50

Hanford Meteorological Station Computer Codes Volume 3 - The TANK Computer Code

**J. W. Buck
G. L. Andrews**

September 1987

**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 of 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 3 - The TANK Computer Code

J. W. Buck
G. L. Andrews

September 1987

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 third in a series of volumes that describe 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 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

At the end of each graveyard shift the Hanford Meteorological Station (HMS), operated by Pacific Northwest Laboratory, issues a forecast of eight hourly average wind speeds and wind gusts for the 50-ft level. The Hanford Waste Management crew uses this forecast, called the tank farm forecast, to schedule daily work loads. The forecast covers an 8-hour period [0800-1600 during Pacific Standard Time (PST) or 0700-1500 during Pacific Daylight Time (PDT)] and the day-shift forecaster may modify the tank farm forecast to reflect changing wind conditions. The TANK computer code is used to archive these forecasts and apply quality assurance checks to the forecast data. The code accesses an input file, which contains the date of the previous forecast, and an output file, which contains tank farm forecasts for the current month. The program includes a data entry form consisting of 12 fields that must be filled in by the user. The information entered on the form is appended to the monthly forecast file, which provides an archive for the tank farm forecasts. This volume describes the implementation and operation of the TANK computer code at the HMS.

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 Van Ramsdell for his helpful suggestions and ideas.

CONTENTS

PREFACE	iii
SUMMARY	v
ACKNOWLEDGMENTS	vii
INTRODUCTION	1
THE TANK COMPUTER CODE	3
PROGRAM OVERVIEW	3
USER'S GUIDE FOR THE TANK COMPUTER CODE	3
DESCRIPTION OF INPUT	6
TROUBLE-SHOOTING THE TANK COMPUTER CODE	11
TERMINAL PROBLEMS/FAILURES.	11
SYSTEM PROBLEMS/FAILURES	11
PROGRAM PROBLEMS/FAILURES	12
APPENDIX A - PROGRAM DESCRIPTION AND FLOWCHART	A.1
APPENDIX B - TANK 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 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).

The HMS staff issue 8-hour tank farm forecasts in support of operations and research for the U.S. Department of Energy at Richland (DOE-RL) and associated contractors within the Hanford Site boundaries. These forecasts are issued at the end of each graveyard shift, and are used mostly by the Westinghouse waste management crews to schedule their daily work loads. If wind conditions are forecast to include gusts over 15 mph, all outside waste management operations must stop. Conversely, if wind gusts are forecast to stay at or under 15 mph, outside waste management operations may continue. The forecasts are issued for the day shift hours [0800-1600 during Pacific Standard Time (PST) or 0700-1500 during Pacific Daylight Time (PDT)], and consist of eight hourly average wind speeds for the 50-ft level. If the hourly average wind speed is forecast to be less than or equal to 15 mph the forecast may include a gust code, which indicates that gusts over 15 mph are expected with the hourly average wind speed. Forecasters use the TANK computer code to archive and apply quality assurance checks to the tank farm forecasts. The archive is maintained in disk files on the HMS computer system (DEC VAX 11/750).

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

THE TANK COMPUTER CODE

The TANK computer code provides a method for archiving data obtained from the tank farm forecasts. The code has replaced manual entry of data via computer cards and a keypunch. The TANK code was written, compiled, and linked on the DEC VAX 11/750 at the HMS and is executed once a day during the day shift.

PROGRAM OVERVIEW

The TANK computer code accesses one input file and one output file. The input file contains the previous forecast's date, and the output file consists of tank farm forecasts for the current month. The convention used in naming the monthly forecast files is TNKFCSTxxx.DAT where the xxx is the three-character abbreviation of the current month (e.g., AUG for August). (Refer to Appendix A for more information about input/output operations.)

The program begins by displaying a data entry form on the terminal screen (Fig. 1). The length of each field is indicated by the number of X's next to the field name. The program fills in the date field by accessing an input file to obtain the previous forecast's date. You are then prompted to enter data for 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 forecast is appended to the current monthly tank farm forecast file.

