QC
874.3
.U61
no.12
c.2

NOAA, NATIONAL WEATHER SERVICE, CENTRAL REGION

COMPUTER PROGRAMS AND PROBLEMS

NWS CRCP - NO. 12

PROGRAM ERROR

JAN [illegible] 1984

N.O.A.A.
U. S. Dept. of Commerce

Thomas F. Schwein
National Weather Service
Central Region Headquarters
Kansas City, MO

December 1983

U.S. DEPARTMENT OF COMMERCE / National Oceanic and Atmospheric Administration / National Weather Service

Cover 1--Tech Report--PMS 187 Red
White Vellum, Sub 100--S/S

1 CLNO H QC874.3.U61 4 RMKS Add to OCLC #8506147

NOAA, National Weather Service
Central Region Computer Programs and Problems Series

NWS CRCP No. 1	A Computer Program for Determining and Displaying the Geostrophic Winds on Lakes Superior and Michigan. R. Somrek. June 1980 (PB80 225675)
NWS CRCP No. 2	Program ALEMBIC. W. Sunkel. March 1982 (PB83 114488)
NWS CRCP No. 3	A Subroutine to Find and Extract Data from an AFOS Database Product. R. Somrek. May 1982 (PB83 118828)
NWS CRCP No. 4	Programs SAVALL and STORALL. W. Sunkel. July 1982. (PB83 118448)
NWS CRCP No. 5	Program ANALYZ. T. Schwein. Sept. 1982
NWS CRCP No. 6	A Regional Weather Depiction Plot. W. Sunkel. October 1982. (PB83 194415)
NWS CRCP No. 7	The Topeka Library (TOP.LB). W. Sunkel. March 1983
NWS CRCP No. 8	A Multipurpose Weather Roundup Program. Warren E. Sunkel May 1983. (PB83 222612)
NWS CRCP No. 9	Assembly Language Graphics Library (EGR1.LB) with Fortran Interfacing. Thomas J. Egger July 1983 (PB83 239624)
NWS CRCP No. 10	A Thermodynamics Library for AFOS Programmers (THERMO.LB) Warren E. Sunkel, November 1983
NWS CRCP No. 11	Meteorological Applications Library (CRH.LB) Thomas F. Schwein November 1983

I/1125

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE

QC
874.3
U61
no. 12
c. 2

NOAA, NATIONAL WEATHER SERVICE, CENTRAL REGION
Computer Programs and Problems

NWS CRCP - No. 12

PROGRAM ERROR

Thomas F. Schwein
National Weather Service
Central Region Headquarters
Kansas City, MO

LIBRARY
FEB 6 1984
N.O.A.A.
U. S. Dept. of Commerce

December 1983

UNITED STATES DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary /
National Oceanic and Atmospheric Administration
John V. Byrne, Administrator /
National Weather Service
Richard E. Hallgren,
Assistant Administrator

PROGRAM ERROR

Thomas F. Schwein
National Weather Service Central Region Headquarters
Kansas City, MO

I. GENERAL INFORMATION

ERROR is an AFOS application program that maintains an automated log of AFOS hangs, along with the time down, system configuration and an indication of what might have caused the failure and how busy the system was.

ERROR runs on the Eclipse S/230 minicomputer under RDOS. It is written in Data General Fortran 4 and requires less than 20 seconds to execute.
Error uses RDOS generated system information and therefore is independent of the various AFOS loads. In short, ERROR can be used with WSFO loads, AWSO loads, WSO loads and ARFC loads. In addition, ERROR does not have to be updated when a new AFOS software version is installed.

II. APPLICATION PROGRAM DESCRIPTION

A. SUPPORTING FILES-

ERROR.SV- Executable Save File
ERROR.DT- Text File Containing Task ID Descriptions
ERROR.LS- Listing of Site Hangs/Times/Causes created by ERROR.SV
MODE.FL- Contains AFOS operational configuration information
FBREAK.SV (BREAK.SV)- The RDOS break file.

