_VAR001      ! 1 Integer      Hour 01 - Elevation (3000 - 8000)
0029 Integer*4  I_VAR002      ! 2 Integer      Hour 01 - Temperature (-2 - +25)
0030 Integer*4  I_VAR003      ! 3 Integer      Hour 01 - Flow (200 - 1500)/ Def
0031 Integer*4  I_VAR004      ! 4 Integer      Hour 03 - Elevation (3000 - 8000)
0032 Integer*4  I_VAR005      ! 5 Integer      Hour 03 - Temperature (-2 - +25)
0033 Integer*4  I_VAR006      ! 6 Integer      Hour 03 - Flow (200 - 1500)/ Def
0034 Integer*4  I_VAR007      ! 7 Integer      Hour 05 - Elevation (3000 - 8000)
0035 Integer*4  I_VAR008      ! 8 Integer      Hour 05 - Temperature (-2 - +25)
0036 Integer*4  I_VAR009      ! 9 Integer      Hour 05 - Flow (200 - 1500)/ Def
0037 Integer*4  I_VAR010      ! 10 Integer     Hour 07 - Elevation (3000 - 8000)
0038 Integer*4  I_VAR011      ! 11 Integer     Hour 07 - Temperature (-2 - +25)
0039 Integer*4  I_VAR012      ! 12 Integer     Hour 07 - Flow (200 - 1500)/ Def
0040 Integer*4  I_VAR013      ! 13 Integer     Hour 09 - Elevation (3000 - 8000)
0041 Integer*4  I_VAR014      ! 14 Integer     Hour 09 - Temperature (-2 - +25)
0042 Integer*4  I_VAR015      ! 15 Integer     Hour 09 - Flow (200 - 1500)/ Def
0043 Integer*4  I_VAR016      ! 16 Integer     Hour 11 - Elevation (3000 - 8000)
0044 Integer*4  I_VAR017      ! 17 Integer     Hour 11 - Temperature (-2 - +25)
0045 Integer*4  I_VAR018      ! 18 Integer     Hour 11 - Flow (200 - 1500)/ Def
0046 Integer*4  I_VAR019      ! 19 Integer     Hour 13 - Elevation (3000 - 8000)
0047 Integer*4  I_VAR020      ! 20 Integer     Hour 13 - Temperature (-2 - +25)
0048 Integer*4  I_VAR021      ! 21 Integer     Hour 13 - Flow (200 - 1500)/ Def
0049 Integer*4  I_VAR022      ! 22 Integer     Hour 15 - Elevation (3000 - 8000)
0050 Integer*4  I_VAR023      ! 23 Integer     Hour 15 - Temperature (-2 - +25)
0051 Integer*4  I_VAR024      ! 24 Integer     Hour 15 - Flow (200 - 1500)/ Def
0052 Integer*4  I_VAR025      ! 25 Integer     Hour 17 - Elevation (3000 - 8000)
0053 Integer*4  I_VAR026      ! 26 Integer     Hour 17 - Temperature (-2 - +25)

```