USER'S GUIDE FOR THE TANK COMPUTER CODE

This section describes how to invoke the TANK 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 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 TANK code. To determine the terminal status, simply type SHOW TERMINAL and depress the RETURN key. A list of terminal characteristics appears on

HMS TANK FARM WIND FORECAST INPUT FORM			
DATE: xxxxxx	GRAVE FCSTR: x	DAY FCSTR: x	
SPD HOUR 1: xxx	SPD HOUR 2: xxx	SPD HOUR 3: xxx	SPD HOUR 4: xxx
SPD HOUR 5: xxx	SPD HOUR 6: xxx	SPD HOUR 7: xxx	SPD HOUR 8: xxx
SPD HOUR 9: xxx			
<p style="text-align: center;">***** NOTE *****</p> <p>During PST fill in hours 2 - 9 and hit default for hour 1</p> <p>During PDT fill in hours 1 - 8 and hit default for hour 9</p>			

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

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 TANK computer code, filling in the data entry form, and terminating the program. These steps are listed below and will be discussed in detail in the following paragraphs:

- log in to the HMS computer (need the FCST password)
- type RUN TANK, 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 then 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 into the FCST account, you are placed in the [FCST] directory and can immediately invoke the TANK computer code.

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

The data entry form appears on the screen as shown in Figure 1 with the cursor in field two (GRAVE FCSTR). The date field (first field) is filled in by the program. A prompt appears on the screen at the bottom of the data entry form for each field. Included with each prompt is a default value and a valid range or list of matching 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 a value is entered in all the fields, you should check the fields for mistakes. If corrections are needed use the up, down, right, and left arrow keys ($\uparrow, \downarrow, \rightarrow, \leftarrow$) 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 TANK 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 file (e.g., TNKFCSTxxx.DAT). A prompt appears just below the information asking if you would care to make any corrections. If no corrections are needed, enter N or n and the program appends the data to the monthly file. If corrections are needed, enter Y or y and the program returns to the data entry form mode. At this time, the 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. When all corrections have been made, you can exit the form by depressing the EXIT key. You can exit the data entry form from any field; it is not necessary to move to the last field.

The TANK program is terminated by answering n(o) or N(o) 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 files for the tank farm forecasts are located in the [FCST.DAT] directory. Each file contains all of the tank farm forecasts for any single month.

DESCRIPTION OF INPUT

The data entry form consists of 12 fields; integer values must be provided for 11 of these fields. The program provides the value for the first field, which is the date field. When you move the cursor to one of the other 11 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 "User's Guide" section for information on how to input the default value). There are nine fields for entering eight 1-hour wind forecasts. A message at the bottom of the form explains how to enter the data during PST or PDT (see Fig. 1). Table 1 describes each of the input fields.

The following list describes each input field, the associated range or list of matching values from which the input must be selected, and the default value.

TABLE 1. Reference Guide to Input Fields

<u>Input Field</u>	<u>Description</u>	<u>Range or List/Default</u>
DATE	YYMMDD	860101-990101 / 999999
GRAVE FCSTR	graveyard forecaster id	1 - 9 / 0
DAY FCSTR	day forecaster id	1 - 9 / 0
SPEED 1	hourly wind spd (mph) & gust code	000 - 115 / 999
SPEED 2	hourly wind spd (mph) & gust code	000 - 115 / 999
SPEED 3	hourly wind spd (mph) & gust code	000 - 115 / 999
SPEED 4	hourly wind spd (mph) & gust code	000 - 115 / 999
SPEED 5	hourly wind spd (mph) & gust code	000 - 115 / 999
SPEED 6	hourly wind spd (mph) & gust code	000 - 115 / 999
SPEED 7	hourly wind spd (mph) & gust code	000 - 115 / 999
SPEED 8	hourly wind spd (mph) & gust code	000 - 115 / 999
SPEED 9	hourly wind spd (mph) & gust code	000 - 115 / 999