ERROR.SV is created by the using the following load line:

```
RLDR ERROR ETIT BNAS TIMEDOWN GDAY CTOP BG UTIL FORT AFOSE1.LB
```

B. PROGRAM STRUCTURE AND LOGIC

Whenever AFOS experiences a hang, a break file is usually created. This break file can be analyzed for information on which tasks were active at the time of the failure, and the date/time information stored in SYS.DR can be used to obtain the time of the failure, as well as the length of time required to restore the system.

ERROR.SV is run everytime that AFOS is started, so that it can analyze the break file, if it exists, and record the information obtained from it in a file called ERROR.LS. It is run by including it as the first command in the AFOS start macros, so that it runs automatically when AFOS is restarted.

ERROR.SV uses several switches to denote what mode AFOS is in when bringing it up. This mode information is stored in an RDOS file called MODE.FL for use the next time that the ERROR program is run. When ERROR logs the operational AFOS mode in ERROR.LS, it always uses the information contained in the MODE.FL created during the previous restart. Permissable Global switches are:

F- Full dual computer AFOS at WSFO's, normal AFOS at WSO's
D- Degraded mode at WSFO's (not used at WSO's)
M- When used with either of the above switches, denotes system was brought up in Modify.

If the ERROR program is run without any switches, no logging of the information obtained takes place. This allows an operator to manually run the ERROR program to obtain the Task ID information before actually bringing AFOS up, and then still get the correct down-time recorded in ERROR.LS when the system is actually brought up.

ERROR always checks location 0 of the break file for a 3515 octal. This value denotes a break file that has already been read, and thus, was not created during the latest AFOS hang. Whenever the ERROR program is finished, it writes 3515 octal into location 0 to indicate that the break file has been used.

There are several possible situations where ERROR may read a break file that was previously read (3515 octal is found in location 0). This will occur whenever AFOS is terminated with a Control F at the Dasher, since Control F does not create a break file. Operators should always use Control K at an ADM to shut down AFOS in order to get break files.
An old break file will also result whenever AFOS goes down with an RDOS crash. Old break files will also be read whenever AFOS hangs and cannot be brought down to FG TERM. In those cases where an old break file is read in, ERROR logs an unknown (UNK) in the ERROR.LS file, along with the system configuration and date/time of system restoration.

Since ERROR is very dependent on a new break file being created each time AFOS shuts down, operators should be trained to shut down AFOS using Control K at an ADM, or using the new AFOS feature that allows an operator to input 377 octal in the switches and hit STOP START in order to recover from a hang in USER MODE. Both of these methods create break files.

III PROGRAM IMPLEMENTATION

A. Put ERROR.SV and ERROR.DT on disk and install appropriate links. In order to implement automated logging of AFOS failures, it is necessary to modify the AFOS start macros so that the ERROR program is executed before AFOS is started. Regional authorization to modify the AFOS start macros is required before making this change.

B. INTERPRETATION OF OUTPUT

Six different pieces of information are recorded in ERROR.LS every time AFOS is restarted. They are described in detail below.

DATE: This is the date when the AFOS system was restarted

TIME: This is the time that the AFOS system was restarted

TDWN: Time down-This is the number of minutes between the time that AFOS went down and when it was restarted. If stars appear in this column, this indicates that more than 999 minutes elapsed before the system was restarted. If a negative number appears, this indicates that at one point, the system clock was set incorrectly. If UNK appears, this indicates that no break file was created, and as a result, the time down could not be calculated.

MODE: This is the system configuration at the time of the AFOS failure.
F indicates normal operation, while D indicates Degraded Mode mode at a WSFO. An M after either F or D indicates that the system was brought up in modify before the last failure.

TSK#: task Number-This is the Task ID number that last issued a system call. Frequently, this Task ID will indicate what the AFOS system was doing when the AFOS programs shut down. Sometimes it can even be used to diagnose a hardware or software problem. The interpretation of the Task ID information is highly subjective, and an understanding of how the RDOS task scheduler works and what each AFOS task does is helpful. A summary of what has been learned so far regarding the meaning of task numbers is found as Appendix 1 of this report.

