

H
QC
874.3
U63
no.6
c.1

NOAA Eastern Region Computer Programs
and Problems NWS ERCP - No. 6



REAL-TIME QUALITY CONTROL OF SAOs

Scientific Services Division
Eastern Region Headquarters
January 1983

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

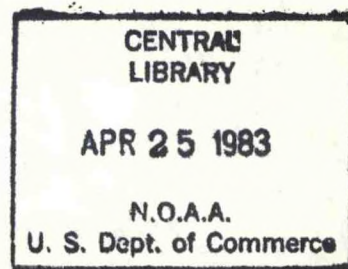
H
QC
874.3
U63
no. 6
c. 1

NOAA Eastern Region Computer Programs and Problems - No. 6

REAL-TIME QUALITY CONTROL OF SAOs
#

John A. Billet
National Weather Service Forecast Office
Charleston, West Virginia

Scientific Services Division
Eastern Region Headquarters
January 1983



UNITED STATES
DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Oceanic and
Atmospheric Administration
John V. Byrne, Administrator

National Weather
Service
Richard E. Hallgren, Director



REAL-TIME QUALITY CONTROL OF SAOs

John A. Billet
National Weather Service Forecast Office
Charleston, W. Va.

I. General Information

A. Summary

This program, SAOCHEC, examines an aviation observation for errors and, if it finds any, indicates what the errors are. If no errors are detected, the program sends the observation out over AFOS addressed to ALL. The program is designed to provide real-time quality control for observations.

B. Environment

The program will run on the Eclipse computer in the background partition of AFOS. The program was written in FORTRAN IV.

C. References

National Weather Service, 1982: Surface Observations, Federal Meteorological Handbook No. 1, National Oceanic and Atmospheric Administration, U.S. Department of Commerce.

II. Application

A. Complete Program Description

SAOCHEC checks an observation (in a workbin) on a real-time basis before it reaches the world. The program will handle specials or hourlies and performs a series of checks on both.

For both specials and hourlies the checks include:

1. If the observation is a valid type, i.e., RS, SA, SP.
2. If there is a time group.
3. Nonreportable cloud height.
4. Ceiling without a ceiling designator.
5. Obscuration reported above 4000 ft.
6. Nonreportable visibility.
7. Visibility less than 7 miles with no obstruction to vision.

8. Visibility greater than 6 miles with an obstruction to vision other than precipitation or thunder.
9. Wrong intensity of drizzle or snow with visibility reported.
10. Fog with dewpoint depression greater than 8 degrees.
11. Blowing snow with wind speed less than 9 knots.
12. If there is a garbling or misplacement in the cloud weather group.
13. If there is an error in the placement of the temperature or pressure group or if they appear garbled.
14. If there is a wind group in the observation.
15. If the wind speed and direction are reportable values.
16. If there is a garbling or misplacement in the altimeter group.
17. If there is a garbling or misplacement in the remarks section.

Additional checks for the hourlies include:

1. Pressure tendency on 3- or 6-hourly.
2. Min max group on 6-hourly.
3. Cloud group on 3- or 6-hourly.
4. If the temperature is a realistic value somewhere between 120 and -60.
5. If the dewpoint is a realistic value somewhere between 90 and -60.
6. If the dewpoint exceeds temperature.
7. If there is a change between the previous hour and the present hour of more than 10 degrees in temperature.
8. If there is a change between the previous hour and present hour of more than 10 degrees in dewpoint.
9. If there is a change between the previous hour and present hour of more than 3 millibars in the sea level pressure.
10. If there is a change between the previous hour and present hour of more than .10 inches in the altimeter setting.

Additional checks are made to determine if a nonspecial should be a special. Checks are made for the following criteria:

1. Thunderstorms beginning, ending, or changing intensity.
2. Hail begins or ends.

3. Freezing precipitation begins, ends, or changes intensity.
4. Ice pellets begin, end, or change intensity.
5. Changes in visibility which would require a special anywhere in the country. Local criteria are not included.
6. Changes in ceiling which would require a special anywhere in the country. Local criteria are not included.

If there is an error in the observation, an appropriate message is written to a file which will be stored in the AFOS database, and the console is alerted. The observation remains in the workbin where it can be corrected. If no errors are found, the program stores the observation under the correct key and transmits it to ALL. The temporary file created for the error messages is deleted if the program finishes successfully.

B. Machine Requirements

The program requires 20K of memory to execute. SAOCHEC.SV and .OL take up around 70 blocks of disk space. Generally, runtime is 20 seconds or less. At most three channels are opened at any one time during program execution.

C. Structure of Software

See Figure 1.

D. Database

The database is accessed for the workbin containing the current observation as well as the various previous versions of the observation used. The database is also accessed when the checked current observation is stored and transmitted, and when the error message product is stored.

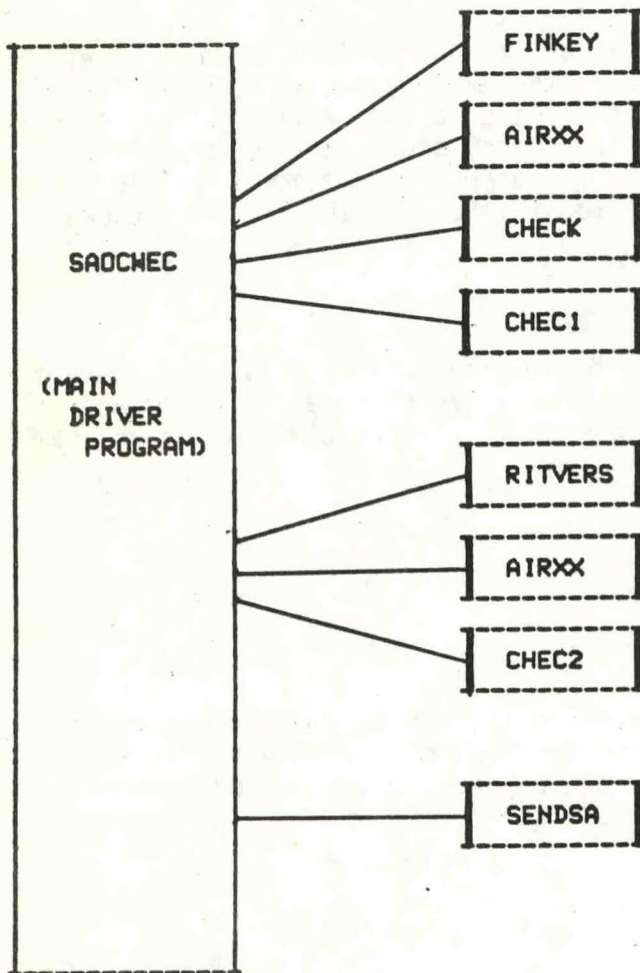
III. Procedures

A. Initiation of Program

Two products must be added to the database in order to run SAOCHEC. One is CCCWRKSAO, the workbin where the current observation is typed by the observer. The other is CCCWRKERR, where SAOCHEC will write the error messages.

To run the program each hour, write the current observation into CCCWRKSAO using message composition. Once the workbin is stored, RUN:SAOCHEC at the ADM. Within 20 seconds the console will be alerted, saying either that product CCCSAOXXX has been stored (no errors found) or that CCCWRKERR has been stored (errors found and noted in this product). The error messages can be automatically displayed if CCCWRKERR is alarmed or alerted for the console where SAOCHEC will normally be run.

Figure 2 shows a sample observation with several errors in a workbin, and the error messages SAOCHEC produced.



EXTRACT (FINKEY) AND DECODE (AIRXX) THE CURRENT OBSERVATION IN CCCURKSAO AND THE PREVIOUS OBSERVATION IN THE DATABASE. CHECK CHECKS THE CURRENT OB ITSELF, AND CHEC1 COMPARES THE CURRENT OB TO THE PREVIOUS ONE.

EXTRACT (RITVERS) AND DECODE (AIRXX) THE PREVIOUS HOURLY OBSERVATION AND COMPARE IT TO THE CURRENT OB (CHEC2).

IF NO ERRORS ARE FOUND, STORE THE CURRENT OB UNDER THE CORRECT KEY AND TRANSMIT (SENDSA). (IF THERE ARE ERRORS, SAOCHEC WILL WRITE ERROR MESSAGES TO CCCURKERR.)

Figure 1. Structure of software (with subroutine functions).

CRWRKSAO
LDUS00 KERH 291500
CRW SA 1351 20 BKN 4 174/49/19/1035/021

CRWRKERR
THE SAO ERRORS ARE
VIS LESS THAN 7 MILES AND NO OBST. TO VISION
CEILING WITHOUT A CEILING DESIGNAT
DP MIGHT BE WRONG

CRWSAOCRU
LDUS00 KCRW 291400
CRW SA 1350 M20 OVC 15 186/30/20/3107/007/HIR CLDS NW LB02E25

Figure 2. Sample observation with errors in workbin (CRWRKSAO), error messages that SAOCHEC produced (CRWRKERR), and previous observation in database for comparison.

B. Cautions or Restrictions on Program Use

1. SAOCHEC must be octal edited to fit it to each site. The following locations should be changed (CCC refers to the node id, XXX to the observation i.d.):

in SAOCHEC.SV

words 455-461 should contain CCCSAOXXX
words 462-466 should contain CCCWRKSAO
words 435-442 should contain CCCWRKERR
words 442-446 " " "

in SAOCHEC.OL

words 34400-34404 should contain CCCSAOXXX
words 34407-34413 " " "
words 34424-34425 should contain the XXX of the station

2. SAOCHEC will indicate an error in 3-hourly observations that are more than 1 line long even if there are no actual errors. It will also return an error message on 6-hourlies when a 4 group is used for the maximum and minimum temperatures.