- 1) 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. An example of a date is 870421 which 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 for the forecast.
- 2) GRAVE FCSTR - This field contains the graveyard forecaster's identification (ID) code. Each forecaster is assigned an ID code that he must enter in the data entry form to indicate who issued the forecast. The range of this code is from 1 through 9 with a default value of 0. The 0 default value indicates an ID has not been assigned.
- 3) DAY FCSTR - This field contains the day forecaster's ID code. Each forecaster is assigned an ID code that he must enter in the data entry form to indicate who issued the forecast. The range of this

code is from 1 through 9 with a default value of 0. The 0 default value indicates an IO code has not been assigned.

- 4) SPEED 1 - The input value for this field includes the forecast hourly wind speed and a gust code for the first hour (0800-0900 PDT) of the tank farm forecast period. The code for this field consists of three digits. The first digit is a gust code that indicates if gusts greater than or equal to 15 mph are expected when the hourly average wind speed is predicted to be 15 mph or less. The values are 1 or 0; 1 indicates gusts will occur and 0 indicates gusts will not occur. The second and third digit indicate the hourly average wind speed in whole miles-per-hour. If the hourly average wind speed is predicted to be greater than 15 mph a gust code of 0 is entered. For example, an entry of 020 indicates the hourly average wind speed is expected to be 20 mph with no gusts. An entry of 110 indicates the hourly average wind speed is expected to be 10 mph with gusts over 15 mph. The valid range is 000 through 115. The default for this field is 999, which indicates wind speed and gusts were not forecast for this hour. This field is filled in during PDT; it is not filled in during PST.
- 5) SPEED 2 - The input value for this field includes the forecast hourly wind speed and a gust code for the second hour (0700-0800 PST or 0900-1000 PDT) of the tank farm forecast period. The explanation for this field is the same as that for field 4, except this field is filled in during both PDT and PST.
- 6) SPEED 3 - The input value for this field includes the forecast hourly wind speed and a gust code for the third hour (0800-0900 PST or 1000-1100 PDT) of the tank farm forecast period. The explanation for this field is the same as that for field 4, except this field is filled in during both PDT and PST.
- 7) SPEED 4 - The input value for this field includes the forecast hourly wind speed and a gust code for the fourth hour (0900-1000 PST or 1100-1200 PDT) of the tank farm forecast period. The explanation

for this field is the same as that for field 4, except this field is filled in during both PDT and PST.

- 8) SPEED 5 - The input value for this field includes the forecast hourly wind speed and a gust code for the fifth hour (1000-1100 PST or 1200-1300 PDT) of the tank farm forecast period. The explanation for this field is the same as that for field 4, except this field is filled in during both PDT and PST.
- 9) SPEED 6 - The input value for this field includes the forecast hourly wind speed and a gust code for the sixth hour (1100-1200 PST or 1300-1400 PDT) of the tank farm forecast period. The explanation for this field is the same as that for field 4, except this field is filled in during both PDT and PST.
- 10) SPEED 7 - The input value for this field includes the forecast hourly wind speed and a gust code for the seventh hour (1200-1300 PST or 1400-1500 PDT) of the tank farm forecast period. The explanation for this field is the same as that for field 4, except this field is filled in during both PDT and PST.
- 11) SPEED 8 - The input value for this field includes the forecast hourly wind speed and a gust code for the eighth hour (1300-1400 PST or 1500-1600 PDT) of the tank farm forecast period. The explanation for this field is the same as that for field 4, except this field is filled in during both PDT and PST.
- 12) SPEED 9 - The input value for this field includes the forecast hourly wind speed and a gust code for the ninth hour (1400-1500 PST) of the tank farm forecast period. The explanation for this field is the same as that for field 4, except this field is filled in during PST and not during PDT.