When interpreting the Task ID's, remember that seeing a certain Task ID once or twice does not necessarily indicate any software or hardware trouble. A problem is more likely when a given Task ID is seen frequently or occurs according to a set pattern.

#ACT: This is the number of ready tasks that were in the active chain of the task scheduler. This number gives a measurement of how busy the AFOS system was when it failed. You will need to collect data for several weeks so you can baseline your system, to see what is normal for your configuration. After that, you can judge how much busier than normal your AFOS system was when the failure occurred.

ERROR also prints a message on the dasher whenever it is run, giving a short analysis of what it found in the break file. It retrieves a plain-language description of the Task ID from ERROR.DT and prints it on the Dasher. If ERROR.DT is not found, ERROR prints only the Task ID number. If the break file has previously been read, the message FBREAK NOT UPDATED is printed on the dasher, indicating an old break file. If no break file is found, the message FBREAK NOT FOUND is printed out and no logging in ERROR.LS takes place.

APPENDIX 1

task #	Indication
6,7,11,12,13,14	Problem occurred in the Asynchronous comms area. Problem may have been just an overload of the asynchronous comms software, or it may have been a bad block, bad product, or faulty piece of hardware. If you continue to get these Task ID's, you probably have a problem in the asynchronous area.
10	This is the starting task for message composition-KBRD1. If you get this Task ID, it points to message composition.
15	Under normal circumstances, this is the Task ID you'll get with a Control K at an ADM. This Task ID can also result from a FICR hang, but this is rather rare.
16	This is the FICR overlay manager task. If you get this Task ID, it indicates that AFOS failed during a FICR command (Product call-up, PIL/CIL retrieval, ect)
17	Legend Indicator task-this is the task that keeps the Date/Time Current on the ADM's, as well as placing DATA DISPLAY or MESSAGE COMP on the legend line.
21	This is a Data Display/Message Comp task.
22	Procedure executive-this task manages the overlays which run AFOS procedures.
24	This is another FICR task that reads commands from the ADM keyboard and

	sends them to other tasks for processing.
25	Dual Processor Watchdog-This task checks periodically to make sure that the other processor (WSFO dual system) is still communicating over the MCA. If you get this Task ID on the DPCM, it indicates that the DCM computer shut down the DPCM, ie, the problem occurred on the DCM. If you run ERROR on the DCM side, you will get this Task ID during a normal shutdown of AFOS.
26	This task manages the Alarm/Alert System and other external events notification.
37	Extended Share task-This is the second most common Task ID you will find in the ERROR.LS listing. This task manages shared buffers in AFOS, and will shut down the system if negative buffer use count or a resource lockout occurs. With the software bugs still in AFOS, this can be considered a "normal" Task ID in the ERROR.LS.
40	Time purge task
41,42	Version purge task
43	Data storage task.
44	Data purge block task
47	Autodialer task
50	MCA write Q'r.
51	Background ICE task. This task interfaces with the BGMON program in background to run application programs
75	MCA writer

76	MCA reader
1Ø3	Asynchronous time scheduler

There are other Task ID's in AFOS, but these are not likely to be seen in the ERROR.LS file. A complete listing of Task ID's is found in Appendix 2 of AFOS Handbook No. 5, Volume 2, Fault Recognition and Recovery.