```

0091 Integer*4 I_VAR027 ! 27 Integer Hour 17 - Flow (200 - 1500)/ Def
0092 Integer*4 I_VAR028 ! 28 Integer Hour 19 - Elevation (3000 - 0000)
0093 Integer*4 I_VAR029 ! 29 Integer Hour 19 - Temperature (-2 - +25)
0094 Integer*4 I_VAR030 ! 30 Integer Hour 19 - Flow (200 - 1500)/ Def
0095 Integer*4 I_VAR031 ! 31 Integer Hour 21 - Elevation (3000 - 0000)
0096 Integer*4 I_VAR032 ! 32 Integer Hour 21 - Temperature (-2 - +25)
0097 Integer*4 I_VAR033 ! 33 Integer Hour 21 - Flow (200 - 1500)/ Def
0098 Integer*4 I_VAR034 ! 34 Integer Hour 23 - Elevation (3000 - 0000)
0099 Integer*4 I_VAR035 ! 35 Integer Hour 23 - Temperature (-2 - +25)
0100 Integer*4 I_VAR036 ! 36 Integer Hour 23 - Flow (200 - 1500)/ Def
0101 Integer*4 I_VAR037 ! 37 Integer Date (800101 - 990101)/ Default =
0102
0103 INTEGER CLOUDS, YEAR, MONTH, DAY, HOUR, DATA(21),
0104 . I_VAR(36), DAYS_IN_MONTH(12), DAYS_IN_MONTH_LY(12),
0105 . check_hour(30)
0106
0107 CHARACTER*80 SKIP_RECORD
0108 CHARACTER*3 MONTH_NAME(12)
0109
0110 Equivalence (Form_Interface_Record, I_VAR001)
0111
0112 Common /Form_Record/ I_VAR001, I_VAR002, I_VAR003, I_VAR004,
0113 . I_VAR005, I_VAR006, I_VAR007, I_VAR008,
0114 . I_VAR009, I_VAR010, I_VAR011, I_VAR012,
0115 . I_VAR013, I_VAR014, I_VAR015, I_VAR016,
0116 . I_VAR017, I_VAR018, I_VAR019, I_VAR020,
0117 . I_VAR021, I_VAR022, I_VAR023, I_VAR024,
0118 . I_VAR025, I_VAR026, I_VAR027, I_VAR028,
0119 . I_VAR029, I_VAR030, I_VAR031, I_VAR032,
0120 . I_VAR033, I_VAR034, I_VAR035, I_VAR036,
0121 . I_VAR037
0122
0123 DATA DAYS_IN_MONTH/31, 28, 31, 30, 31, 30, 31, 31, 30, 31,
0124 . 30, 31/
0125
0126 DATA DAYS_IN_MONTH_LY/31, 29, 31, 30, 31, 30, 31, 31, 30,
0127 . 31, 30, 31/
0128
0129 DATA MONTH_NAME/'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN',
0130 . 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC'/
0131
0132
0133
0134 C..... OPEN THE DAILYFCOBS.DAT FILE
0135
0136
0137 OPEN(UNIT=10, FILE='[FCST.DAT]DAILYFCOBS.DAT', STATUS='OLD',
0138 . READONLY, IOSTAT=IER)
0139 IF(IER.NE.0) THEN
0140 WRITE (*,10) IER
0141 10 FORMAT(/5X,' Error Opening [FCST.DAT]DAILYFCOBS.DAT Code - ',
0142 . ,I4)
0143 STOP
0144 ENDIF

```

```

0145
0146
0147 C..... COUNT THE NUMBER OF RECORDS IN DAILYFCOBS.DAT FILE; SHOULD BE 24
0148
0149
0150     kount = 1
0151 300 read(10,'(11x,i2)',end=200) check_hour(kount)
0152     kount = kount + 1
0153     goto 300
0154 200     kount = kount - 1
0155     close unit = 10
0156     if(kount .ne. 24) then
0157         PRINT *, ' '
0158         PRINT *, ' '
0159         PRINT *, '          ***** WARNING ',
0160         .      '*****'
0161         PRINT *, ' '
0162         PRINT *, '      Incorrect number of hours in [FCST.DAT]',
0163         .      'DAILYFCOBS.DAT.'
0164         PRINT *, ' '
0165         PRINT *, ' '
0166         print *, ' ** DO NOT ** Run SFC or RIVER Until the',
0167         .      ' DAILYFCOBS.DAT File has been Corrected'
0168         STOP
0169
0170     endif
0171
0172
0173 C..... HOURS OK NOW CHECK FOR HOURS OUT OF SEQUENCE
0174
0175
0176     do i=1, 24
0177         if(i .ne. check_hour(i)) then
0178             PRINT *, ' '
0179             PRINT *, ' '
0180             PRINT *, '          ***** WARNING ',
0181             .      '*****'
0182             PRINT *, ' '
0183             PRINT *, '      Hours in [FCST.DAT]DAILYFCOBS.DAT',
0184             .      ' Out of Sequence'
0185             PRINT *, ' '
0186             PRINT *, ' '
0187             print *, ' ** DO NOT ** Run SFC or RIVER Until the',
0188             .      ' DAILYFCOBS.DAT File has been Corrected'
0189             STOP
0190         endif
0191     enddo
0192
0193
0194
0195 C..... OPEN RIVER_DATE.DAT FROM PREVIOUS_DATE SUB-DIRECTORY REDPEN
0196 C..... DAILYFCOBS.DAT FILE
0197
0198

```

