NOAA Eastern Region Computer Programs
and Problems NWS ERCP - No. 26

SAOSUM: A SHORT SUMMARY OF OBSERVATIONS

Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, Ohio

Scientific Services Division
Eastern Region Headquarters
October 1984

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

NOAA TECHNICAL MEMORANDUM

National Weather Service, Eastern Region Computer Programs and Problems

The Eastern Region Computer Programs and Problems (ERCP) series is a subset of the Eastern Region Technical Memorandum series. It will serve as the vehicle for the transfer of information about fully documented AFOS application programs. The format ERCP - No. 1 will serve as the model for future issuances in this series.

1 An AFOS version of the Flash Flood Checklist. Cynthia M. Scott, March 1981. (PB81 211252).

2 An AFOS Applications Program to Compute Three-Hourly Stream Stages. Alan P. Blackburn, September 1981. (PB82 156886).

3 PUPPY (AFOS Hydrologic Data Reporting Program). Daniel P. Provost, December 1981. (PB82 199720).

4 Special Search Computer Program. Alan P. Blackburn, April 1982. (PB83 175455).

5 Conversion of ALEMBIC$ Workbins. Alan P. Blackburn, October 1982. (PB83 138313).

6 Real-Time Quality Control of SAOs. John A. Billet, January 1983. (PB83 166082).

7 Automated Hourly Weather Collective from HRR Data Input. Lawrence Cedrone, January 1983 (PB83 167122).

8 Decoders for FRH, FTJ and FD Products. Cynthia M. Scott, February 1983. (PB83 176057).

9 Stability Analysis Program. Hugh M. Stone, March 1983. (PB83 197947).

10 Help for AFOS Message Comp. Alan P. Blackburn, May 1983. (PB83 213561).

11 Stability and Other Parameters from the First Transmission RAOB Data. Charles D. Little, May 1983. (PB83 220475).

12 TERR, PERR, and BIGC: Three Programs to Compute Verification Statistics. Matthew R. Peroutka, August 1983. (PB84 127521).

13 Decoder for Manually Digitized Radar Observations. Matthew R. Peroutka, June 1983. (PB84 127539).

14 Slick and Quick Data Entry for AFOS Era Verification (AEV) Program. Alan P. Blackburn, December 1983. (PB84 138726).

15 MDR--Processing Manually Digitized Radar Observations. Matthew R. Peroutka, November 1983. (PB84 161462)

16 RANP: Stability Analysis Program. Hugh M. Stone, February 1984.(PB84 1614

17 ZONES. Gerald G. Rigdon, March 1984. (PB84 174325)

18 Automated Analysis of Upper Air Soundings to Specify Precipitation Type. Joseph R. Bocchieri and Gerald G. Rigdon, March 1984. (PB84 174333)

(Continued on Inside Rear Cover)

NOAA EASTERN REGION COMPUTER PROGRAMS AND PROBLEMS - No. 26

SAOSUM: A SHORT SUMMARY OF OBSERVATIONS

Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, Ohio

Scientific Services Division
Eastern Region Headquarters
October 1984

SAOSUM: A SHORT SUMMARY OF OBSERVATIONS

Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, Ohio

I. General Information

A. Summary

As more and more forecasters turn to AFOS to keep track of the hourly observations, aviation log sheets are disappearing. Although the AFOS system is suited to watching the current weather, there is no quick way to scan the past eight hours of data for a given geographic area. Console procedures and multiple product call-ups have helped to fill this need, but these techniques can be slow and cumbersome, especially when AFOS is under a heavy workload.

This paper describes a program which quickly produces a list of all the observations taken at a group of stations during the past eight hours. The program can help meteorologists prepare their forecasts and monitor the weather, but its main purpose is to brief a forecaster who has just started his or her shift.

B. Environment

The program is written in Data General's FORTRAN IV, and it runs in the background partition of the Eclipse minicomputer. The listing is routed directly to the Printer-Plotter Module (PPM) to save time.

II. Application

A. Complete Program Description

The subroutine SAOP checks the system clock, and generates the Julian time of the oldest product which will be printed. This value is returned in the real variable BTIM. SAOP then opens the data file SAOSUM.DA. The main program then attempts to open the file $LPT (the PPM). The subroutine SAOKY reads the key for the next observation from the file SAOSUM.DA. The routine SAOST drives the loop which calls up succeeding previous versions of the current key and prints them using the FTSPT subroutine.