A P P E N D I X 2

```
C         PROGRAM: ERROR.FR

      INTEGER JDATE(3),JTIME(3)
      INTEGER IDATE(3),IBUF(16),ITIME(3)
      INTEGER IDAT(7),ISW(2)
      INTEGER MESSAGE(24)
C GOING INTO WHAT MODE
C       D=DEGRADED MODE
C        F=FULL AFOS
C        FM=DUAL AFOS-BROUGHT UP IN MODIFY
C        DM=DEGRADED AFOS-BROUGHT UP IN MODIFY
C       U=UNKNOWN-NO LOGGING TAKES PLACE-ASSUME KEYBOARD COMMAND
      IVAL=Ø
      CALL GDAY(IDATE,ITIME);RETRIEVE SYSTEM INFO ON CREATION TIME FOR FBREAK.SV
      CALL FCOM(IC,IER)
      MODE=2HU
      CALL COMCM(IC,IDAT,N,ISW,IER)
      IF(ISWSET(ISW,"F")) MODE=2HF
      IF(ISWSET(ISW,"D")) MODE=2HD
      IF(ISWSET(ISW,"M").AND.ISWSET(ISW,"F")) MODE=2HFM
      IF(ISWSET(ISW,"M").AND.ISWSET(ISW,"D")) MODE=2HDM
      CALL COMCM(IC,IDAT,N,ISW,IER);CLOSE COMC.CM
C READ IN ERROR LISTING-IF IT DOESNT EXIST, CREATE IT
      CALL OPEN(3,"ERROR.LS",2,IER,37)
      READ(3,2Ø2,END=1Ø) NUM
 2Ø2  FORMAT(I5,3ØX)
      GO TO 11
 1Ø   NUM=Ø;CREATE A NEW ERROR.LS
      WRITE(3,2Ø1) NUM
      WRITE(3,2Ø3)
      WRITE(3,2Ø4)
 2Ø3  FORMAT(3X,"DATE",4X,"TIME  TDWN MODE",1X,"TSK#"," #ACT")
 11   CONTINUE
C OPEN AND READ IN FBREAK.SV-CHECK WORD Ø-HAS ERROR ALREADY LOOKED AT THIS
C BREAK FILE ?
 2Ø4  FORMAT(36X)
      CALL OPEN(2,"FBREAK.SV",2,IER,2)
      READ BINARY(2,END=1) IV
      IF(IV.NE.3515K) GO TO 4
C ALREADY SEEN THIS FBREAK-MAYBE AN RDOS CRASH-ANYWAY LOG AS AN UNKNOWN CAUSE
      TYPE "FBREAK.SV NOT UPDATED"
      CALL DATE(IDATE,IER);FBREAK NOT CURRENT-DONT USE SYSTEM DATE/TIME IN
      CALL TIME(ITIME,IER);                                SYS.DR
      IVAL=-99
C  READ IN OLD MODE FILE-UPDATE WITH NEW INFORMATION
 4    CONTINUE
      IF(MODE.EQ.2HU ) GO TO 6Ø
      CALL OPEN(4,"MODE.FL",Ø,IER,2)
      JMODE=2H
      READ BINARY(4,END=45) KMODE
      JMODE=KMODE
 45   REWIND 4
      WRITE BINARY(4) MODE
      CALL CLOSE(4,IER)
      MODE=JMODE
 6Ø   IF(IVAL.EQ.-99.AND.MODE.EQ.2HU ) CALL EXIT
      IF(IVAL.EQ.-99) GO TO 7;FBREAK NOT NEW-NO ADDITIONAL INFORMATION HERE
C READ IN TCB INFORMATION FROM FBREAK FILE
      ILOC=414K-16K
      CALL FSEEK(2,ILOC)
      READ BINARY(2,END=1) ILOC1
      GO TO 2
 1    TYPE "FILE DOES NOT EXIST-FBREAK.SV"
      CALL EXIT
  2   ILOC1=ILOC1-16K+12K
      CALL FSEEK(2,ILOC1)
      READ BINARY(2) IVAL
C HOW MANY TASKS ACTIVE?
      NTASKS=Ø
      ILOC=415K-16K
      CALL FSEEK(2,ILOC)
      READ BINARY(2) ILOC
      ILOC=ILOC-16K
 36   CALL FSEEK(2,ILOC)
      DO 35 J=1,16
 35   READ BINARY(2) IBUF(J)
      ISTAT=ISHFT(IAND(IBUF(6),1774ØØK),-8)
      IF(ISTAT.LT.7) NTASKS=NTASKS+1
      ILOC=IBUF(8)-16K
      IF(IBUF(8).GE.Ø) GO TO 36
C REWIND FBREAK.SV AND WRITE 3515K INTO WORD Ø-THIS SIGNIFIES THAT
C THIS FBREAK HAS ALREADY BEEN READ FROM
      IF(MODE.EQ.2HU ) GO TO 61
      REWIND 2
      IV=3515K
      WRITE BINARY(2) IV
 61   CALL CLOSE(2,IER)
C INTERPRATE ERROR CODE-PRINT ON DASHER
      IVAL=IVAL.AND.377K
      IF(IVAL.LT.51K) IREC=IVAL
      IF(IVAL.GT.51K.AND.IVAL.LT.7ØK) WRITE(1Ø,1ØØ) IVAL
      IF(IVAL.GT.51K.AND.IVAL.LT.7ØK) CALL EXIT
      IF(IVAL.GE.7ØK.AND.IVAL.LT.1ØØK) IREC=IVAL-15