TROUBLE-SHOOTING THE TANK COMPUTER CODE

This section provides information that may help resolve problems that occur when operating the TANK 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 discuss problems or failures that may occur when using the TANK computer code and the suggested recovery procedures. If a series of recovery actions are suggested, you should try them in sequence until the problem or failure is resolved. Remember that the date is incremented each time you execute the TANK program. Therefore, you may need to edit the [FCST.PREVIOUS_DATE]TNKFCST_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 the 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 TANK.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]TANK.EXE [FCST]TANK.EXE. Try running the program again.

Problem: Input file ([FCST.PREVIOUS_DATE]TNKFCST_DATE.DAT) or output file ([FCST.DAT]TNKFCSTxxx.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 TANK.EXE to the [FCST] directory (see above for directions). Try running the code again.

APPENDIX A

PROGRAM DESCRIPTION AND FLOWCHART

APPENDIX A

PROGRAM DESCRIPTION AND FLOWCHART

This discussion describes the operations and logical flow of the TANK computer code. A flowchart indicating the general operations and logical sequence of the TANK 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 that 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 program begins by opening the file `TNKFCST_DATE.DAT`, which contains one record with the previous forecast's date. The program then reads the data. The `TNKFCST_DATE.DAT` file resides in the `[FCST.PREVIOUS_DATE]` sub-directory. The month from that date is then used to open the file `TNKFCSTxxx.DAT` (where xxx is the three-character abbreviation for the current month), which contains the tank farm forecast data for the current month. The `TNKFCSTxxx.DAT` file resides in the `[FCST.DAT]` sub-directory. The date is then incremented and entered into the `TNKFCST_DATE.DAT` file. If the incremented date crosses over into the next month, a new `TNKFCSTxxx.DAT` file is opened (where xxx will then reflect the new month). At this point the code calls several subroutines that control and display a data entry form at your terminal.

The TANK 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 forecast variable. Each field is displayed in reverse video to highlight the field length. The fields are initially blank. Prompts will appear at the bottom of the form for an input value to describe the data required for the current field. The prompt consists of a brief description of the field, the valid range or list of values from which the input must be selected, and a default value. A valid entry must be made in the current field 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 and the cursor remains in that field. When this occurs, the prompt reappears.

After entering all the input data, the data entry process is terminated by depressing the EXIT key on the Tektronix 4107 terminal. The screen is then cleared, and the input data are displayed at the terminal in the data format used in the TNKFCSTxxx.DAT file. You are then asked if any corrections are needed before the record is appended to the current monthly file. If no corrections are needed, answer "no" to the "need corrections" prompt and the data are appended to the TNKFCSTxxx.DAT file. A message is displayed on the screen indicating the forecast has been appended to the current monthly file. The program terminates when the message FORTRAN STOP appears on the terminal. If corrections are needed, answer "yes" to the "need corrections" prompt and the code returns to the data entry form. The input values entered by you are displayed. You can move to the appropriate input field(s) (refer to the "User's Guide" section for information on how to move the cursor within the form) and change the value(s). Once all the value(s) are corrected, the data record is again displayed on the screen and you are given another opportunity to correct any mistakes.