```

0199
0200 OPEN(UNIT=10, FILE='[FCST.DAT]DAILYSFCS.DAT', STATUS='OLD',
0201 READONLY, IOSTAT=IER)
0202 IF(IER.NE.0) THEN
0203 WRITE(*,101) IER
0204 101 FORMAT(//5X,' Error Opening [FCST.DAT]DAILYSFCS.DAT Code - ',
0205 ,I4)
0206 STOP
0207 ENDIF
0208
0209 OPEN(UNIT=11, FILE='[FCST.PREVIOUS_DATE]RIVER_DATE.DAT',
0210 STATUS='OLD', IOSTAT=IER)
0211 IF(IER.NE.0) THEN
0212 WRITE(*,20) IER
0213 20 FORMAT(//5X,' Error Opening [FCST.PREVIOUS_DATE]RIVER_DATE.DAT
0214 Code - ',I4)
0215 STOP
0216 ENDIF
0217
0218
0219 READ(10,90,END=99) CLOUDS, YEAR, MONTH, DAY, HOUR,
0220 (DATA(I),I=1,21)
0221 90 FORMAT(I5,4I2,I3,I4,I3,I1,I2,I3,I2,I4,I3,2I2,I4,3I3,4I2,I1,I3)
0222 READ(11,'(4I2)') I_YEAR, I_MONTH, I_DAY, I_HOUR
0223
0224 IF(I_YEAR.EQ.YEAR.AND. I_MONTH.EQ.MONTH.AND.
0225 I_DAY.EQ.DAY.AND. I_HOUR.EQ.HOUR) THEN
0226
0227 OPEN(UNIT=12, FILE='[FCST.DAT]SFCOBS'//MONTH_NAME(I_MONTH)
0228 //''.DAT', STATUS='OLD', ACCESS='APPEND',
0229 CARRIAGECONTROL='LIST', IOSTAT=IER)
0230 IF(IER.NE.0) THEN
0231 WRITE(*,30) IER
0232 30 FORMAT(//5X,' Error Opening Monthly Surface Obs. File in
0233 [FCST.DAT] Code - ',I4)
0234 STOP
0235 ENDIF
0236
0237
0238 C..... BUILD DATE
0239
0240
0241 I_VAR037 = (YEAR*10000) + (MONTH*100) + DAY
0242
0243
0244 C..... PREVENT USER FROM ENTERING DATE AND FORCE START CURSOR AT
0245 C..... FIRST VARIABLE
0246
0247
0248 IFLAGS(37) = FWS_NO_WRITE
0249 IFLAGS(1) = FWS_FORCE_START
0250
0251
0252 C..... MAKE CALLS TO FORM SUBROUTINES TO BUILD FORM AND ALLOW

```



```

0253 C..... FOR INPUT
0254
0255
0256 CALL SUBPROCESS_OPEN_IO
0257 CALL SELECT_ANSI ! put terminal in ANSI mode
0258 Call FM_Form_Open (20, 'HMSFORMS.FRM', IError)
0259 Call FM_Form_Input ('RIVER', Form_Interface_Record,
0260 Iflags, Ierror, User_Check, *999)
0261 Call FM_Form_Close (IError)
0262 CALL SUBPROCESS_CLOSE_IO
0263
0264
0265 C..... PUT VARIABLES FROM FORM INPUT INTO AN ARRAY
0266
0267 I_VAR(1) = I_VAR001
0268 I_VAR(2) = I_VAR002
0269 I_VAR(3) = I_VAR003
0270 I_VAR(4) = I_VAR004
0271 I_VAR(5) = I_VAR005
0272 I_VAR(6) = I_VAR006
0273 I_VAR(7) = I_VAR007
0274 I_VAR(8) = I_VAR008
0275 I_VAR(9) = I_VAR009
0276 I_VAR(10) = I_VAR010
0277 I_VAR(11) = I_VAR011
0278 I_VAR(12) = I_VAR012
0279 I_VAR(13) = I_VAR013
0280 I_VAR(14) = I_VAR014
0281 I_VAR(15) = I_VAR015
0282 I_VAR(16) = I_VAR016
0283 I_VAR(17) = I_VAR017
0284 I_VAR(18) = I_VAR018
0285 I_VAR(19) = I_VAR019
0286 I_VAR(20) = I_VAR020
0287 I_VAR(21) = I_VAR021
0288 I_VAR(22) = I_VAR022
0289 I_VAR(23) = I_VAR023
0290 I_VAR(24) = I_VAR024
0291 I_VAR(25) = I_VAR025
0292 I_VAR(26) = I_VAR026
0293 I_VAR(27) = I_VAR027
0294 I_VAR(28) = I_VAR028
0295 I_VAR(29) = I_VAR029
0296 I_VAR(30) = I_VAR030
0297 I_VAR(31) = I_VAR031
0298 I_VAR(32) = I_VAR032
0299 I_VAR(33) = I_VAR033
0300 I_VAR(34) = I_VAR034
0301 I_VAR(35) = I_VAR035
0302 I_VAR(36) = I_VAR036
0303
0304 C..... APPEND RIVER DATA TO DAILYSCOBs DATA THAN WRITE TO THE
0305 C..... MONTHLY FILE AND READ NEXT RECORD OF DAILYSCOBs DATA
0306 C..... TO SKIP A RECORD AND WRITE THAT RECORD TO THE MONTHLY FILE

```