      IF(IVAL.EQ.1ØØK) WRITE(1Ø,1ØØ) IVAL
      IF(IVAL.EQ.1ØØK) CALL EXIT
      IF(IVAL.GT.1ØØK.AND.IVAL.LT.1Ø6K) IREC=IVAL-16
      IF(IVAL.GE.1Ø6K) WRITE(1Ø,1ØØ) IVAL
      IF(IVAL.GE.106K) CALL EXIT
 100  FORMAT(1X,"NON-DEFINED TASK CODE: ",OI4)
      CALL OPEN(2,"ERROR.DT",2,IER,48)
      CALL FSEEK(2,IREC)
      READ BINARY(2,END=3) MESSAGE
      CALL CLOSE(2,IER)
      WRITE(10,101) IVAL,MESSAGE
 101  FORMAT(1X,"AFOS WAS TERMINATED IN TASK ID # ",OI3,3X,24(A2))
      IF(MODE.NE.2HU ) GO TO 7
      CALL CLOSE(3,IER)
      CALL EXIT
C LOG THE INFORMATION IN ERROR.LS
 7    CALL DATE(JDATE,IER)
      IF(JDATE(3).GT.100) JDATE(3)=JDATE(3)-1900
      CALL TIME(JTIME,IER)
      CALL TIMEDOWN(IDATE,ITIME,JDATE,JTIME,NMIN)
      DO 46 K=1,3
      CALL BNAS(IDATE(K))
 46   CALL BNAS(ITIME(K))
      NUM=NUM+1
      KREC=NUM+2
      CALL FSEEK(3,KREC)
      IF(IVAL.NE.-99) WRITE(3,200) IDATE,ITIME(1),ITIME(2),NMIN,MODE,IVAL,NTASKS
 200  FORMAT(1X,A2,"/",A2,"/",A2,2X,A2,":",A2,2X,I4,2X,A2,2X,OI3,2X,I3)
      IF(IVAL.EQ.-99) WRITE(3,210) IDATE,ITIME(1),ITIME(2),MODE
 210  FORMAT(1X,A2,"/",A2,"/",A2,2X,A2,":",A2,2X," UNK",2X,A2,2X,"UNK",5X)
      REWIND 3
      WRITE(3,201) NUM
 201  FORMAT(1X,I5,30X)
      CALL CLOSE(3,IER)
      CALL EXIT
 3    WRITE(10,102) IVAL
 102  FORMAT(1X,"ERROR.DT NOT FOUND-TASK CODE WAS: ",OI4)
      GO TO 7;WRITE ERROR CODE TO ERROR.LS
      CALL EXIT
      END
```

```
C         PROGRAM: TIMEDOWN.FR
C
      SUBROUTINE TIMEDOWN(IDATE,ITIME,JDATE,JTIME,NMIN)
      INTEGER IDATE(3),ITIME(3),JDATE(3),JTIME(3)
      REAL JTIME1,ITIME1
      COMMON/AB/IMO(12)
      DATA IMO/31,28,31,3Ø,31,3Ø,31,31,3Ø,31,3Ø,31/
C CALCULATE ELAPSED TIME FROM IDATE,ITIME TO JDATE,JTIME
      IF(MOD(JDATE(3),4).EQ.Ø) IMO(2)=29
C CALCULATE JULIAN MINUTES
      I1=IDATE(1);STARTING MONTH
      IDAY=-IMO(I1)
      DO 1 I=1,I1
 1    IDAY=IDAY+IMO(I)
      IDAY=IDAY+IDATE(2)-1;JULIAN DATE-1
      ITIME1=IDAY*24*6Ø.;JULIAN MINUTES
      ITIME1=ITIME1+6Ø*(ITIME(1))+ITIME(2)
      I1=JDATE(1);ENDING MONTH
      JDAY=-IMO(I1)
      DO 2 I=1,I1
  2   JDAY=JDAY+IMO(I)
      JDAY=JDAY+JDATE(2)-1
      JTIME1=JDAY*24*6Ø.;JULIAN MINUTES TO PREVIOUS DAY
      JTIME1=JTIME1+6Ø*JTIME(1)+JTIME(2)
      IF(IDATE(3).EQ.JDATE(3)) GO TO 3
C DIFFERENT YEARS
      JTIME1=JTIME1+5256ØØ.Ø
      IF(MOD(IDATE(3),4).EQ.Ø) JTIME1=JTIME1+144Ø
 3    NMIN=JTIME1-ITIME1+Ø.5
      RETURN
      END
```

```
C         PROGRAM: BNAS.FR
C
      SUBROUTINE BNAS(IAR)
      DIMENSION IAR(3)
      DO 1 I=1,3
      IDUM=IAR(I)
      I1=IDUM/1Ø
      I2=IDUM-1Ø*I1+48
      I1=I1+48
 2    IAR(I)=ISHFT(I1,8)+I2
      RETURN
      END
```

```
C          PROGRAM: GDAY.FR
C
      SUBROUTINE GDAY(IDATE,ITIME)
      INTEGER IDATE(3),ITIME(3),IAR(25)
      CALL STAT("FBREAK.SV",IAR,IER)
      IF(IER.EQ.1) GO TO 1
      CALL DATE(IDATE,IER)
      CALL TIME(ITIME,IER)
      RETURN
      ITIME(1)=ISHFT(IAND(IAR(14),177400K),-8)
      ITIME(2)=IAND(IAR(14),377K)
      RETURN
      END
```

```
C         PROGRAM: ETIT.SR
C

        .TITL ETIT
        .REV 1,Ø5.
        .END
```