APPENDIX B

TANK SOURCE CODE LISTING

21-Jul-1987 08:44:23VAX FORTRAN V4.5-230 9-Apr-1987
15:32:54DUA0:[FCST.DATA_ENTRY]TANK.FOR;7

```

0001 C *****
0002 C
0003 C      PROGRAM: TANK.FOR
0004 C
0005 C      PROGRAMMER: GREGG ANDREWS
0006 C
0007 C      FUNCTION: THIS PROGRAM WAS DEVELOPED TO FACILITATE THE ARCHIVING
0008 C                  OF EIGHT HOUR TANK FARM FORECASTS. THE PROGRAM ACCESSES
0009 C                  A PREVIOUS_DATE FILE TO CHECK WHEN THE LAST DATA ENTRY
0010 C                  WAS THEN INCREMENTS THE DATE AND MAKES CALLS TO
0011 C                  SUBROUTINES TO CREATE THE INPUT FORM. THE DATA IS
0012 C                  ENTERED AND THEN WRITTEN TO A MONTHLY FILE. THE
0013 C                  PROGRAM WILL CREATE A NEW MONTHLY FILE WHEN NECESSARY.
0014 C
0015 C      DATE: 1/27/86
0016 C
0017 C *****
0018
0019
0020      INCLUDE 'ERDS$INCLUDE:FORMFLAGS.INC'
0058      DIMENSION MONTH_NAME(12)
0059
0060      Byte      Form_Interface_Record( 48)
0061      Integer*4  Iflags( 12)      ! Form flags
0062 C
0063 C      Type      Name      Field Prompt String
0064 C
0065      Integer*4  I_VAR001      ! Date of Forecast (000101 - 990101)/ Default = 999999
0066      Integer*4  I_VAR002      ! Grave Yard Forecaster Identification (1 - 9)/ Default = 0
0067      Integer*4  I_VAR003      ! Day Forecaster Identification (1 -9)/ Default = 0
0068      Integer*4  I_VAR004      ! Wind Spd Hour 1 (0 - 115)/ Default = 999
0069      Integer*4  I_VAR005      ! Wind Spd Hour 2 (0 - 115)/ Default = 999
0070      Integer*4  I_VAR006      ! Wind Spd Hour 3 (0 - 115)/ Default = 999
0071      Integer*4  I_VAR007      ! Wind Spd Hour 4 (0 - 115)/ Default = 999
0072      Integer*4  I_VAR008      ! Wind Spd Hour 5 (0 - 115)/ Default = 999
0073      Integer*4  I_VAR009      ! Wind Spd Hour 6 (0 - 115)/ Default = 999
0074      Integer*4  I_VAR010      ! Wind Spd Hour 7 (0 - 115)/ Default = 999
0075      Integer*4  I_VAR011      ! Wind Spd Hour 8 (0 - 115)/ Default = 999
0076      Integer*4  I_VAR012      ! Wind Spd Hour 9 (0 - 115)/ Default = 999
0077
0078      INTEGER*4  YEAR, MONTH, DAY, DAYS_IN_MONTH(12),
0079                  DAYS_IN_MONTH_LY(12)
0080
0081      CHARACTER*1 SELECT
0082      CHARACTER*3 MONTH_NAME
0083
0084      Equivalence (Form_Interface_Record, I_VAR001)
0085
0086      Common /Form_Record/ I_VAR001, I_VAR002, I_VAR003, I_VAR004,
0087                      I_VAR005, I_VAR006, I_VAR007, I_VAR008,

```

```

0088      I_VAR009, I_VAR010, I_VAR011, I_VAR012
0089
0090      DATA DAYS_IN_MONTH/31, 28, 31, 30, 31, 30, 31, 31, 30, 31,
0091      30, 31/
0092
0093      DATA DAYS_IN_MONTH_LY/31, 29, 31, 30, 31, 30, 31, 31, 30,
0094      31, 30, 31/
0095
0096      DATA MONTH_NAME/'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN',
0097      'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC'/
0098
0099
0100      MONTH_FLAG = 0
0101
0102      OPEN(UNIT=10, FILE='[FCST.PREVIOUS_DATE]TNKFCST_DATE.DAT',
0103      STATUS='OLD', IOSTAT=IER)
0104      IF(IER.NE. 0) WRITE(*,10) IER
0105 10      FORMAT(/5X,'Error Opening [FCST.PREVIOUS_DATE]TNKFCST_DATE',
0106      'DAT Code - ',I4)
0107
0108
0109      C..... READ PREVIOUS DATE AND HOUR FROM SUMMARY DATE FILE
0110      C..... AND OPEN MONTHLY FILE
0111
0112
0113      READ(10,'(3I2)') YEAR, MONTH, DAY
0114
0115      OPEN(UNIT=15, FILE='[FCST.DAT]TNKFCST'//MONTH_NAME(MONTH)//
0116      '.DAT', STATUS='OLD', ACCESS='APPEND',
0117      CARRIAGECONTROL='LIST', IOSTAT=IER)
0118      IF(IER.NE. 0) WRITE(*,80) IER, MONTH_NAME(MONTH)
0119 80      FORMAT(/5X,'Error Opening [FCST.DAT]TNKFCST',A3,
0120      'DAT Code - ', I4)
0121
0122
0123      C..... INCREMENT DATE AND OPEN NEW MONTHLY FILE IF NEEDED
0124
0125
0126      IF(DAY.EQ.31 .AND. MONTH.EQ.12) THEN      ! check for end of year
0127
0128          YEAR = YEAR + 1
0129          MONTH = 1
0130          DAY = 1
0131          MONTH_FLAG = 1
0132          OPEN(UNIT=11, FILE='[FCST.DAT]TNKFCST'//MONTH_NAME(MONTH)//
0133          '.DAT', STATUS='NEW', CARRIAGECONTROL='LIST',
0134          IOSTAT=IER)
0135          IF(IER.NE. 0) WRITE(*,20) IER, MONTH_NAME(MONTH)
0136 20      FORMAT(/5X,'Error Opening [FCST.DAT]TNKFCST',A3,
0137          'DAT Code - ', I4)
0138
0139      ELSE
0140
0141          IF(MOD(YEAR,4) .EQ. 0) THEN      ! check for leap year