The full text of each observation is printed, even if it requires more than one line on the printer. This provides the forecaster with a more complete record than most weather logs which are currently kept by hand. Special observations are printed as well as record observations. Since the full text of the observations is printed, remarks and additive data are available to the forecaster as well as the basic observational data.

B. Machine Requirements

The SAOSUM program requires about 2ØK of memory to run. The program file requires about twenty-five RDOS blocks of disk space, and the data file is usually one block in length. Each observation takes less than a second to print when the system is not under excessive load.

Figure 1: Sample SAOSUM.DA File

```
CLESAOTOL
CLESAOCLE
CLESAOYNG
CLESAODAY
CLESAOCVG
CLESAOZZV
CLESAOFDY
CLESAOMFD
CLESAOCAK
CLESAOCMH
CLESAOLUK
```

C. Database

The observations which are printed must be part of the AFOS database. Enough versions of each observation should be present to ensure at least eight hours of data in each product, otherwise infinite loops may occur due to a defect in the PRVF routine.

III. Procedures

A. Installation

The file SAOSUM.DA must be created with the M:F/ command or one of the text editors. Each line contains the nine-letter AFOS key name followed by a single carriage return. The program stops when it senses the end of the data file.

The SAOSUM.SV program can be loaded on DPØ or on DPØF with a link on DPØ. Figure 1 contains a sample SAOSUM.DA file.

Figure 2: Sample Output from SAOSUM.SV

```
 TOL SA 1552 250 -BKN 10 290/48/36/3105/037
 TOL SA 1452 250 -BKN 8 290/45/35/0000/037/  210 1001
 TOL SA 1352 120 SCT 250 -BKN 8 288/41/35/3405/036
 TOL SA 1250 100 SCT 250 -BKN 5H 285/38/33/0000/036
 TOL SA 1151 M150 BKN 7 280/37/33/3503/034/ 207 1070 45737
 TOL SA 1050 150 SCT 12 277/37/33/3504/033
 TOL SA 0951 150 SCT 10 276/40/34/3405/033/SCT V BKN
 TOL SA 0851 M150 BKN 10 269/40/35/3606/032/ 803 1070
-----------------------------------------------------------------------------
 CLE SA 1552 30 SCT 85 SCT 250 -OVC 15 280/53/38/0000/035
 CLE SA 1453 E85 BKN 15 277/51/37/1804/034/FEW CU/ 107 1170
 CLE SA 1351 E 85 OVC 10 277/49/39/1804/034/BINOVC
 CLE SA 1252 E85 OVC 10 275/46/40/1904/033/BINOVC
 CLE SA 1150 M85 OVC 20 270/45/40/2804/032/LB25E35/ 30300 107/ 41
 CLE SA 1050 M95 OVC 20 266/45/40/0000/031
 CLE SA 0950 M95 OVC 20 266/44/39/0000/031
 CLE SA 0850 M110 OVC 20 266/44/39/0000/031/ 603 107/
-----------------------------------------------------------------------------
 YNG SA 1550 E80 OVC 10 278/52/41/3304/033
 YNG SA 1450 E80 BKN 200 OVC 10 277/50/43/0000/033/ 110 1077
 YNG SA 1350 E80 OVC 10 279/48/43/0403/033
 YNG SA 1250 E80 OVC 10 270/45/44/0804/031
 YNG SA 1150 E80 OVC 7 268/45/43/0306/030/ 10301 107/ 20001 45344
 YNG SA 1050 E65 OVC 7 268/45/43/3604/030/RE30
 YNG SA 0950 E60 OVC 7RW- 268/45/43/0105/030/RB10
 YNG SA 0850 E80 OVC 7 264/45/42/0304/029/ 707 107/
-----------------------------------------------------------------------------
 DAY SA 1550 E70 OVC 8 280/51/39/0505/035
 DAY SA 1450 E70 OVC 8 278/49/39/0506/035/ 21200 107/
 DAY SA 1350 E65 OVC 10 273/48/40/0308/033/BINOVC
 DAY SA 1250 E50 OVC 8 270/46/40/0205/032/BINOVC RE10
 DAY SA 1150 E50 OVC 8R- 266/44/41/0206/031/ 10302 15// 44 20003
RADAT 95097
 DAY SA 1050 E50 OVC 8R- 266/45/41/3603/031
 DAY SA 0950 E50 OVC 8R- 261/45/40/0307/029
 DAY SA 0850 E50 OVC 12R- 262/45/40/3605/030/ 80201 15//
-----------------------------------------------------------------------------
 CVG SA 1550 M27 OVC 7 271/52/46/0105/033
 CVG RS 1450 M25 OVC 7 268/51/46/0105/032/ 312 15//
 CVG SA 1350 26 SCT M38 OVC 10 264/50/44/0703/031
 CVG SA 1250 28 SCT M40 OVC 10 256/49/43/0203/028/THIN SPOTS IOVC
 CVG SP 1218 M48 OVC 10 0105/028
 CVG SA 1152 M28 OVC 5H 255/49/44/0103/028/ 30701 15// 49 20001
 CVG SA 1052 M28 OVC 5H 252/49/44/0203/027/RE15
 CVG SA M27 OVC 4R-H 249/50/43/3605/026/RB38
 CVG SP 0935 M28 OVC 5H 3604/026
  .
  .
  .
  .
```