C. Complete Program Listing

```

C   PROGRAM SAOCHEC  CHECKS SAOS
      INTEGER IBUF(60),TDATA(256),IBUF1(60),TDATA1(256),IBUF2(60)
      INTEGER ZTIME
      COMMON/BLK/KEY(5),JMSG(10)
      COMMON/BLKK/KEY1(5)
      COMMON/EBLK/JCH,IOF(5),IEHDR(11)
      EXTERNAL AIRXX,CHECK,CHEC1,SENDSA,OV1,OV2,OV3,OV4
      DATA KEY/"CRWJRKSAO"/ ;OBS TO BE SENT
      DATA KEY1/"CRWSAOCRW"/ ;OBS
      DATA IOF/"CRWJRKERR"/ ;ERROR PRODUCT
      DATA IEHDR/"CRWJRKERR000",-1,-1,2400K,142600K,6412K/
      CAN=0
      CALL GCHN(ICHN4,IER)
      CALL OVOPN(ICHN4,"SAOCHEC.OL",IER)
      CALL DELETE(IOF,IER)
      CALL CRAND(IOF,IER)
      CALL GCHN(JCH,IER)
      CALL OPENN(JCH,IOF,0,IER)
      CALL ERROR(IER,"OUTPUT FILE OPEN")
      CALL WRS(JCH,IEHDR,17,IER)
      CALL FINKEY(TDATA,TDATA1) ;FIND PREVIOUS OBS TDATA1 AND CURRENT TDATA
      CALL OVLOD(ICHN4,OV1,1,IER)
      CALL AIRXX(TDATA,IBUF,CAN,IER) ;DECODE CURRENT OBS
      IIER=IER
      CALL FOVRL(OV1,IER)
      CALL OVLOD(ICHN4,OV1,1,IER)
      CALL AIRXX(TDATA1,IBUF1,CAN,IER) ;DECODE PREVIOUS OBS
      CALL FOVRL(OV1,IER)
      CALL OVLOD(ICHN4,OV2,1,IER)
      CALL CHECK(IBUF,IIER,KN) ;CHECKS CURRENT OBS FOR ERRORS
      CALL FOVRL(OV2,IER)
      KN1=0
      IF(IBUF(6).GT.1) GO TO 50
      CALL OVLOD(ICHN4,OV3,1,IER)
      CALL CHEC1(IBUF,IBUF1,KN1) ;CHECKS CURRENT OBS AGST PREV OBS
      CALL FOVRL(OV3,IER)
50  CONTINUE
      KN3=0
      IF(IBUF(6).GT.2) GO TO 60
      ZTIME=IBUF(5)-100 ;LOOKING FOR LAST HRLY TIME
      IF(ZTIME.LT.0) ZTIME=2360+ZTIME
      CALL RITVERS(ZTIME,TDATA,IER) ;LOOKING FOR LAST HRLY
      CALL OVLOD(ICHN4,OV1,1,IER)
      CALL AIRXX(TDATA,IBUF2,CAN,IER) ;DECODE LAST HRLY
      CALL FOVRL(OV1,IER)
      CALL CHEC2(IBUF,IBUF2,KN3) CHECK CUR AGST PREV HRLY
60  CONTINUE
      CALL OVLOD(ICHN4,OV4,1,IER)
      IF(KN.EQ.1.AND.KN1.EQ.0.AND.KN3.EQ.0) CALL SENDSA ;IF OBS OK SEND
      CALL FOVRL(OV4,IER)
      IF(KN.EQ.1.AND.KN1.EQ.0.AND.KN3.EQ.0) GO TO 30
      CALL WRS(JCH,203K,2,IER)
      CALL KLOSE(JCH,IER)
      CALL FSTORE(IOF,0,IER)
      CALL WAIT(1,2,IER)

```

```
CALL FORKP("SAOCHEC", IOF, IER)
GO TO 40
30 CONTINUE
CALL KLOSE(JCH, IER)
40 CALL DFILW(IOF, IER)
STOP
END
```

```
C TDATA IS CURRENT OBS TDATA1 IS PREVIOUS OBS
C FINKEY WILL FIND CORRECT KEY
SUBROUTINE FINKEY(TDATA, TDATA1)
INTEGER TDATA(256), IDATA(128), KEYREC(15), TDATA1(256), KEY1REC(15)
COMMON/BLK/ KEY(5), JMSG(10)
COMMON/BLKK/ KEY1(5)
DATA JMSG /"PRODUCT SEARCH ERROR"/
CALL KSRCF(KEY, KEYREC, IER)
CALL ERROR(IER, JMSG)
CALL RDBKF(0, IDATA, IER)
CALL ERROR(IER, JMSG)
CALL UNPACK(IDATA, 256, TDATA)
CALL KSRCF(KEY1, KEY1REC, IER)
CALL ERROR(IER, JMSG)
CALL RDBKF(0, IDATA, IER)
CALL ERROR(IER, JMSG)
CALL UNPACK(IDATA, 256, TDATA1)
RETURN
END
```

C
C PROGRAMMER - RICH THOMAS SXB, ISL, SDO 9/79
C

C THIS SUBROUTINE ACCEPTS AN ARRAY(TDATA(256)) WHICH CONTAINS AN
C UNPACKED SAO WHICH WAS READ IN FROM THE AFOS DATA BASE BY THE CALLING PROGRAM.
C THE OBSERVATION IS DECODED AND RETURNED IN ARRAY IBUF(60) -MIXED INTEGER/ASCII FORM.
C INPUT ARGUMENTS:

C TDATA -ARRAY CONTAINING RAW SAO
C IBUF -RETURNED ARRAY CONTAINING DECODED SAO
C CAN -FLAG INDICATING CANADIAN SAO (1=CAN.)(0=US)
C IER -RETURN CODE
C

C THIS SUBROUTINE REQUIRES SUBROUTINES:

C ANDEQ
C ANDGO
C ORGO
C NUMBR
C OVERLAY OV1
C

C PARAMETER CHBL0=23 ;NUMBER OF WORDS IN BLK 0 - SEE FPAFOS.FR
C SUBROUTINE AIRXX(TDATA,IBUF,CAN,IER)

C REAL LEVAP

C INTEGER V1,V2,VX,VY,P,Q,R,T,HGT(3)

C INTEGER TDATA(256),IBUF(60),CX

C INTEGER CP,CP1,CP2,CP3,CP4,CAN

C INUM(IA,IB,IC,ID)=(IA-48)*1000+(IB-48)*100+(IC-48)*10+
C 1ID-48 ; THIS FUNCTION CONVERTS ASCII TO INTEGER

C BEGIN DECODING

C IER=0

C INITIALIZE ARRAY TO BE USED FOR FORMATTED DATA

C DO 80 I=1,60

80 IBUF(I)=-99 ; -99 DENOTES MISSING FOR INTEGERS

C DO 85 N=1,4

85 IBUF(N)=32 ; ASCII BLANKS

C DO 90 N=24,33

90 IBUF(N)=32

C DO 95 N=44,49

95 IBUF(N)=32

C DO 100 N=10,18,4

IBUF(N)=32

IBUF(N+1)=0

IBUF(N+2)=0

100 IBUF(N+3)=32

IBUF(23)=32

IBUF(53)=32

IBUF(54)=32

IBUF(55)=32

M=0

IC=CHBL0+21

JC=CHBL0+31

DO 110 CP=IC,JC ; CP IS CHARACTER POINTER IN TDATA

IF(TDATA(CP).GT.47)GO TO 105 ;LETTER OR NUMBER (ID)

IF(M.GT.0)GO TO 115 ; IF M=0 NO LETTERS/NUMBERS FOUND SO FAR
GO TO 110

105 M=M+1 ; INCREMENT INDEX - LETTER/NUMBER FOUND

IBUF(M)=TDATA(CP) ;LOAD INTO IBUF

110 CONTINUE

IER=4 ; IER=4 ID NOT FOUND;(OR UNKNOWN OB TYPE)

RETURN

C CP IS LOCATION OF BLANK AFTER ID

```

C   GET OB TYPE
115 J=CP+1      ; ADVANCE ONE CHARACTER POSITION
    K=J+8      ; SEARCH FOR OB TYPE IS LIMITED TO NEXT 8 CHARACTERS
    IBUF(6)=0  ; INITIALIZE TO 0
    DO 120 CP=J,K
    CP1=CP+1
    ITM=TDATA(CP)
    ITM2=TDATA(CP1)
    IF(ITM.EQ.83)GO TO 130      ; S- TYPE
    VX=82      ; 'R'
    VY=83      ; 'S'
    CALL ANDGO(ITM,VX,ITM2,VY,$125) ; RS TYPE
    V2=4
    VY=65
    CALL ANDEQ(ITM,VX,ITM2,VY,IBUF(6),V2) ; UNMANNED RAMOS
    IF(ITM.EQ.65)IBUF(6)=4      ; UNMANNED AMOS OR AUTOB
    IF(ITM.EQ.32)GO TO 120
    GO TO 135
120 CONTINUE
125 IBUF(6)=2    ;RS
130 IF(ITM2.EQ.65)IBUF(6)=1      ;SA
    IF(ITM2.EQ.80)IBUF(6)=3      ;SP
    IF(ITM2.EQ.87)IBUF(6)=6      ;SW
135 IF(IBUF(6).EQ.0)IER=4        ; IER=4 UNKNOWN OB TYPE;(OR ID NOT FOUND)
    MINCP=CP1+3 ;SET MIN CP FOR RAMOS/AMOS (UNMANNED)
    IF(IBUF(6).EQ.4)GO TO 255     ;NO TIME IN UNMANNED AMOS/RAMOS OBS
    J=CP+2
    K=J+8
    DO 200 CP=J,K
    CP1=CP+1
    CP2=CP+2
    CP3=CP+3
    CALL NUMBR(TDATA(CP),$200,$210) ; CHECK FOR A 4 NUMBER SEQUENCE
200 CONTINUE
205 IER=5      ; IER=5 ERROR IN LOCATING TIME
    GO TO 270
210 DO 220 JCP=CP1,CP3
    IF(TDATA(JCP)-48)205,215,215 ;BETWEEN 0
215 IF(TDATA(JCP)-57)220,220,205 ; AND 9-?
220 CONTINUE
    IBUF(5)=INUM(TDATA(CP),TDATA(CP1),TDATA(CP2),TDATA(CP3)) ;TIME
    IHOURL=0
    IHOURL=IBUF(5)/100
    IF(IBUF(5)-IHOURL*100.GE.45)IHOURL=IHOURL+1 ; COVER CASES WHERE TIME IS JUST BEFORE HOUR
    J=CP3+1
    K=J+8
    DO 230 CP=J,K ;CHECK FOR RAMOS/AMOS
    IF(TDATA(CP).NE.32)GO TO 235
230 CONTINUE
    MINCP=CP3
    GO TO 205
235 VX=82      ; 'R'
    VY=65      ; 'A' RAMOS
    V1=TDATA(CP)
    CALL ORGO(V1,VX,V1,VY,$240) ; 'R--' OR 'A---' ? IF SO GO TO 240
    MINCP=CP-1
    GO TO 255
240 IBUF(6)=5 ; MANNED RAMOS/AMOS TYPE
    J=CP+1
    K=J+5