```

```

0142
0143         IF(DAY.EQ.DAYS_IN_MONTH_LY(MONTH)) THEN
0144
0145             MONTH = MONTH + 1
0146             DAY = 1
0147             MONTH_FLAG = 1
0148             OPEN(UNIT=11, FILE='[FCST.DAT]TNKFCST'//
0149                 MONTH_NAME(MONTH)//'.DAT', STATUS='NEW',
0150                 CARRIAGECONTROL='LIST', IOSTAT=IER)
0151             IF(IER.NE.0) WRITE(*,20) IER, MONTH_NAME(MONTH)
0152
0153         ELSE
0154
0155             DAY = DAY + 1
0156
0157         ENDIF
0158
0159     ELSE
0160
0161         IF(DAY.EQ.DAYS_IN_MONTH(MONTH)) THEN
0162
0163             MONTH = MONTH + 1
0164             DAY = 1
0165             MONTH_FLAG = 1
0166             OPEN(UNIT=11, FILE='[FCST.DAT]TNKFCST'//
0167                 MONTH_NAME(MONTH)//'.DAT', STATUS='NEW',
0168                 CARRIAGECONTROL='LIST', IOSTAT=IER)
0169             IF(IER.NE.0) WRITE(*,20) IER, MONTH_NAME(MONTH)
0170
0171         ELSE
0172
0173             DAY = DAY + 1
0174
0175         ENDIF
0176
0177     ENDIF
0178
0179 ENDIF
0180
0181
0182 C..... WRITE NEW DATE TO PREVIOUS DATE FILE
0183
0184
0185     BACKSPACE (UNIT=10,IOSTAT=IER)
0186     IF(IER.NE.0) WRITE(*,30) IER
0187 30   FORMAT(/5X,' Error During Backspace Code - ',I4/
0188         5X,' Check [FCST.PREVIOUS_DATE]TNKFCST_DATE.DAT File')
0189     WRITE(10,'(3I2)') YEAR, MONTH, DAY
0190
0191
0192 C..... FIX DATE TO THE VARIABLES I_VAR001
0193
0194
0195     I_VAR001 = (YEAR+10000) + (MONTH+100) + DAY