B. Initiation

To start the program, simply type RUN:SAOSUM at an ADM. The program will soon start printing observations at the Printer-Plotter Module.

C. Output

Figure 2 shows a sample of output from the SAOSUM program.

D. Cautions

The SAOSUM program takes its products from the AFOS database, and so its output is subject to all the vagaries which affect the AFOS system.

There must be at least eight hours' worth of observations for each site selected, otherwise infinite loops will occur.

IV. Program Listings

```
C       SAOSUM.SV                  OBSERVATION SUMMARY
        DIMENSION KYFT(5), KYSA(5)
        COMMON /FTS/ ISPCR(37)
        DATA ISPCR /"------------------------------------",
       +"------------------------------------",6415K/
C
C        THIS PROGRAM PRINTS A CHRONOLOGIC SUMMARY OF SAO PRODUCTS FOR THE
C        CLE NODE ON $LPT.  ALL PRODUCTS PRODUCED WITHIN SIX HOURS ARE
C        WRITTEN.
C
C        RLDR SAOSUM SAOST SAOP SAOKY FTSPT JDATE MAST <BG UTIL FORT>.LB
C
        KYFT(1) = 0
        CALL SAOP(BTIM,IRC)                  ; OPEN DATA FILE AND GET TIME.
        DO 100 I = 1, 5                      ; TRY TO OPEN PRINTER.
        CALL GCHN(IC,IER)
        CALL OPENE(IC,"$LPT",0,IER)
        IF (IER.EQ.1) GOTO 200
        CALL WAIT(5,2,IER)
   100  CONTINUE
        CALL FORKE("FTASUM","PPM TROUBLE",IER)    ; BOMB OUT!
        GOTO 900
   200  CALL SAOKY(IRC,KYSA,IER)
        IF (IER.NE.1) GOTO 900
        CALL SAOST(IC,KYSA,BTIM,IER)
        CALL WRL(IC,ISPCR,N,IER)
        GOTO 200
   900  CALL FORK("SAOSUM",IER)
        CALL KLOSE(IC,IER)
        CALL KLOSE(IRC,IER)
        CALL EXIT
        END
```

```
      SUBROUTINE SAOP(BTIM,IRC)
      DIMENSION IDT(3)
C
C      THIS SUBROUTINE COMPUTES THE TIME OF THE OLDEST VERSION WHICH WILL
C      BE PRINTED, AND OPENS THE DATA FILE.
C
      CALL DATE(IDT,IER)                        ; GET CURRENT DATE AND TIME.
      CALL FGTIM(IHR,I,J)
      JD = JDATE(IDT(1),IDT(2),IDT(3)) - 1    ; CONVERT TO JULIAN MINUTES.
      BTIM = JD*1440. + IHR*60. - 480.
      CALL GCHN(IRC,IER)                        ; OPEN DATA FILE.
      CALL OPENR(IRC,"SAOSUM.DA",0,IER)
      RETURN
      END
```

```
      SUBROUTINE SAOKY(IRC,KYSA,IRER)
      DIMENSION KYSA(5)
C
C      THIS SUBROUTINE READS THE SAO KEYS.  IRER RETURNS 1 FOR
C      SUCCESS.
C
      IRER = 1
      CALL RDL(IRC,KYSA,I,IER)
      IF (IER.EQ.1) GOTO 900
  800 IRER = -1
  900 RETURN
      END
```

```
      SUBROUTINE SAOST(IC,KYSA,BTIM,IRER)
      DIMENSION KYSA(5), KREC(20), IBF(128)
      PTIM(I,J) = ISHFT(I,-8)*16384. + IAND(I,377K)*128. + ISHFT(J,-8)
C
C      THIS SUBROUTINE PRINTS EACH VERSION OF KYSA UNTIL THE PREVIOUS VERSION
C      HAS A CREATION TIME BEFORE BTIM.
C
      IRER = 1