```

0307 C..... READ ANOTHER RECORD AND REPEAT
0308
0309 DO I=1, 36, 3
0310
0311 WRITE(12,11) CLOUDS, YEAR, MONTH, DAY, HOUR,
0312 (DATA(J),J=1,21), I_VAR(I), I_VAR(I+1),
0313 I_VAR(I+2)
0314 11 FORMAT(I6,4I2,I3,I4,I3,I1,I2,I3,I2,I4,I3,2I2,I4,
0315 3I3,4I2,I1,I3,1X,I4,I3,I4)
0316 READ (10,'
0317 WRITE(12,'(A00)') SKIP_RECORD
0318 READ (10,90,END=99) CLOUDS, YEAR, MONTH, DAY, HOUR,
0319 (DATA(J),J=1,21)
0320
0321 ENDDO
0322
0323 99 PRINT *, '
0324 PRINT *, '
0325 PRINT *, ' Successfully Appended River Data to the',
0326 ' Monthly Surface Observation File'
0327
0328
0329 C..... CHECK FOR END OF MONTH IF SO OPEN NEW MONTHLY FILE ALSO OPEN NEW
0330 C..... DAILY FILE IF NOT JUST OPEN NEW DAILY FILE
0331
0332
0333 IF(MOD(I_YEAR,4) .EQ. 0) THEN
0334
0335 IF(I_DAY .EQ. DAYS_IN_MONTH_LY(I_MONTH)) THEN
0336 CLOSE UNIT=10
0337 IF(I_MONTH .EQ. 12) THEN
0338 NEXT_MONTH = 1
0339 ELSE
0340 NEXT_MONTH = I_MONTH + 1
0341 ENDF
0342 OPEN(UNIT=13, FILE='[FCST.DAT]SFCOBS'//
0343 MONTH_NAME(NEXT_MONTH)//'.DAT', STATUS='NEW',
0344 CARRIAGECONTROL='LIST', IOSTAT=IER)
0345 IF(IER .NE. 0) WRITE (*,40) IER
0346 40 FORMAT(//6X, ' Error Opening New Monthly Surface Obs. '
0347 , 'File in [FCST.DAT] Code - ',I4)
0348
0349 OPEN(UNIT=14, FILE='[FCST.DAT]DAILYFCOBS.DAT',
0350 STATUS='NEW', CARRIAGECONTROL='LIST', IOSTAT=IER)
0351 IF(IER .NE. 0) WRITE (*,50) IER
0352 50 FORMAT(//6X, ' Error Opening New Daily Surface Obs. '
0353 , 'File in [FCST.DAT] Code - ',I4)
0354 WRITE(*,12)
0355 12 FORMAT(//10X, 'Opened New Monthly and Daily Surface',
0356 'Obs. Files'//)
0357 ELSE
0358 CLOSE UNIT=10
0359 OPEN(UNIT=14, FILE='[FCST.DAT]DAILYFCOBS.DAT',
0360 STATUS='NEW', CARRIAGECONTROL='LIST', IOSTAT=IER)

```