```

```

DO 245 CP=J,K
IF(TDATA(CP).EQ.32)GO TO 250 ; FIND FIRST SPACE PAST 'RAMDS'
245 CONTINUE
GO TO 205
250 MINCP=CP ;SET MIN CHARACTER POSITION
255 CONTINUE
C SEARCH FOR WIND GROUP
GO TO 270
265 IER=6 ; IER=6 COULDN'T FIND WIND GROUP
CP=MINCP+1
CX=CP+50
JXT=24
DO 266 JX=CP,CX
IF(TDATA(JX).EQ.32)GO TO 266
IF(TDATA(JX).LT.32)GO TO 267
IF(TDATA(JX).EQ.131)GO TO 267
IBUF(JXT)=TDATA(JX)
JXT=JXT+1
IF(JXT.GT.33)GO TO 267
266 CONTINUE
267 RETURN
270 CP=CP+1
IF(CP.GT.256)GO TO 265 ;OUT OF ARRAY
IF(TDATA(CP).EQ.131)GO TO 265 ;END OF MESSAGE
IF(TDATA(CP).NE.47)GO TO 270 ; SLASH?
IY=CP ;HOOK FOR ALT. SETTING AND REMARKS POINTS TO SLASH AFTER WIND
275 IY1=CP-1
IY2=CP-2
IY3=CP-3
IY4=CP-4
V1=TDATA(IY3)
VX=81
VY=71
CALL ORGO(V1,VX,V1,VY,$285) ; G OR Q INDICATOR
DO 280 IX=IY4,IY1
CALL NUMBR(TDATA(IX),$282,$280) ;CHECK IF ALL 4 ARE NUMBERS
280 CONTINUE
IA=48
IB=48
IBUF(39)=INUM(IA,IB,TDATA(IY2),TDATA(IY1)) ;WIND SPEED
ID=48
IBUF(38)=INUM(IA,TDATA(IY4),TDATA(IY3),ID) ;WIND DIRECTION
GO TO 295
282 DO 283 IX=IY4,IY1
IF(TDATA(IX).NE.'<0>M')GO TO 270
283 CONTINUE
GO TO 295
285 DO 290 IX=IY2,IY1
CALL NUMBR(TDATA(IX),$270,$290) ;CHECK IF THERE ARE 2 NUMBERS AFTER 'G' OR 'Q'
290 CONTINUE
IA=48
IB=48
IBUF(40)=INUM(IA,IB,TDATA(IY2),TDATA(IY1)) ;WIND GUSTS
CP=CP -3 ; RESET CHARACTER POSITION
GO TO 275
C TEMPERATURE/DEWPOINT SEARCHES
295 CP=CP-5 ;CP IS AT SLASH AFTER WIND
IF(TDATA(CP).EQ.69)CP=CP-1 ;'E' FOR ESTIMATED WIND (SKIP OVER)
M=36 ;DEWPOINT DEG F
N=43 ;DEWPOINT DEG C

```

```

CONST=5./9.
IF(TDATA(CP).NE.47)GO TO 365 ; CHECK TO BE SURE ITS A SLASH
CP=CP-1
IX=CP+1
JSIGN=0
300 IF(TDATA(CP).EQ.77)GO TO 375 ;MISSING (M)
305 IX=IX-1
ITM=TDATA(IX)
IF(ITM-48)315,310,310 ;BETWEEN 0 AND 9?
310 IF(ITM-57)305,305,315
315 IF(ITM.EQ.45)JSIGN=1 ;NEGATIVE?
IF(ITM.EQ.45)GO TO 305
IF(CP.EQ.IX)GO TO 370 ; NO TEMP - UNEXPECTED CHARACTER
IX=IX+JSIGN
I=CP-IX
IF(I-3)320,320,370
320 IF(I-1)370,325,325 ;DETERMINE HOW MANY CHARACTERS IN TEMP
325 P=48 ;SET TO '0'
Q=48
R=48
GO TO (340,335,330)I
330 I=CP-2
P=TDATA(I)
335 I=CP-1
Q=TDATA(I)
340 R=TDATA(CP)
IA=48
IBUF(M)=INUM(IA,P,Q,R) ;CONVERT TO INTEGER - TEMP/DEWPT
IF(JSIGN.EQ.1)IBUF(M)=-IBUF(M) ; NEGATIVE
IF(CAN.EQ.0)GO TO 345 ; US DATA - CONVRT FOR CELCIUS
IBUF(N)=IBUF(M) ;CANADIANS - CONVERT FOR FAHRENHEIT
IBUF(M)=IBUF(M)*100
Q=IBUF(M)/5*9
P=Q
Q=Q/100
R=P-Q
IF(R.GT.50)Q=Q+1
IBUF(M)=Q+32
GO TO 355
345 IF(IBUF(M).LT.0)GO TO 350 ; DEG F TO DEG C
IBUF(N)=IFIX(((IBUF(M)-32.)*CONST)+.5)
GO TO 355
350 IBUF(N)=IFIX(((IBUF(M)-32.)*CONST)-.5)
355 IF(M.EQ.35)GO TO 385 ;ALREADY DID DEWPOINT
360 IX=IX-JSIGN ;RESET FOR DEWPOINT
M=35 ;TEMP DEG F
N=42 ;TEMP DEG C
JSIGN=0
IF(TDATA(IX).NE.47)GO TO 370 ;SHOULD BE A SLASH
CP=IX-1 ;SLASH BETWEEN TEMP AND DEWPT = IX
GO TO 300
365 V1=TDATA(CP)
V2=IBUF(6) ;CHECK IF THERE IS A BLANK BEFORE WIND
VX=32 ;AND IT IS AN SP TYPE
VY=3
CALL ANDGO(V1,VX,V2,VY,$420) ; GO TO OB VIS
370 IER=7 ; IER =7 - TEMP/PRESSURE DECODE ERRORS
RETURN
375 CP=CP-1
IF(TDATA(CP).NE.47)GO TO 380 ;SHOULD BE A SLASH (IF THERE IS A PRESSURE)

```

```

IF(M.EQ.35)GO TO 390
IX=CP
GO TO 360
380 VX=32 ;BLANK
VY=35 ; '-'
CALL ANDGO(TDATA(CP),VX,M,VY,$420)
IF(TDATA(CP).EQ.77)GO TO 375 ;ADDITIONAL M'S
GO TO 370
385 IX=IX-JSIGN
CP=IX
ITM=TDATA(CP)
IF(ITM.EQ.32)GO TO 420 ;SKIP PRESSURE
IF(ITM.EQ.47)GO TO 390 ;CP AT SLASH BEFORE TEMP
GO TO 370
C SL PRESSURE (CP IS AT SLASH BEFORE TEMP)
390 J1=CP
395 J2=CP-1
IF(TDATA(J2).EQ.32)GO TO 400 ; LOOK FOR FIRST BLANK BEFORE SLASH
IF(TDATA(J2).EQ.69)GO TO 400 ;EST. PRESSURE
CP=CP-1
GO TO 395
400 IF(J1-CP-3)405,410,370 ; SHOULD HAVE BEEN 3 NUMBERS
405 IF(J1.EQ.CP) GO TO 420 ; NO PRESSURE
GO TO 370
410 J1=CP+1
J2=CP+2
IA=48
IBUF(34)=INUM(IA,TDATA(CP),TDATA(J1),TDATA(J2)) ;CONVERT ASCII PRESSURE TO INTEGER
ITM=IBUF(34)
IF(ITM.GE.500)IBUF(34)=ITM+9000
IF(ITM.LT.500)IBUF(34)=ITM+10000
415 CP=CP-1
C POINTER NOW AT SPACE BEFORE PRESS (OR TEMP IF NO PRES)
C DECODE OBSTRUCTION TO VISION
420 IF(IBUF(6).EQ.4)GO TO 455 ;UNMANNED - SKIP TO ALT. SETTING
J1=CP
DO 440 J2=0,9
IF(J1.LE.MINCP)GO TO 455 ;SKIP TO ALT SETTING
IF(TDATA(J1).EQ.32)GO TO 430
IF(TDATA(J1).EQ.69)GO TO 430 ;EST. PRESSURE
IF(TDATA(J1)-48)435,425,425 ;BETWEEN 0 AND 9 ?
425 IF(TDATA(J1)-57)444,444,435 ;NUMBER WOULD BE VISIBILITY
430 IF(J2.EQ.0)GO TO 415
435 ITM=TDATA(J1)
IF(TDATA(J1).EQ.86)GO TO 445 ;VARIABLE VISIBILITY
IBUF(33-J2)=ITM ;LOAD OBSTRUCTIONS TO VISION
J1=J1-1
440 CONTINUE
GO TO 460
C DECODE VISIBILITY
444 IBUF(23)=32
GO TO 468
445 IBUF(23)=32
IF(TDATA(J1).EQ.86)IBUF(23)=86 ; 'V' FOR VARIABLE
IF(IBUF(23).EQ.32)GO TO 468
447 J1=J1-1
R=48 ;'0'
P=48 ;'0'
ITM=TDATA(J1)
CALL NUMBR(ITM,$450,$468)