      CALL KSRCF(KYSA,KREC,IER)                        ; OPEN PRODUCT.
      IF (IER.NE.1) GOTO 800
  100 CALL RDBKF(0,IBF,IER)
      IF (IER.NE.1) GOTO 800
      IF (PTIM(IBF(9),IBF(10)).LT.BTIM) GOTO 900    ; VERSION TOO OLD.
      CALL FTSPT(IC,IBF,0,IER)
      IF (IER.NE.1) GOTO 800
      CALL PRVRF(IER)
      IF (IER.NE.1) GOTO 800
      GOTO 100
  800 IRER = -1
  900 RETURN
      END
```

```
      SUBROUTINE FTSPT(ICHN,IBF,ISP,IRER)
      DIMENSION IBF(128), IBFT(256), IBFU(1008), LINE(41)
C
C      THIS SUBROUTINE PRINTS UP TO FOUR BLOCKS OF ANY AFOS PRODUCT ON RDOS
C      CHANNEL ICHN.  AFOS BLOCK ZERO OF THE PRODUCT MUST RESIDE IN IBF.  IF
C      ISP IS 1, BLANK LINES ARE PRINTED BEFORE AND AFTER THE PRODUCT.  IRER
C      RETURNS 1 FOR SUCCESS.
C
      IRER  = 1
      DO 200 I = 1, 4                    ; LOAD THE BLOCKS INTO IBFU.
       CALL UNPACK(IBF,256,IBFT)
       LAST = I*252
       DO 100 J = 5, 256
  100  IBFU(LAST-256+J) = IBFT(J)
       CALL NXBKF(IBF,IER)
       IF (IER.NE.1) GOTO 300
  200 CONTINUE
  300 IF (ISP.EQ.1) CALL WRL(ICHN,"<40><15>",N,IER)  ; BLANK LINE.
      IF (ISP.EQ.1.AND.IER.NE.1) GOTO 800
      IB = MASK("<0><305><0><200>",2,IBFU,1,50); FIND START OF DATA.
      IF (IB.EQ.-1) GOTO 800
      IB = MASK(5015K,1,IBFU,IB,50)       ; FIND START OF DATA.
      IF (IB.EQ.-1) GOTO 800
      IB = MASK(20203K,1,IBFU,IB,100)
      IF (IB.EQ.-1) GOTO 800
      LAST = MASK(203K,1,IBFU,IB,LAST)    ; FIND END OF TEXT.
      IF (LAST.EQ.-1) GOTO 800
  400 IE = MASK(5015K,1,IBFU,IB,LAST)     ; FIND END OF LINE.
      IF (IE.EQ.-1) IE = LAST
      IBFU(IE) = 15K                      ; SINGLE CR ENDS OUTPUT LINE.
      N = IE - IB + 1                     ; LENGTH OF LINE.
      IF (N.GT.81) GOTO 800
      CALL PACK(IBFU(IB),N,LINE)          ; LOAD OUTPUT ARRAY.
      CALL WRS(ICHN,LINE,N,IER)           ; WRITE LINE.
      IF (IER.NE.1) GOTO 800
      IF (IE.EQ.LAST) GOTO 500            ; FIND START OF NEXT LINE.
      IB = MASK(20203K,1,IBFU,IE,LAST)
      IF (IB.EQ.-1) GOTO 800
      IF (IB.NE.LAST) GOTO 400
  500 IF (ISP.EQ.1) CALL WRL(ICHN,"<40><15>",N,IER)   ; BLANK LINE
      IF (IER.NE.1) GOTO 800
      GOTO 900                            ; END OF LOOP.
  800 IRER = -1                           ; END WITH ERROR.
  900 RETURN
      END
```

```
      FUNCTION MASK(MSK,LMSK,IUP,IBGN,ISTOP)
      DIMENSION MSK(1), IUP(1)
C
C      THIS FUNCTION SEARCHES IUP (AN UNPACKED ARRAY OF ASCII CHARACTERS) FROM
C      IBGN TO ISTOP FOR A STRING WHICH MATCHES THE MASK MSK.  MSK IS LMSK
C      WORDS IN LENGTH AND HAS THE FOLLOWING CHARACTERISTICS:
C
C           1.  IF THE LEFT BYTE OF A WORD IS 0, THE RIGHT BYTE NEEDS AN
C               EXACT MATCH.
C           2.  IF THE LEFT BYTE IS NON-ZERO, THEN A MATCH MUST BE GREATER