```

0361             IF(IER .NE. 0) WRITE (*,50) IER
0362             WRITE(*,13)
0363 13          FORMAT(//10X,'Opened New Daily Surface Obs. Files'//)
0364             ENDIF
0365
0366             ELSE
0367
0368             IF(I_DAY .EQ. DAYS_IN_MONTH(I_MONTH)) THEN
0369                 CLOSE UNIT=10
0370                 IF(I_MONTH .EQ. 12) THEN
0371                     NEXT_MONTH = 1
0372                 ELSE
0373                     NEXT_MONTH = I_MONTH + 1
0374                 ENDIF
0375                 OPEN(UNIT=13, FILE='[FCST.DAT]SFCOBS'//
0376                     MONTH_NAME(NEXT_MONTH)//'.DAT', STATUS='NEW',
0377                     CARRIAGECONTROL='LIST', IOSTAT=IER)
0378                 IF(IER .NE. 0) WRITE (*,40) IER
0379
0380                 OPEN(UNIT=14, FILE='[FCST.DAT]DAILY_SFCOBS.DAT',
0381                     STATUS='NEW', CARRIAGECONTROL='LIST', IOSTAT=IER)
0382                 IF(IER .NE. 0) WRITE (*,50) IER
0383                 WRITE(*,12)
0384             ELSE
0385                 CLOSE UNIT=10
0386                 OPEN(UNIT=14, FILE='[FCST.DAT]DAILY_SFCOBS.DAT',
0387                     STATUS='NEW', CARRIAGECONTROL='LIST', IOSTAT=IER)
0388                 IF(IER .NE. 0) WRITE (*,50) IER
0389                 WRITE(*,13)
0390             ENDIF
0391
0392             ENDIF
0393
0394
0395 C..... INCREMENT DATE FOR NEXT SET OF RIVER DATA. HOUR REMAINS #1
0396
0397
0398             IF (I_DAY.EQ.31 .AND. I_MONTH.EQ.12) THEN
0399                 I_YEAR = I_YEAR + 1
0400                 I_MONTH = 1
0401                 I_DAY = 1
0402             ELSE
0403
0404                 IF(MOD(I_YEAR,4) .EQ. 0) THEN
0405                     IF(I_DAY .EQ. DAYS_IN_MONTH_LY(I_MONTH)) THEN
0406                         I_MONTH = I_MONTH + 1
0407                         I_DAY = 1
0408                     ELSE
0409                         I_DAY = I_DAY + 1
0410                     ENDIF
0411                 ELSE
0412                     IF(I_DAY .EQ. DAYS_IN_MONTH(I_MONTH)) THEN
0413                         I_MONTH = I_MONTH + 1
0414                         I_DAY = 1

```