```

```

0196
0197
0198
0199 C..... PREVENT USER FROM ENTERING DATE AND FORCE CURSOR TO START
0200 C..... AT FIRST VARIABLE
0201
0202
0203         IFLAGS(1) = FM$ _NO_WRITE
0204         IFLAGS(2) = FM$ _FORCE_START
0205
0206
0207 C..... MAKE CALLS TO SUBROUTINES TO CREATE INPUT FORM AND ENTER DATA
0208
0209
0210 100  CALL SUBPROCESS_OPEN_IO
0211      CALL SELECT_ANSI
0212      Call FM_Form_Open (12, 'HMSFORMS.FRM', IError)
0213      Call FM_Form_Input ('TNKFCST', Form_Interface_Record,
0214                          Iflags, Ierror, User_Check, *990)
0215      Call FM_Form_Close (IError)
0216      CALL SUBPROCESS_CLOSE_IO
0217
0218
0219 C..... WRITE DATA TO SCREEN AS WILL BE WRITTEN TO FILE AND
0220 C..... PROMPT FOR CORRECTION
0221
0222
0223      TYPE *, ' '
0224      TYPE *, ' '
0225      TYPE *, ' '
0226      TYPE *, ' The Following Data Record Will be Appended to ',
0227      'Monthly Tank Farm Forecast File : '
0228
0229      WRITE(*,40)
0230 40  FORMAT(/,1X,'123456789 123456789 123456789 123456789 ',
0231      '123456789 123456789 123456789')
0232      WRITE(*,50) I_VAR001, I_VAR002, I_VAR003, I_VAR004,
0233      I_VAR005, I_VAR006, I_VAR007, I_VAR008,
0234      I_VAR009, I_VAR010, I_VAR011, I_VAR012
0235
0236 50  FORMAT(/,1X,I0,2I1,21X,9I3)
0237
0238      SELECT = 'N'
0239
0240      TYPE *, ' Do You Need To Make Any Corrections ?'
0241      TYPE *, ' Enter N or Y'
0242      READ(*,'(A)') SELECT
0243
0244      IF(SELECT.EQ.'Y' .OR. SELECT.EQ.'y') GOTO 100
0245
0246
0247 C..... WRITE DATA TO MONTHLY FILE
0248
0249

```

```

0250      IF(MONTH_FLAG .EQ. 1) THEN
0251          LUN = 11
0252      ELSE
0253          LUN = 15
0254      ENDIF
0255
0256      WRITE(LUN, 00) I_VAR001, I_VAR002, I_VAR003, I_VAR004,
0257          I_VAR005, I_VAR006, I_VAR007, I_VAR008,
0258          I_VAR009, I_VAR010, I_VAR011, I_VAR012
0259      00  FORMAT(I0,2I1,2I1,9I3)
0260
0261
0262      TYPE *, ' '
0263      TYPE *, ' '
0264      TYPE *, ' Data Appended to: [FCST.DAT]TNKFCST'
0265          '//MONTH_NAME(MONTH)//'.DAT File.'
0266      TYPE *, ' '
0267      TYPE *, ' '
0268
0269      GO TO 200
0270
0271      990  TYPE *, ' '
0272          TYPE *, '      Error During Form Input - Call Gregg A.'
0273          TYPE *, '      Home 948-5482   Work 376-0136'
0274      200  Call Exit
0275      END

```

PROGRAM SECTIONS

Name Bytes Attributes

0 \$CODE	1771	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	730	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	460	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 FORM_RECORD	46	PIC OVR REL GBL SHR NOEXE RD WRT LONG