```



```

450 IF(TDATA(J1).EQ.32)GO TO 447
    GO TO 445
455 GO TO 715
460 IER=8
    J1=J1+1
461 J1=J1-1
    IF(TDATA(J1).EQ.86)GO TO 463
    IF(TDATA(J1)-48)461,462,462
462 IF(TDATA(J1)-57)463,463,461
463 J3=J1+10
    J2=J1+1
    JX=24
    DO 464 J4=J2,J3
    IBUF(JX)=TDATA(J4)
    JX=JX+1
464 CONTINUE
    GO TO 445
468 R=48
    Q=48
    P=48
    JB=J1 ;BEGINNING CHARACTER PSN
    JE=J1 ;ENDING CHARACTER PSN
    R=TDATA(J1) ;RIGHT MOST NUMBER
    J1=J1-1
    ITM=TDATA(J1)
    CALL NUMBR(ITM,$510,$500)
500 JB=J1
    P=TDATA(J1)
    J1=J1-1
    ITM=TDATA(J1)
    CALL NUMBR(ITM,$510,$504)
504 JB=J1
    Q=TDATA(J1)
    J1=J1-1
    ITM=TDATA(J1)
    CALL NUMBR(ITM,$510,$505)
505 IF(ITM-67)506,557,557
506 IER=8
    GO TO 715
510 IF(ITM.EQ.32)GO TO 545 ;ONE OR TWO DIGIT VISIBILITY
    IF(ITM.NE.47)GO TO 544 ;DEALING WITH A FRACTION
    IA=48
    IB=48
    IDN=INUM(IA,IB,P,R) ;DENOMINATOR
    J1=J1-1
    ITM=TDATA(J1)
    CALL NUMBR(ITM,$505,$515)
515 N=ITM-48 ;NUMERATOR
    IDN=(N*1000)/IDN ;FRACTION (THOUSENTHS)
    J1=J1-1
    ITM=TDATA(J1)
    CALL NUMBR(ITM,$525,$520)
520 N=(ITM-48)*1000
    JB=J1 ;RESET BEGINNING PSN (NO SPACE SEPARATING WHOLE DIGIT & FRACTION)
    GO TO 540
525 IF(ITM.NE.32)GO TO 535
    J1=J1-1
    ITM=TDATA(J1)
    CALL NUMBR(ITM,$535,$530)
530 N=(ITM-48)*1000 ;WHOLE DIGIT - 2 SPACES BEFORE FRACTION (SPACE SEPARATOR)

```

```

      JB=J1
      GO TO 540
535 N=0
      JB=J1+1
      J1=J1+1
540 IBUF(22)=-((N+IDN)           ;STORE INTEGR VALUE OF VIS. - NEG INDICATES THOUSENTHS OF MILES
      GO TO 550
544 J1=J1+1
545 IBUF(22)=-((Q-48)*100+(P-48)*10+(R-48)) ;CONVERT VISIBILITY TO INTEGER
550 M=44
      DO 555 I=JB,JE
      IBUF(M)=TDATA(I)           ;LOAD IN ASCII STRING FOR VISIBILITY
555 M=M+1
C   CP (=J1) IS PSN OF SPACE BEFORE VIS OR FIRST CHAR OF VIS
C   DECODE UP TO 3 CLOUD LAYERS
      GO TO 558
557 J1=J1+1
558 M=10
      I=1
560 IF(I.GT.3)GO TO 710           ; 1 PASS FOR EACH OF 3 CLOUD LAYERS
565 J1=J1-1
570 IF(J1.LT.MINCP)GO TO 715     ; BEGINNING OF OB FOUND
      DO 575 J=1,3
575 HGT(J)=48
      ITM=TDATA(J1)
      IF(ITM-67)565,600,580       ;(OV)C
580 IF(ITM-70)710,600,585       ;(BK)N
585 IF(ITM-82)710,600,590       ;(CL)R
590 IF(ITM-84)710,600,595       ;(SC)T
595 IF(ITM-88)710,605,710       ;X
600 J1=J1-2
605 IBUF(M)=TDATA(J1)
610 J1=J1-1
      IF(TDATA(J1)-45)620,615,630
615 IBUF(M+1)=1                 ; FLAG FOR 'THIN' OR 'PARTIAL'
      GO TO 610
620 IF(TDATA(J1)-32)630,610,630
625 J1=J1-1
630 DO 655 J=1,3
      IF(TDATA(J1)-48)660,635,635 ;BETWEEN 0 AND 9 ?
635 IF(TDATA(J1)-57)645,645,640
640 IF(TDATA(J1).EQ.86)IBUF(M+1)=IBUF(M+1)+50 ;'VARIABLE' FLAG
      IF(TDATA(J1).EQ.86)GO TO 625
      GO TO 660
645 HGT(J)=TDATA(J1)
      IF(J1-MINCP)715,660,650
650 J1=J1-1
655 CONTINUE
660 IA=48
      IBUF(M+2)=INUM(IA,HGT(3),HGT(2),HGT(1)) ;COMPUTE HEIGHT
665 IF(TDATA(J1)-32)670,700,675
670 IF(J1.EQ.MINCP)GO TO 715
      J1=J1+1
      GO TO 705
675 ITM=TDATA(J1)
      IF(ITM-69)695,690,680       ;'E'
680 IF(ITM-77)695,690,685       ;'M'
685 IF(ITM-87)695,690,695       ;'W'
690 IBUF(M+3)=TDATA(J1)         ;LOAD CEILING INDICATOR
      J1=J1-1

```

```

GO TO 665
695 I=I+1
M=M+4 ;SET FOR NEXT CLOUD GROUP
GO TO 570 ;ALREADY AT ANOTHER CHARACTER
700 IF(J1-MINCP)715,715,701 ;NO CEILING
701 J1=J1-1
ITM=TDATA(J1)
IF(ITM.EQ.69.OR.ITM.EQ.77.OR.ITM.EQ.87)GO TO 675
J1=J1+1
705 M=M+4 ;SET FOR NEXT CLOUD GROUP
IF(I.EQ.3)GO TO 715 ;AT A SPACE NOW
I=I+1
GO TO 560
C GET ALTIMETER SETTING (IY WAS PSN OF SLASH AFTER WIND GROUP
710 IER=8 ; IER = 8 CLOUDS/WEATHER/VISIBILITY ERROR
715 IF(TDATA(IY+1)-77)720,740,835 ;CHECK FOR MISSING
720 DO 730 J=1,3
IF(TDATA(IY+J)-48)735,725,725 ;CHECK TO MAKE SURE NEXT 3 CHARACTERS ARE NUMBERS
725 IF(TDATA(IY+J)-57)730,730,735
730 CONTINUE
IA=48
IBUF(41)=INUM(IA,TDATA(IY+1),TDATA(IY+2),TDATA(IY+3)) ;CALCULATE ALT. SETTING
IF(IBUF(41).GT.450)IBUF(41)=IBUF(41)+2000
IF(IBUF(41).LE.450)IBUF(41)=IBUF(41)+3000
GO TO 740
735 IER=10 ;IER=10 ALTIMETER GROUP ERROR
GO TO 835
740 IF(IHOUR/3*3.NE.IHOUR)GO TO 835 ;BYPASS REMARKS IF NOT 3 HRLY
IA=1
IB=2 ;BYPASS REMARKS IF NOT AN SA OR RS
CALL ORGO(IBUF(6),IA,IBUF(6),IB,$745)
GO TO 835
C FIND APP GROUP
745 CP=IY+4
750 DO 825 IY=CP,256
ITM=TDATA(IY)
IF(ITM.EQ.131)GO TO 835 ;END OFOB
IF(ITM.NE.32)GO TO 825
CALL NUMBR(TDATA(IY+1),$825,$755)
755 ITM=TDATA(IY+2)
CALL NUMBR(ITM,$760,$765)
760 IA=49
IB=47
CALL ANDGO(TDATA(IY+1),IA,ITM,IB,$805) ; '1/' COMBINATION
GO TO 825
765 ITM=TDATA(IY+3)
CALL NUMBR(ITM,$770,$775)
770 IF(ITM.EQ.32)GO TO 830
IF(ITM.GT.125)GO TO 830 ;SPECIAL CHARACTER (END OF OB)
IF(ITM.LT.20)GO TO 830 ; CR LF (END OF OB)
IA=49
IB=47
CALL ANDGO(TDATA(IY+1),IA,ITM,IB,$805) ; '1*' SEQUENCE
GO TO 825
775 ITM=TDATA(IY+4)
CALL NUMBR(ITM,$785,$780)
780 CALL NUMBR(TDATA(IY+5),$815,$790)
785 IF(ITM.EQ.47)GO TO 820 ; ***/ SEQUENCE
IF(ITM-32)790,790,825
IF(ITM-90)825,825,790