```

0415             ELSE
0416                 I_DAY = I_DAY + 1
0417             ENDIF
0418             ENDIF
0419         ENDIF
0420
0421
0422     C..... WRITE NEXT DATE TO BE ENTERED TO RIVER_DATE
0423
0424
0425         BACKSPACE(UNIT=11, IOSTAT=IER)
0426         IF(IER .NE. 0) WRITE (*,80) IER
0427     80      FORMAT(/5X,' Error During Backspace Code - ',I4/
0428             5X,'Check [FCST.PREVIOUS_DATE] RIVER_DATE.DAT',
0429             ' File')
0430         WRITE(11,'(4I2)') I_YEAR, I_MONTH, I_DAY, I_HOUR
0431
0432     ELSE
0433
0434         PRINT *,' '
0435         PRINT *,' '
0436         PRINT *,'   Error matching dates in first record of '
0437         PRINT *,'   File'
0438         PRINT *,'   ** DO NOT ** run the SFC or RIVER programs'
0439         PRINT *,'   until Ora or Gregg have been contacted'
0440     ENDIF
0441     GO TO 100
0442 990 PRINT *,' '
0443     PRINT *,' '
0444     PRINT *,' Error during form input mode. * DO NOT * run SFC or'
0445     PRINT *,' RIVER until Ora or Gregg have been contacted.'
0446 100 Call Exit
0447     END

```

PROGRAM SECTIONS

Name Bytes Attributes

0 \$CODE	2944	PIC CON REL LCL SHR EXE	RD NOWRT LONG
1 \$PDATA	1398	PIC CON REL LCL SHR NOEXE	RD NOWRT LONG
2 \$LOCAL	1318	PIC CON REL LCL NOSHR NOEXE	RD WRT LONG
3 FORM_RECORD	148	PIC OVR REL QBL SHR NOEXE	RD WRT LONG

Total Space Allocated 5800

ENTRY POINTS

Address	Type	Name	References
---------	------	------	------------

0-00000000		RIVERS\$MAIN	
------------	--	--------------	--

VARIABLES

Address	Type	Name	Attributes		References					
**	R+4	CLOSEUNIT			155=	336=	358=	369=	385=	
2-000002C4	I+4	CLOUDS			183	219=	311	318=		
2-000002D8	I+4	DAY			183	219=	224	241	311	318=
2-000002D4	I+4	HOUR			183	219=	224	311	318=	
**	I+4	I			176=	177(2)	219(2)=	309=	311(3)	
2-000002D8	I+4	IER			137=A	139	140	200=A	202	203
209=A		211	212	227=A	230	231				
342=A		345(2)	349=A	351(2)	359=A	361(2)				
375=A		378(2)	386=A	382(2)	388=A	388(2)				
425=		426(2)								
2-000002F0	I+4	IERROR			259A					
2-000002EC	I+4	IOERROR			250A	261A				
2-000002E4	I+4	I_DAY			222=	224	335	368	398	401=
405		407=	409(2)=	412	414=	416(2)=				
430										
2-000002E8	I+4	I_HOUR			222=	224	430			
2-000002E0	I+4	I_MONTH			222=	224	227	335	337	340
300		370	373	398	400=	405				
406(2)=		412	413(2)=	430						
3-00000000	I+4	I_VAR001			COMM EQUIV	65	110	112	267	
3-00000004	I+4	I_VAR002			COMM EQUIV	66	112	268		
3-00000008	I+4	I_VAR003			COMM EQUIV	67	112	269		
3-0000000C	I+4	I_VAR004			COMM EQUIV	68	112	270		
3-00000010	I+4	I_VAR005			COMM EQUIV	69	112	271		
3-00000014	I+4	I_VAR006			COMM EQUIV	70	112	272		
3-00000018	I+4	I_VAR007			COMM EQUIV	71	112	273		
3-0000001C	I+4	I_VAR008			COMM EQUIV	72	112	274		
3-00000020	I+4	I_VAR009			COMM EQUIV	73	112	275		
3-00000024	I+4	I_VAR010			COMM EQUIV	74	112	276		
3-00000028	I+4	I_VAR011			COMM EQUIV	75	112	277		
3-0000002C	I+4	I_VAR012			COMM EQUIV	76	112	278		
3-00000030	I+4	I_VAR013			COMM EQUIV	77	112	279		
3-00000034	I+4	I_VAR014			COMM EQUIV	78	112	280		
3-00000038	I+4	I_VAR015			COMM EQUIV	79	112	281		
3-0000003C	I+4	I_VAR016			COMM EQUIV	80	112	282		
3-00000040	I+4	I_VAR017			COMM EQUIV	81	112	283		
3-00000044	I+4	I_VAR018			COMM EQUIV	82	112	284		
3-00000048	I+4	I_VAR019			COMM EQUIV	83	112	285		
3-0000004C	I+4	I_VAR020			COMM EQUIV	84	112	286		
3-00000050	I+4	I_VAR021			COMM EQUIV	85	112	287		
3-00000054	I+4	I_VAR022			COMM EQUIV	86	112	288		
3-00000058	I+4	I_VAR023			COMM EQUIV	87	112	289		
3-0000005C	I+4	I_VAR024			COMM EQUIV	88	112	290		
3-00000060	I+4	I_VAR025			COMM EQUIV	89	112	291		

3-00000064	I+4	I_VAR026	COMM EQUIV	90	112	292			
3-00000068	I+4	I_VAR027	COMM EQUIV	91	112	293			
3-0000006C	I+4	I_VAR028	COMM EQUIV	92	112	294			
3-00000070	I+4	I_VAR029	COMM EQUIV	93	112	295			
3-00000074	I+4	I_VAR030	COMM EQUIV	94	112	296			
3-00000078	I+4	I_VAR031	COMM EQUIV	95	112	297			
3-0000007C	I+4	I_VAR032	COMM EQUIV	96	112	298			
3-00000080	I+4	I_VAR033	COMM EQUIV	97	112	299			
3-00000084	I+4	I_VAR034	COMM EQUIV	98	112	300			
3-00000088	I+4	I_VAR035	COMM EQUIV	99	112	301			
3-0000008C	I+4	I_VAR036	COMM EQUIV	100	112	302			
3-00000090	I+4	I_VAR037	COMM EQUIV	101	112	241=			
2-000002DC	I+4	I_YEAR		222=	224	333	399(2)=	404	430
**	I+4	J		311(2)=	318(2)=				
**	I+4	KOUNT		150=	151	152(2)=	154(2)=	156	
2-000002CC	I+4	MONTH		103	219=	224	241	311	318=
**	I+4	NEXT_MONTH		338=	340=	342	371=	373=	375
2-00000274	CHAR	SKIP_RECORD		107	318=	317			
2-000002F4	R+4	USER_CHECK		259A					
2-000002C8	I+4	YEAR		103	219=	224	241	311	318=

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References
2-000001D8	I+4	CHECK_HOUR		120	(30)	103 151= 177
2-00000094	I+4	DATA		84	(21)	103 219= 311
318=						
2-00000178	I+4	DAYS_IN_MONTH		48	(12)	103 1230 368
412						
2-000001A8	I+4	DAYS_IN_MONTH_LY		48	(12)	103 1280 335
405						
3-00000000	L+1	FORM_INTERFACE_RECORD	COMM EQUIV	140	(140)	60 110 259A
2-00000000	I+4	IFLAGS		140	(37)	61 248= 249=
259A						
2-000000E8	I+4	I_VAR		144	(36)	103 267= 268=
269=	270=	271=	272=	273=		
274=	275=	276=	277=	278=		
279=	280=	281=	282=	283=		
284=	285=	286=	287=	288=		
289=	290=	291=	292=	293=		
294=	295=	296=	297=	298=		
299=	300=	301=	302=	311(3)		
2-00000250	CHAR	MONTH_NAME		36	(12)	108 1290 227
342	375					

PARAMETER CONSTANTS

Type	Name	References				
I+4	FW\$ _FIELD_BOLD	48	58			
I+4	FW\$ _FIELD_UNDERLINE	48	58			
I+4	FW\$ _FORCE_START	48	58	249		
I+4	FW\$ _NO_ECHO	48	58			
I+4	FW\$ _NO_SEE	48	58			
I+4	FW\$ _NO_WRITE	48	58	248		
I+4	FW\$ _SET_ERROR	48	58			
I+4	FW\$ _SET_EXCEPTION	48	58			
I+4	FW\$ _USER_CHECK	48	58			

LABELS

Address	Label	References				
1-000002A7	10'	140	141			
1-000003D7	11'	311	314			
1-000004A5	12'	354	355	383		
1-000004DE	13'	362	363	389		
1-0000031E	20'	212	213			
1-00000390	36'	231	232			
1-0000040D	40'	345	346	378		
1-0000045A	50'	351	352	381	382	388
1-0000050A	80'	426	427			
1-00000306	90'	219	221	318		
0-00000610	99	219	316	318	323	
0-00000679	100	441	448			
1-000002E5	101'	203	204			
0-0000007C	200	151	154			
0-00000048	300	151	153			
0-000006F0	990	259A	442			

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References						
	FW_FORM_CLOSE	261						
	FW_FORM_INPUT	259						
	FW_FORM_OPEN	258						
	FOR\$EXIT	446						
	FOR\$OPEN	137	260	269	227	342	349	
		359	375	380	386			

SELECT_ANSI	257
SUBPROCESS_CLOSE_IO	252
SUBPROCESS_OPEN_IO	256