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

The National Oceanic and Atmospheric Administration was established as part of the Department of Commerce on October 3, 1970. The mission responsibilities of NOAA are to assess the socioeconomic impact of natural and technological changes in the environment and to monitor and predict the state of the solid Earth, the oceans and their living resources, the atmosphere, and the space environment of the Earth.

The major components of NOAA regularly produce various types of scientific and technical information in the following kinds of publications:

PROFESSIONAL PAPERS — Important definitive research results, major techniques, and special investigations.

CONTRACT AND GRANT REPORTS — Reports prepared by contractors or grantees under NOAA sponsorship.

ATLAS — Presentation of analyzed data generally in the form of maps showing distribution of rainfall, chemical and physical conditions of oceans and atmosphere, distribution of fishes and marine mammals, ionospheric conditions, etc.

TECHNICAL SERVICE PUBLICATIONS — Reports containing data, observations, instructions, etc. A partial listing includes data serials; prediction and outlook periodicals; technical manuals, training papers, planning reports, and information serials; and miscellaneous technical publications.

TECHNICAL REPORTS — Journal quality with extensive details, mathematical developments, or data listings.

TECHNICAL MEMORANDUMS — Reports of preliminary, partial, or negative research or technology results, interim instructions, and the like.

NOAA CENTRAL LIBRARY
3 8398 1014 6922 3

Information on availability of NOAA publications can be obtained from:

ENVIRONMENTAL SCIENCE INFORMATION CENTER
ENVIRONMENTAL DATA AND INFORMATION SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

Rockville, MD 20852