```

```

790 IF (IBUF(50).NE.-99)GO TO 825
   IBUF(50)=TDATA(IY+1)-48      ;PRESSURE CHARACTERISTIC
   IA=48
   IB=48
   IBUF(51)=INUM(IA, IB, TDATA(IY+2), TDATA(IY+3)) ;PRESSURE TENDENCY
   CP=IY+4
   CALL NUMBR(TDATA(CP), $750, $795)
795 IA=48
   IB=48
   IBUF(52)=INUM(IA, IB, TDATA(CP), TDATA(CP+1)) ;PRECIP
   ITM=TDATA(CP+3)
   IF (ITM.LE.57)GO TO 800 ;NUMBER
   ITM1=IBUF(52)
   IF (ITM.EQ.79) IBUF(52)=ITM1+100 ; 'ONE'      INCHES OF PRECIP
   ITM2=TDATA(CP+4)
   IA=84
   IB=87
   IV2=ITM1+200
   CALL ANDEQ(ITM, IA, ITM2, IB, IBUF(52), IV2) ; 'TWO'
   IB=72
   IV2=ITM1+300
   CALL ANDEQ(ITM, IA, ITM2, IB, IBUF(52), IV2) ; 'THREE'
   IA=70
   IB=79
   IV2=ITM1+400
   CALL ANDEQ(ITM, IA, ITM2, IB, IBUF(52), IV2) ; 'FOUR'
   IB=73
   IV2=ITM1+500
   CALL ANDEQ(ITM, IA, ITM2, IB, IBUF(52), IV2) ; 'FIVE'
   IF (IBUF(52).GT.99)GO TO 800
   IER=11      ;      IER=11      REMARKS ERROR
   GO TO 835
800 CP=CP+2
C   LOOK FOR CLOUD GROUP
   GO TO 750
805 IB=47
   CALL ANDGO(TDATA(IY+3), IB, TDATA(IY+4), IB, $810) ; '---//' SEQUENCE
   GO TO 825
810 IF (IBUF(53).NE.32)GO TO 825
   IBUF(53)=TDATA(IY+2) ;LOW CLOUDS
   IBUF(54)=TDATA(IY+3) ;MID CLOUDS
   IBUF(55)=TDATA(IY+4) ;HIGH CLOUDS
   CP=IY+5
   GO TO 750 ;LOOK FOR MAX/MIN TEMP
815 IF (TDATA(IY+5)-32)820,820,816
816 IF (TDATA(IY+5)-90)817,817,820
817 IER=11      ;      IER=11      REMARKS ERROR
   GO TO 835
820 IF (TDATA(IY+1).EQ.49)GO TO 810
825 CONTINUE
   GO TO 835
830 IF (IBUF(56).NE.-99)GO TO 835
   IA=48
   IB=48
   IBUF(56)=INUM(IA, IB, TDATA(IY+1), TDATA(IY+2)) ;MAX/MIN
C   MIXING RATIO CALCULATION
835 TX=IBUF(42)
   TD=IBUF(43)
   IF (TX.LT.-60.0.OR.TX.GT.60.)GO TO 840
   IF (TD.LT.-60.0.OR.TD.GT.60.)GO TO 840

```

```

IF(IBUF(34).LT.9000.OR.IBUF(34).GT.11000)GO TO 840
PRES=FLOAT(IBUF(34))/10.
LEVAP=597.3-.566*TX
E=6.11*EXP(9.045*LEVAP*(1./273.-1./(TD+273)))
QGKG=.622*E/(PRES-E)*1000.
IBUF(37)=IFIX(QGKG*10.+5)
840 IF(IER.EQ.0)IER=1
RETURN
END

```

```

C THIS SUBROUTINE IS JUST A LOGICAL IF STATEMENT
C IF A=B AND C=D THEN IV1=IV2
C USED TO SAVE DISK SPACE/CORE REQUIREMENT -
C
C PROGRAMMER-RICH THOMAS SXB,ISL,SDO 7/79 (SAODECODER)
C

```

```

SUBROUTINE ANDEQ(A,B,C,D,IV1,IV2)
INTEGER A,B,C,D
IF(A.EQ.B.AND.C.EQ.D)IV1=IV2
RETURN
END

```

```

C THIS SUBROUTINE IS A LOGICAL IF STATEMENT
C IF A=B AND C=D THEN GO TO STATEMENT * ISTN IN
C CALLING PROGRAM. USED TO CUT DOWN PROGRAM SIZE
C BY SUBROUTINING AN OPERATION REPEATED MANY TIMES.
C PROGRAMMER -RICH THOMAS SXB,ISL,SDO 7/79 (SAODECODER)
C SUBROUTINE ANDGO(A,B,C,D,ISTN)
C INTEGER A,B,C,D
C IF(A.EQ.B.AND.C.EQ.D)RETURN ISTN
C RETURN
C END

```

```

C THIS SUBROUTINE DETERMINES IF 'A' IS A NUMBER AND
C RETURNS TO STATEMENT * B IN CALLING PROGRAM IF NOT.
C RETURNS TO STATEMENT * C IN CALLING PROGRAM IF IT IS
C
C PROGRAMMER-RICH THOMAS SXB,ISL,SDO 9/79 (SAODECODER)
C

```

```

SUBROUTINE NUMBR(A,B,C)
INTEGER A,B,C
IF(A-48)900,901,901
901 IF(A-57)902,902,900
900 RETURN B ;NOT A NUMBER
902 RETURN C ;0-9
END

```

```

C THIS SUBROUTINE IS A LOGICAL IF STATEMENT
C IF A=B OR C=D GO TO STATEMENT * ISTN
C IN CALLING PROGRAM. USED TO SAVE SPACE/CORE.
C
C PROGRAMMER-RICH THOMAS SXB,ISL,SDO 9/79 (SAODECODER)
C

```

```

SUBROUTINE ORGO(A,B,C,D,ISTN)
INTEGER A,B,C,D
IF(A.EQ.B.OR.C.EQ.D)RETURN ISTN
RETURN
END

```

```

C      THIS SUBROUTINE CHECKS SA FOR ERRORS
      OVERLAY OV2
      SUBROUTINE CHECK(IBUF, IIER, KN)
      COMMON/EBLK/JCH, IOF(5), IEHDR(11)
      INTEGER IBUF(60), IIBUF(3)
      COMMON/BLK1/ JMSG1(16), JMSG2(11), JMSG3(21), JMSG4(21)
      1, JMSG5(24), JMSG6(20), JMSG7(13), JMSG9(23), JMSG10(23)
      2, JMSG11(13), JMSG12(18), JMSG13(16), JMSG14(22), JMSG15(18)
      3, JMSG16(15), JMSG17(12), JMSG18(14), JMSG19(12)
      COMMON/JBLK/JMSG21(17), JMSG22(11), JMSG23(10), JMSG24(9)
      1, JMSG25(6), JMSG26(18), JMSG27(12)
      DATA JMSG7/" NONREPORTABLE CLOUD HGT"/
      DATA JMSG11/" NONREPORTABLE VISIBILITY"/
      DATA JMSG12/" WRONG INTENSITY OF L OR S WITH VIS"/
      DATA JMSG9/" VIS GREATER THAN 6 MILES WITH OBST TO VISION"/
      DATA JMSG10/" BLOWING SNOW WITH WIND SPEED LESS THAN 9 MPH"/
      DATA JMSG1/" NO PRESSURE TENDENCY ON 3HRLY"/
      DATA JMSG2/" NO MIN,MAX ON 6HRLY"/
      DATA JMSG3/" FOG WITH DP DEP. GREATER THAN 8 DEGREES"/
      DATA JMSG4/" NO CLD GROUP ON 3/6HRLY IF CLDS REPORTED"/
      DATA JMSG5/" VIS LESS THAN 7 MILES AND NO OBST. TO VISION"/
      DATA JMSG6/" CEILING WITHOUT A CEILING DESIGNATOR"/
      DATA JMSG13/" UNKNOWN OB TYPE OB NOT DECODED"/
      DATA JMSG14/" COULD NOT LOCATE WIND GROUP OB NOT DECODED"/
      DATA JMSG15/" TEMP PRESSURE ERROR OB NOT DECODED"/
      DATA JMSG16/" REMARKS ERROR OB NOT DECODED"/
      DATA JMSG17/" ERROR IN LOCATING TIME"/
      DATA JMSG18/" CLOUD WEATHER GROUPS ERROR"/
      DATA JMSG19/" ALTIMETER GROUP ERROR"/
      DATA JMSG21/" STATION PRESSURE COULD BE WRONG"/
      DATA JMSG22/" TEMP COULD BE WRONG"/
      DATA JMSG23/" DP COULD BE WRONG"/
      DATA JMSG24/" DP EXCEEDS TEMP"/
      DATA JMSG25/" WIND WRONG"/
      DATA JMSG26/" OBSCURATION REPORTED ABOVE 4000 FT"/
      DATA JMSG27/" THE SAO ERRORS ARE"/
      CALL WRS(JCH, JMSG27, 23, IER)
      CALL WRS(JCH, 6400K, 1, IER)
      N=0
      NN=1
      IF(IIER.EQ.4) N=12 ;CHECKS IF KNOWN OB TYPE
      IF(IIER.EQ.6) N=13 ;CHECKS IF WIND GROUP
      IF(IIER.EQ.7) N=14 ;CHECKS IF TEMP OR PRES GROUP
      IF(IIER.EQ.11) N=15 ;CHECKS IF REMARK ERROR
      IF(N.EQ.12)CALL WRS(JCH, JMSG13, 32, IER)
      IF(N.EQ.13)CALL WRS(JCH, JMSG14, 44, IER)
      IF(N.EQ.14)CALL WRS(JCH, JMSG15, 36, IER)
      IF(N.EQ.15)CALL WRS(JCH, JMSG16, 30, IER)
      IF(N.GE.12.AND.N.LE.15)CALL WRS(JCH, 6400K, 1, IER)
      IF(N.EQ.12.AND.N.LE.15) GO TO 350
      IF(IIER.EQ.5) N=16 ;CHECKS TIME GROUP
      IF(IIER.EQ.8) N=17 ;CHECKS CLOUD WX GROUP ERROR
      IF(IIER.EQ.10) N=18 ;CHECKS IF ALT GROUP ERROR
      IF(N.EQ.16)CALL WRS(JCH, JMSG17, 25, IER)
      IF(N.EQ.17)CALL WRS(JCH, JMSG18, 28, IER)
      IF(N.EQ.18)CALL WRS(JCH, JMSG19, 23, IER)
      IF(N.GE.16.AND.N.LE.18)CALL WRS(JCH, 6400K, 1, IER)
      IF(N.EQ.16) GO TO 10
      IF (IBUF(6).EQ.3) GO TO 10 ;SKIPS IF A SP
      I HOUR=IBUF(5)/100