```

+-----+
|          KEY TO REFERENCE FLAGS          |
|  =  - Value Modified                     |
|  #  - Defining Reference                 |
|  A  - Actual Argument, possibly modified |
|  D  - Data Initialization                 |
|  (n) - Number of occurrences on line     |
+-----+

```

COMMAND QUALIFIERS

```

FOR/LIS/CROSS/CHECK=ALL RIVER

/CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
/CONTINUATIONS=19 /CROSS_REFERENCE /NO_LINES /NOEXTEND_SOURCE /F77
/NOG_FLOATING /I4 /NOMACHINE_CODE /OPTIMIZE

```

COMPILATION STATISTICS

```

Run Time:      14.13 seconds
Elapsed Time:  19.79 seconds
Page Faults:   888
Dynamic Memory: 553 pages

```


DISTRIBUTION

No. of
Copies

OFFSITE

2 DOE/Office of Scientific and Technical Information

ONSITE

DOE Richland Operations Office

J. J. Sutey

29 Pacific Northwest Laboratory

G. L. Andrews (5)
K. A. Borgeson
J. W. Buck
D. W. Dragnich
O. P. Gifford
M. J. Graham
J. M. Hales
P. C. Hays
D. J. Hoitink
D. J. Newland
W. T. Pennell
P. J. Perrault
E. R. Preston
J. V. Ramsdell
G. N. Reinecke
M. N. Schwartz
L. L. Wendell
R. E. Wildung
Publishing Coordination (2)
Technical Report Files (5)