Total Space Allocated 3025

ENTRY POINTS

Address Type Name References

0-00000000 TANK\$MAIN

VARIABLES

Address Type Name Attributes References

2-000000C0	I+4	DAY	70	113=	126	130=	143
140=							
155(2)=	161	164=	173(2)=	189	195		

2-000000C8	I*4	IER			102=A	104(2)	115=A	118(2)	132=A
135(2)									
		148=A	151(2)	166=A	169(2)	185=	186(2)		
2-000000D0	I*4	IERROR				213A			
2-000000CC	I*4	IGERROR				212A	215A		
3-00000000	I*4	I_VAR001			COMM EQUIV	85	84	86	195= 232
258									
3-00000004	I*4	I_VAR002			COMM EQUIV	86	86	232	258
3-00000008	I*4	I_VAR003			COMM EQUIV	87	86	232	256
3-0000000C	I*4	I_VAR004			COMM EQUIV	88	86	232	256
3-00000010	I*4	I_VAR005			COMM EQUIV	89	86	232	256
3-00000014	I*4	I_VAR006			COMM EQUIV	70	86	232	256
3-00000018	I*4	I_VAR007			COMM EQUIV	71	86	232	256
3-0000001C	I*4	I_VAR008			COMM EQUIV	72	86	232	256
3-00000020	I*4	I_VAR009			COMM EQUIV	73	86	232	256
3-00000024	I*4	I_VAR010			COMM EQUIV	74	86	232	256
3-00000028	I*4	I_VAR011			COMM EQUIV	75	86	232	256
3-0000002C	I*4	I_VAR012			COMM EQUIV	76	86	232	256
**	I*4	LUN				251=	253=	258	
2-000000BC	I*4	MONTH				78	113=	115	118 126
129=									
		132	135	143	145(2)=	148	151		
		161	163(2)=	166	169	189	195		
264									
2-000000C4	I*4	MONTH_FLAG				100=	131=	147=	165= 258
2-000000B4	CHAR	SELECT				81	238=	242=	244(2)
2-000000D4	R*4	USER_CHECK				213A			
2-000000B8	I*4	YEAR				78	113=	128(2)=	141 189
195									

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References
2-00000030	I*4	DAYS_IN_MONTH		48	(12)	78 90D
181						
2-00000060	I*4	DAYS_IN_MONTH_LY		48	(12)	78 93D
143						
3-00000000	L*1	FORM_INTERFACE_RECORD	COMM EQUIV	48	(48)	60 84
213A						
2-00000000	I*4	IFLAGS		48	(12)	81 203=
204=						
213A						
2-00000090	CHAR	MONTH_NAME		36	(12)	58 82
96D						
115	118	132	135	148		
151	166	189	264			

PARAMETER CONSTANTS

Type	Name	References			
I*4	FM\$ _FIELD_BOLD	44	48		
I*4	FM\$ _FIELD_UNDERLINE	44	48		
I*4	FM\$ _FORCE_START	44	48	204	
I*4	FM\$ _NO_ECHO	44	48		
I*4	FM\$ _NO_SEE	44	48		
I*4	FM\$ _NO_WRITE	44	48	203	
I*4	FM\$ _SET_ERROR	44	48		
I*4	FM\$ _SET_EXCEPTION	44	48		
I*4	FM\$ _USER_CHECK	44	48		

LABELS

Address	Label	References			
1-00000121	10'	104	105		
1-000001A1	20'	135	136	151	169
1-000001D8	30'	188	187		
1-00000235	40'	229	230		
1-00000283	50'	232	236		
1-00000295	80'	258	259		
1-0000018A	80'	118	119		
0-0000038C	100	210	244		
0-000006E4	200	269	274		
0-00000890	990	213A	271		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References				
	FM_FORM_CLOSE	215				
	FM_FORM_INPUT	213				
	FM_FORM_OPEN	212				
	FOR\$EXIT	274				
	FOR\$OPEN	102	115	132	148	166
	SELECT_ANSI	211				
	SUBPROCESS_CLOSE_IO	210				
	SUBPROCESS_OPEN_IO	210				

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 TANK

/CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)

/DEBUG=(NOSYMBOLS,TRACEBACK)

/STANDARD=(NOSYNTAX,NOSOURCE_FORM)

/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)

/WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)

/CONTINUATIONS=10 /CROSS_REFERENCE /NOD_LINES /NOEXTEND_SOURCE /F77

/NOG_FLOATING /I4 /NOMACHINE_CODE /OPTIMIZE

COMPILATION STATISTICS

Run Time:	7.99 seconds
Elapsed Time:	10.94 seconds
Page Faults:	408
Dynamic Memory:	432 pages

DISTRIBUTION

No. of
Copies

OFFSITE

2 DOE Technical Information Center

ONSITE

DOE Richland Operations Office

J. J. Sutey

28 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)