```

```

IF (IBUF(5)-Ihour*100.GE.40) Ihour=Ihour+1
IF (Ihour/3*3.NE.Ihour) GO TO 10 ;CHECK IF 3 HRLY
IF (IBUF(50).EQ.32)N=1 ;CHECK PRES TEND ON 3 HRLY
IF (N.EQ.1) CALL WRS(JCH,JMSG1,31,IER)
IF (N.EQ.1) CALL WRS(JCH,6400K,1,IER)
C CHECKS CLOUD CODE GROUP ON 3 HRLY
IF (IBUF(10).EQ.67) GO TO 13
IF (IBUF(13).EQ.87) GO TO 13
IF (IBUF(10).EQ.88.AND.IBUF(14).EQ.32) GO TO 13
DO 15 I=10,18,4
15 IF (IBUF(I).NE.32.AND.IBUF(53).EQ.32)N=2
IF (N.EQ.2) CALL WRS(JCH,JMSG4,42,IER)
IF (N.EQ.2) CALL WRS(JCH,6400K,1,IER)
13 IF (Ihour/6*6.NE.Ihour) GO TO 10 ;CHECK IF 6 HRLY
IF (IBUF(56).EQ.-99) N=3 ;CHECKS MAX MIN INCLUDED
IF (N.EQ.3) CALL WRS(JCH,JMSG2,21,IER)
IF (N.EQ.3) CALL WRS(JCH,6400K,1,IER)
10 CONTINUE
C CHECKS FOG W/DP LESS THAN 8 DEGREES
DO 20 I=24,33
20 IF (IBUF(I).EQ.70.AND.IBUF(35)-IBUF(36).GE.9)N=4
IF (N.EQ.4)CALL WRS(JCH,JMSG3,41,IER)
IF (N.EQ.4) CALL WRS(JCH,6400K,1,IER)
C CHECKS VIS LESS THAN 7 AND NO OBST TO VIS
IF (IBUF(22).GE.7) GO TO 40
IF (N.EQ.0) NN=0
N=5
DO 45 I=24,33
45 IF (IBUF(I).NE.32) N=4
IF (N.EQ.5) CALL WRS(JCH,JMSG5,47,IER)
IF (N.EQ.5) CALL WRS(JCH,6400K,1,IER)
40 CONTINUE
C CHECK BS W/WIND LESS THAN 9 MPH
DO 50 I=24,32
IF (IBUF(I).EQ.66.AND.IBUF(I+1).EQ.83) GO TO 55
50 CONTINUE
GO TO 60
55 IF (IBUF(39).LE.9) N=6
IF (N.EQ.6) CALL WRS(JCH,JMSG10,45,IER)
IF (N.EQ.6) CALL WRS(JCH,6400K,1,IER)
60 CONTINUE
C CHECKS FOR CEILING DESIGNATOR
IIBUF(1)=IBUF(18)
IIBUF(2)=IBUF(14)
IIBUF(3)=IBUF(10)
DO 85 I=1,3
IF (IIBUF(I).EQ.66.OR.IIBUF(I).EQ.79) GO TO 65
IF (IIBUF(I).EQ.88) GO TO 65
85 CONTINUE
GO TO 70
65 II=22-(4*I)
IF (IBUF(II+1).EQ.1.OR.IBUF(II+1).EQ.51) GO TO 75
IF (IBUF(II+3).EQ.77.OR.IBUF(II+3).EQ.69) GO TO 70
IF (IBUF(II+3).EQ.87) GO TO 70
GO TO 72
75 IF (I.EQ.3) GO TO 70
I=I+1
IF (IIBUF(I).EQ.66.OR.IIBUF(I).EQ.79) GO TO 65
IF (IIBUF(I).EQ.88) GO TO 65
GO TO 75

```

```

72   N=7
      CALL WRS(JCH,JMSG6,36,IER)
      CALL WRS(JCH,6400K,1,IER)
70   CONTINUE
C    CHECKS VIS GT 6 MI W/OBST TO VIS
      IF(IBUF(22).LT.7) GO TO 110
      DO 95 I=24,33
      IF(IBUF(I).EQ.70) GO TO 100
      IF(IBUF(I).EQ.72) GO TO 100
      IF(IBUF(I).EQ.75) GO TO 100
      IF(IBUF(I).EQ.68) GO TO 100
95   CONTINUE
      DO 105 I=24,32
      IF(IBUF(I).EQ.66.AND.IBUF(I+1).EQ.83) GO TO 100
      IF(IBUF(I).EQ.66.AND.IBUF(I+1).EQ.79) GO TO 100
      IF(IBUF(I).EQ.66.AND.IBUF(I+1).EQ.68) GO TO 100
      IF(IBUF(I).EQ.73.AND.IBUF(I+1).EQ.70) GO TO 100
      IF(IBUF(I).EQ.66.AND.IBUF(I+1).EQ.89) GO TO 100
      IF(IBUF(I).EQ.71.AND.IBUF(I+1).EQ.70) GO TO 100
105  CONTINUE
      GO TO 110
100  N=8
      CALL WRS(JCH,JMSG9,46,IER)
      CALL WRS(JCH,6400K,1,IER)
110  CONTINUE
C    CHECKS FOR REPORTABLE CLD HGT
S6400K  IF(IBUF(18).EQ.67) GO TO 175
      II=10
180  DO 120 I=II,18,4
      IF(IBUF(I).NE.32) GO TO 140
120  CONTINUE
      GO TO 175
140  II=I+4
      IF(IBUF(I+2).LT.0) GO TO 170
      IF(IBUF(I+2).LE.50) GO TO 175
      IF(IBUF(I+2).LE.100) GO TO 160
      IF(IBUF(I+2)/10*10.NE.IBUF(I+2)) GO TO 170
      IF(II.LT.18) GO TO 180
      GO TO 175
160  IF(IBUF(I+2)/5*5.NE.IBUF(I+2)) GO TO 170
      IF(II.LT.18) GO TO 180
      GO TO 175
170  N=9
      CALL WRS(JCH,JMSG7,27,IER)
      CALL WRS(JCH,6400K,1,IER)
C    CHECKS FOR REPORTABLE VIS
175  CONTINUE
      IF(IBUF(22).LE.0) IBUF(22)=IBUF(22)*10
      IF(IBUF(22).EQ.-620) IBUF(22)=-625
      IF(IBUF(22).LE.0.AND.IBUF(22).GE.-3750) GO TO 190
      IF(IBUF(22).LE.-3750.AND.IBUF(22).GE.-20000) GO TO 200
      IF(IBUF(22).EQ.1.OR.IBUF(22).EQ.2) GO TO 230
      IF(IBUF(22).LT.-20000.AND.IBUF(22).GE.-30000) GO TO 210
      IF(IBUF(22).LT.-30000) GO TO 220
      IF(IBUF(22).GE.3.AND.IBUF(22).LE.15) GO TO 230
      IF(IBUF(22)/5*5.NE.IBUF(22)) GO TO 220
      GO TO 230
190  IF(IBUF(22)/625*625.NE.IBUF(22)) GO TO 220
      GO TO 230
200  IF(IBUF(22)/1250*1250.NE.IBUF(22)) GO TO 220

```



```

GO TO 230
210 IF (IBUF(22)/2500*2500.NE.IBUF(22)) GO TO 220
GO TO 230
220 N=10
CALL WRS(JCH,JMSG11,25,IER)
CALL WRS(JCH,6400K,1,IER)
230 CONTINUE
C CHECKS INTENSITY OF L OR S
DO 240 I=24,33
IF (IBUF(I).EQ.76) GO TO 250
IF (IBUF(I).EQ.83.AND.IBUF(I-1).NE.66) GO TO 250
240 CONTINUE
GO TO 260
250 DO 270 I=24,33
IF (IBUF(I).EQ.76.AND.IBUF(I+1).EQ.45) GO TO 280
IF (IBUF(I).EQ.76.AND.IBUF(I+1).EQ.43) GO TO 290
IF (IBUF(I).EQ.76) GO TO 300
IF (IBUF(I).EQ.83.AND.IBUF(I+1).EQ.45) GO TO 280
IF (IBUF(I).EQ.83.AND.IBUF(I+1).EQ.43) GO TO 290
IF (IBUF(I).EQ.83) GO TO 300
270 CONTINUE
280 IF (IBUF(22).LT.-5000.OR.IBUF(22).GE.1) GO TO 260
GO TO 265
290 IF (IBUF(22).GE.-5000.AND.IBUF(22).LE.-3125) GO TO 260
GO TO 265
300 IF (IBUF(22).GT.-3125.AND.IBUF(22).LE.0) GO TO 260
265 N=11
CALL WRS(JCH,JMSG12,36,IER)
CALL WRS(JCH,6400K,1,IER)
260 CONTINUE
IF (IBUF(41).GT.3130.OR.IBUF(41).LT.2775) N=21 ;CHECKS STATION PRES
IF (N.EQ.21) CALL WRS(JCH,JMSG21,33,IER)
IF (N.EQ.21) CALL WRS(JCH,6400K,1,IER)
IF (IBUF(38).EQ.0.AND.IBUF(39).NE.0) N=25 ;CHECKS WIND
IF (N.EQ.25) CALL WRS(JCH,JMSG25,12,IER)
IF (N.EQ.25) CALL WRS(JCH,6400K,1,IER)
IF (IBUF(38).GT.360.OR.IBUF(38).LT.0) N=26 ;CHECKS WIND
IF (N.EQ.26) CALL WRS(JCH,JMSG25,12,IER)
IF (N.EQ.26) CALL WRS(JCH,6400K,1,IER)
IF (IBUF(10).EQ.88.AND.IBUF(12).GT.40) N=27 ;CHECKS OBSCURATIO ABV 4000FT
IF (N.EQ.27) CALL WRS(JCH,JMSG26,36,IER)
IF (N.EQ.27) CALL WRS(JCH,6400K,1,IER)
IF (IBUF(6).EQ.3) GO TO 350
IF (IBUF(35).GT.125.OR.IBUF(35).LT.-60) N=22 ;CHECKS TEMP
IF (N.EQ.22) CALL WRS(JCH,JMSG22,21,IER)
IF (N.EQ.22) CALL WRS(JCH,6400K,1,IER)
IF (IBUF(36).GT.90.OR.IBUF(36).LT.-60) N=23 ;CHECKS DP
IF (N.EQ.23) CALL WRS(JCH,JMSG23,19,IER)
IF (N.EQ.23) CALL WRS(JCH,6400K,1,IER)
IF (IBUF(36).GT.IBUF(35)) N=24 ;CHECKS DP EXCEEDS TEMP
IF (N.EQ.24) CALL WRS(JCH,JMSG24,17,IER)
IF (N.EQ.24) CALL WRS(JCH,6400K,1,IER)
350 CONTINUE
IF (N.NE.4) NN=N
IF (N.NE.0.AND.NN.EQ.0) KN=1
IF (N.EQ.0) KN=1
RETURN
END