C               THAN OR EQUAL TO THE LEFT BYTE, AND LESS THEN OR EQUAL TO THE
C               RIGHT BYTE.  "09" OR "AZ", E. G.
C           3.  IF A WORD EQUALS -1, IT WILL MATCH ANY BYTE (A WILD CARD).
C
C      THE FUNCTION RETURNS THE LOCATION IN IUP OF THE BEGINNING OF THE
C      SUBSTRING.  MASK RETURNS -1 IF THE SEARCH FAILS.
C
      LIMIT = ISTOP - LMSK + 1
      DO 600 MASK = IBGN, LIMIT
       DO 500 I = 1, LMSK
        M = MSK(I)
        IF (M.EQ.-1) GOTO 500              ;WILD CARD.
        ML = ISHFT(M,-8)
        MR = IAND(M,377K)
        L = IUP(MASK+I-1)
        IF (ML.NE.0) GOTO 200              ;NEED AN EXACT MATCH.
         IF (MR.EQ.L) GOTO 500
         GOTO 600
  200   IF (L.LT.ML.OR.L.GT.MR) GOTO 600    ;RANGE FOR MATCH.
  500  CONTINUE
       GOTO 700
  600 CONTINUE
      MASK = -1                            ;UNSUCCESSFUL SEARCH.
  700 RETURN
      END
```

```
      INTEGER FUNCTION JDATE(MONTH,IDAY,IYEAR)
C
C      THIS FUNCTION WILL GENERATE A JULIAN DATE FOR THE MONTH, DAY, AND
C      YEAR INPUT.  IF THERE IS A PROBLEM WITH THE INPUT DATA, THE FUNC-
C      TION WILL RETURN A VALUE OF ZERO.
C
      COMMON /QJULQ/ MLIST(12)
      DATA MLIST/31,28,31,30,31,30,31,31,30,31,30,31/
      JDATE = 0                                  ;CHECK INPUT DATA.
      IF (MONTH.LT.1.OR.MONTH.GT.12) RETURN
      IF (IYEAR.LT.1) RETURN
      IF (IDAY.LT.1) RETURN
      MLIST(2) = 28
      I = IYEAR/4*4                              ;IS THIS A LEAP YEAR?
      IF (I.EQ.IYEAR) MLIST(2) = 29
      I = IYEAR/100*100
      IF (I.EQ.IYEAR) MLIST(2) = 28
      I = IYEAR/400*400
      IF (I.EQ.IYEAR) MLIST(2) = 29
      IF (IDAY.GT.MLIST(MONTH)) RETURN
      DO 100 I = 1,MONTH                         ;COMPUTE JULIAN DATE.
  100 JDATE = JDATE + MLIST(I)
      JDATE = JDATE - MLIST(MONTH) + IDAY
      RETURN
      END
```

Eastern Region Computer Programs and Problems (Continued)

19 Verification of Asynchronous Transmissions. Lawrence Cedrone, March 1984. (PB84 189885)

20 AFOS Hurricane Plotter. Charles Little, May 1984. (PB84 199629)

21 WARN - A Warning Formatter. Gerald G. Rigdon, June 1984. (PB84 204551)

22 Plotting TDL Coastal Wind Forecasts, Paula Severe, June 1984 (Revised) (PB84-220789)

23 Severe Weather Statistics STADTS Decoder (SWX) and Plotter (SWY), Hugh M. Stone, June 1984. (PB84-213693)

24 WXR, Harold Opitz, August 1984. (PB84-23722)

25 FTASUM: Aviation Forecast Summaries, Matthew Peroutka, August 1984. (PB85-112977)

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.

Information on availability of NOAA publications can be obtained from:

ENVIRONMENTAL SCIENCE INFORMATION CENTER (D822)
ENVIRONMENTAL DATA AND INFORMATION SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

6009 Executive Boulevard
Rockville, MD 20852