```

```

OVERLAY OV3
C   CHEC1 WILL CHECK FOR SPECIALS
SUBROUTINE CHEC1(IBUF,IBUF1,KN1)
INTEGER IBUF(60),IBUF1(60)
COMMON/CIBLK/JIMSG(12)
COMMON/EBLK/JCH,IOF(5),IEHDR(11)
DATA JIMSG/" THIS SHD BE A SPECIAL"/
C   THIS AREA CHECKS FOR SPECIALS DUE TO TSTMS
N=0
N1=0
DO 20 I=24,33
IF(IBUF(I).EQ.84) N=1
IF(IBUF1(I).EQ.84) N1=1
20  CONTINUE
DO 25 I=24,32
IF(IBUF(I).EQ.84.AND.IBUF(I+1).EQ.43) N=2
IF(IBUF1(I).EQ.84.AND.IBUF1(I+1).EQ.43) N1=2
25  CONTINUE
IF(N.GE.1.AND.N1.EQ.0) GO TO 100
IF(N.EQ.1.AND.N1.EQ.2) GO TO 100
IF(N.EQ.2.AND.N1.EQ.1) GO TO 100
IF(N.EQ.0.AND.N1.GE.1) GO TO 100
C   THIS AREA CHECKS FOR SPECIALS DUE TO HAIL
N=0
N1=0
DO 30 I=24,33
IF(IBUF(I).EQ.65) N=1
IF(IBUF1(I).EQ.65) N1=1
30  CONTINUE
IF(N.EQ.1.AND.N1.EQ.0) GO TO 100
IF(N.EQ.0.AND.N1.EQ.1) GO TO 100
C   THIS AREA CHECKS FOR SPECIALS DUE TO FRZG PRECIP
C   OR IP
N=0
N1=0
N2=0
N3=0
DO 40 I=24,33
IF(IBUF(I).EQ.90)N=1
IF(IBUF1(I).EQ.90) N1=1
40  CONTINUE
IF(N.EQ.1.AND.N1.EQ.0) GO TO 100
IF(N.EQ.0.AND.N1.EQ.1) GO TO 100
DO 45 I=24,32
IF(IBUF(I).EQ.73.AND.IBUF(I+1).EQ.80) N2=1
IF(IBUF1(I).EQ.73.AND.IBUF1(I+1).EQ.80) N3=1
45  CONTINUE
IF(N2.EQ.1.AND.N3.EQ.0) GO TO 100
IF(N2.EQ.0.AND.N3.EQ.1) GO TO 100
IF(N.EQ.0.AND.N1.EQ.0.AND.N2.EQ.0.AND.N3.EQ.0) GO TO 115
C   THIS AREA CHECKS FOR SPECIALS DUE TO CHANGE OF
C   INTENSITY OF FRZG PRECIP
N=0
N1=0
DO 50 I=24,31
IF(IBUF(I).EQ.90.AND.IBUF(I+1).EQ.82.AND.IBUF(I+2).EQ.45)N=1
IF(IBUF(I).EQ.90.AND.IBUF(I+1).EQ.82.AND.IBUF(I+2).NE.45.
1AND.IBUF(I+2).NE.43.AND.IBUF(I+2).NE.87)N=2
IF(IBUF(I).EQ.90.AND.IBUF(I+1).EQ.82.AND.IBUF(I+2).EQ.43)N=3
50  CONTINUE

```

```

IF(N.EQ.3) GO TO 65
DO 55 I=24,31
IF(IBUF(I).EQ.90.AND.IBUF(I+1).EQ.76.AND.IBUF(I+2).EQ.45.AND.N.LT.2)N=1
IF(IBUF(I).EQ.90.AND.IBUF(I+1).EQ.76.AND.IBUF(I+2).NE.45.AND.IBUF(I+2)
1NE.43.AND.N.LT.3) N=2
IF(IBUF(I).EQ.90.AND.IBUF(I+1).EQ.76.AND.IBUF(I+2).EQ.43)N=3
55 CONTINUE
IF(N.EQ.3) GO TO 65
DO 60 I=24,30
IF(IBUF(I).EQ.90.AND.IBUF(I+1).EQ.82.AND.IBUF(I+2).EQ.87.AND.IBUF(I+3)
1EQ.45.AND.N.LT.2) N=1
IF(IBUF(I).EQ.90.AND.IBUF(I+1).EQ.82.AND.IBUF(I+2).EQ.87.AND.IBUF(I+3)
1NE.43.AND.IBUF(I+3).NE.45.AND.N.LT.3) N=2
IF(IBUF(I).EQ.90.AND.IBUF(I+1).EQ.82.AND.IBUF(I+2).EQ.87.AND.IBUF(I+3)
1EQ.43) N=3
60 CONTINUE
65 CONTINUE
DO 70 I=24,31
IF(IBUF1(I).EQ.90.AND.IBUF1(I+1).EQ.82.AND.IBUF1(I+2).EQ.45)N1=1
IF(IBUF1(I).EQ.90.AND.IBUF1(I+1).EQ.82.AND.IBUF1(I+2).NE.45
1AND.IBUF1(I+2).NE.43.AND.IBUF1(I+2).NE.87) N1=2
IF(IBUF1(I).EQ.90.AND.IBUF1(I+1).EQ.82.AND.IBUF1(I+2).EQ.43)N1=3
70 CONTINUE
IF(N1.EQ.3)GO TO 85
DO 75 I=24,31
IF(IBUF1(I).EQ.90.AND.IBUF1(I+1).EQ.76.AND.IBUF1(I+2).EQ.45
1AND.N1.LT.2)N1=1
IF(IBUF1(I).EQ.90.AND.IBUF1(I+1).EQ.76.AND.IBUF1(I+2).NE.45.AND
1IBUF1(I+2).NE.43.AND.N1.LT.3)N1=2
IF(IBUF1(I).EQ.90.AND.IBUF1(I+1).EQ.76.AND.IBUF1(I+2).EQ.43)N1=3
75 CONTINUE
IF(N1.EQ.3) GO TO 85
DO 80 I=24,30
IF(IBUF1(I).EQ.90.AND.IBUF1(I+1).EQ.82.AND.IBUF1(I+2).EQ.87.AND.IBUF1(I+3)
1EQ.45.AND.N1.LT.2)N1=1
IF(IBUF1(I).EQ.90.AND.IBUF1(I+1).EQ.82.AND.IBUF1(I+2).EQ.87.AND.IBUF1(I+3)
1NE.43.AND.IBUF1(I+3).NE.45.AND.N.LT.3) N1=2
IF(IBUF1(I).EQ.90.AND.IBUF1(I+1).EQ.82.AND.IBUF1(I+2).EQ.87.AND.IBUF1(I+3)
1EQ.43)N1=3
80 CONTINUE
85 CONTINUE
IF(N.LT.N1) GO TO 100
IF(N.GT.N1) GO TO 100
THIS AREA CHECKS FOR SPECIALS DUE TO CHANGE OF
INTENSITY OF IP
N=0
N1=0
DO 90 I=24,31
IF(IBUF(I).EQ.73.AND.IBUF(I+1).EQ.80.AND.IBUF(I+2).EQ.45)N=1
IF(IBUF(I).EQ.73.AND.IBUF(I+1).EQ.80.AND.IBUF(I+2).NE.43.AND
1IBUF(I+2).NE.45.AND.IBUF(I+2).NE.87)N=2
IF(IBUF(I).EQ.73.AND.IBUF(I+1).EQ.80.AND.IBUF(I+2).EQ.43)N=3
90 CONTINUE
IF(N.EQ.3) GO TO 97
DO 95 I=24,30
IF(IBUF(I).EQ.73.AND.IBUF(I+1).EQ.80.AND.IBUF(I+2).EQ.87
1AND.IBUF(I+3).EQ.45.AND.N.LT.2)N=1
IF(IBUF(I).EQ.73.AND.IBUF(I+1).EQ.80.AND.IBUF(I+2).EQ.87
1AND.IBUF(I+3).NE.43.AND.IBUF(I+3).NE.45.AND.N.LT.3)N=2
IF(IBUF(I).EQ.73.AND.IBUF(I+1).EQ.80.AND.IBUF(I+2).EQ.87

```

```

1AND. IBUF(I+3).EQ.43)N=3
95   CONTINUE
97   CONTINUE
    DO 105 I=24,31
    IF( IBUF(I).EQ.73.AND. IBUF(I+1).EQ.80.AND. IBUF(I+2).
1EQ.45)N1=1
    IF( IBUF(I).EQ.73.AND. IBUF(I+1).EQ.80.AND. IBUF(I+2).
1NE.43.AND. IBUF(I+2).NE.45.AND. IBUF(I+2).NE.87)N1=2
    IF( IBUF(I).EQ.73.AND. IBUF(I+1).EQ.80.AND. IBUF(I+2).
1EQ.43)N1=3
105  CONTINUE
    IF(N1.EQ.3) GO TO 112
    DO 111 I=24,30
    IF( IBUF(I).EQ.73.AND. IBUF(I+1).EQ.80.AND. IBUF(I+2).EQ.87.
1AND. IBUF(I+3).EQ.45.AND. N1.LT.2)N1=1
    IF( IBUF(I).EQ.73.AND. IBUF(I+1).EQ.80.AND. IBUF(I+2).EQ.87.
1AND. IBUF(I+3).NE.45.AND. IBUF(I+3).NE.43.AND. N1.LT.3)N1=2
    IF( IBUF(I).EQ.73.AND. IBUF(I+1).EQ.80.AND. IBUF(I+2).EQ.87.AND.
1IBUF(I+3).EQ.43)N1=3
111  CONTINUE
112  CONTINUE
    IF(N.LT.N1) GO TO 100
    IF(N.GT.N1) GO TO 100
115  CONTINUE
C    THIS AREA CHECKS FOR SPECIALS DUE TO VSBY
    N1=0
    N=0
    IF( IBUF(22).LT.3)N=1
    IF( IBUF1(22).LT.3)N1=1
    IF(N.EQ.0.AND. N1.EQ.0) GO TO 130
    IF(N.EQ.1.AND. N1.EQ.0) GO TO 100
    IF(N.EQ.0.AND. N1.EQ.1) GO TO 100
    N=0
    N1=0
    IF( IBUF(22).LT.2.AND. IBUF(22).GT.-2000) N=1
    IF( IBUF1(22).LT.2.AND. IBUF1(22).GT.-2000)N1=1
    IF(N.EQ.1.AND. N1.EQ.0) GO TO 100
    IF(N.EQ.0.AND. N1.EQ.1) GO TO 100
    N=0
    N1=0
    IF( IBUF(22).LT.2.AND. IBUF(22).GT.-1500)N=1
    IF( IBUF1(22).LT.2.AND. IBUF1(22).GT.-1500)N1=1
    IF(N.EQ.1.AND. N1.EQ.0) GO TO 100
    IF(N.EQ.0.AND. N1.EQ.1) GO TO 100
    N=0
    N1=0
    IF( IBUF(22).LT.1.AND. IBUF(22).GT.-1000) N=1
    IF( IBUF1(22).LT.1.AND. IBUF1(22).GT.-1000) N1=0
    IF(N.EQ.1.AND. N1.EQ.0) GO TO 100
    IF(N.EQ.0.AND. N1.EQ.1) GO TO 100
130  CONTINUE
C    THIS AREA CHECKS FOR SPECIALS DUE TO CEILINGS
    NY=0
    DO 200 I=10,18,4
    NN=0
    IF( IBUF(I).EQ.66) NN=1
    IF( IBUF(I).EQ.79) NN=1
    IF( IBUF(I).EQ.88) NN=1
    N1=0
    IF( IBUF(I+1).EQ.0.OR. IBUF(I+1).EQ.50) N1=1

```

```

      IF(NN.EQ.1.AND.N1.EQ.1) NY=((I-10)/4+1)
200  CONTINUE
      NY1=0
      DO 210 I=10,18,4
      NN=0
      IF(IBUF1(I).EQ.66) NN=1
      IF(IBUF1(I).EQ.79) NN=1
      IF(IBUF1(I).EQ.88) NN=1
      N1=0
      IF( (IBUF1(I+1).EQ.0.OR.IBUF1(I+1).EQ.50) N1=1
      IF(NN.EQ.1.AND.N1.EQ.1) NY1=((I-10)/4+1)
210  CONTINUE
      IF(NY.EQ.0.AND.NY1.NE.0) GO TO 220
      IF(NY.EQ.0.AND.NY1.EQ.0) GO TO 140
      I=((NY-1)*4+10)
      IF(IBUF(I+2).GE.30) N2=1
      IF(IBUF(I+2).GE.10.AND.IBUF(I+2).LT.30) N2=2
      IF(IBUF(I+2).GE.5.AND.IBUF(I+2).LT.10) N2=3
      IF(IBUF(I+2).LT.5) N2=4
220  IF(NY.EQ.0)N2=1
      IF(NY1.EQ.0) GO TO 230
      II=((NY1-1)*4+10)
      IF(IBUF1(II+2).GE.30) N3=1
      IF(IBUF1(II+2).GE.10.AND.IBUF1(II+2).LT.30) N3=2
      IF(IBUF1(II+2).GE.5.AND.IBUF1(II+2).LT.10) N3=3
      IF(IBUF1(II+2).LT.5) N3=4
230  IF(NY1.EQ.0)N3=1
      N=0
      IF(N2.GT.N3)N=1
      IF(N2.LT.N3) N=1
      IF(N.EQ.1.AND.IBUF(6).EQ.1) GO TO 100
140  CONTINUE
      KN1=0
C     RETURN TO MAIN PROGRAM IF ALL IS OK
      GO TO 110
C     WRITE SPECIAL IS NEEDED
100  CALL WRS(JCH,JIMSG,23,IER)
      CALL WRS(JCH,6400K,1,IER)
      KN1=1
110  CONTINUE
      RETURN
      END

```

```

SUBROUTINE RITVERS(ZTIME, TDATA, IER)
C RITVERS LOOKS THROUGH AVAILABLE VERSIONS OF SAO KEYS LOOKING FOR THE
C VERSION NEEDED. SAO MUST BE IN THE FORM "CLE SA 0557". RITVERS STRIPS
C LEADING BLANKS. "RS AND "SW" TYPE OBSERVATIONS ARE PERMISSIBLE. ERRORS
C RETURN IER = 0 TO MAIN PROGRAM.
C
      DIMENSION LINE(40), NLINE(80) ; NLINE IS UNPACKED LINE
      COMMON/BLKK/KEY1(5)
      INTEGER IDATA(128), TDATA(256)
      INTEGER ZTIME, OBTIME
      IER = 0
C:OPEN PRODUCT "KEY1" FOR READING
      CALL AFREAD(1, KEY1, $450) ; IF ERROR RETURN ZERO TO MAIN
      ITRY=0
C:READ LINE FROM PRODUCT KEY1
      100 CALL AFREAD(2, LINE, $450, $450)
          ITRY=ITRY+1
C:STRIP ANY LEADING BLANKS
      CALL UNPACK(LINE, 80, NLINE)
      DO 125 I=1, 5
          IF (NLINE(I).NE.000040K) GOTO 150 ; IF 1ST NON-BLANK
125 CONTINUE
          GOTO 450 ; RETURN ERROR TO MAIN
150 DO 175 J=1, 75
          NLINE(J)=NLINE(J+I-1)
175 CONTINUE
          DO 180 I=5, 20
          IF(NLINE(I).GE.48.AND.NLINE(I).LE.57) GOTO 182
180 CONTINUE
          GOTO 450
C:CHECK TO SEE IF TIME OF OBSERVATION IS ALL NUMERALS, THEN INTCVT TO OBTIME
      182 NUM1=1 ; NUM1 = 1ST ASCII NUMERAL PASSED STN ID
          NUM2=NUM1 + 1
          NUM4=NUM1 + 3
          DO 185 K = NUM2, NUM4
          IF(NLINE(K).LT.48.OR.NLINE(K).GT.57) GOTO 450
185 CONTINUE
          CALL PACK (NLINE, 80, LINE) ; NEW LINE WITH NO LEADING BLANK
          OBTIME = INTCVT(NUM1, 5) ; SEE AFREAD DOC FOR INTCVT INFO
C:CHECK OBTIME, CALL PREV VERS IF NOT IN WINDOW, GIVE ERR RETURN IF BFR WINDOW
      IF (ITRY.EQ.1) INITTM=OBTIME ; INITTM IS TIME OF LATEST OB
      IF (ITRY.NE.1.AND.OBTIME.EQ.INITTM) GOTO 450; WE'VE SEEN THIS OB BEFORE
      IF (ZTIME.EQ.0000.AND.OBTIME.GE.1800) OBTIME=OBTIME-2360
      IF (OBTIME .LT.(ZTIME-65))GOTO 450 ; VERSION TOO EARLY
      IF (OBTIME .GT.(ZTIME+15))GOTO 200 ; OBSERVATION NOT EARLY ENUF
      IF (LINE(3).EQ.51501K) GOTO 400 ; TYPE OF OBS "SA"
      IF (LINE(3).EQ.51123K) GOTO 400 ; TYPE OF OBS "RS"
      IF (LINE(3).EQ.51527K) GOTO 400 ; TYPE OF OBS "SW"
200 CALL PRVRF(IER)
          IF(IER.NE.1) GOTO 450
          CALL AFREAD(3, IDUM, $450)
          GOTO 100
C:OBSERVATION IS CORRECT VERSION. RETURN LINE TO MAIN PROGRAM
      400 IER = 1
          CALL RDBKF(0, IDATA, IER)
          CALL UNPACK(IDATA, 256, TDATA)
          RETURN
      450 IER = 0
          RETURN
      END

```

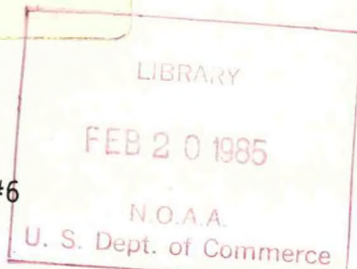
C

```
      CHEC2 CHECKS AGST PREVIOUS HRLY
      SUBROUTINE CHEC2( IBUF, IBUF2, KN3)
      INTEGER IBUF(60), IBUF2(60)
      COMMON/EBLK/JCH, IOF(5), IEHDR(11)
      COMMON/C2BLK/JERR1(11), JERR2(18), JERR3(10), JERR4(14), JERR5(15)
      DATA JERR1/"    TEMP MIGHT BE WRONG"/
      DATA JERR2/"    SEA LEVEL PRESSURE MIGHT BE WRONG"/
      DATA JERR3/"    DP MIGHT BE WRONG"/
      DATA JERR4/"    WIND SPEED MIGHT BE WRONG"/
      DATA JERR5/"    ALT. SETTING MIGHT BE WRONG"/
      N=0
      ISEL=IABS( IBUF(34)-IBUF2(34) ) ;CHECKS SEA LEVEL PRES
      IF( ISEL.GT.30) N=1
      IF( N.EQ.1) CALL WRS( JCH, JERR2, 36, IER)
      IF( N.EQ.1) CALL WRS( JCH, 6400K, 1, IER)
      ITEMP=IABS( IBUF(35)-IBUF2(35) ) ;CHECKS TEMP
      IF( ITEMP.GT.10) N=2
      IF( N.EQ.2) CALL WRS( JCH, JERR1, 22, IER)
      IF( N.EQ.2) CALL WRS( JCH, 6400K, 1, IER)
      IDP=IABS( IBUF(36)-IBUF2(36) ) ;CHECKS DP
      IF( IDP.GT.10) N=3
      IF( N.EQ.3) CALL WRS( JCH, JERR3, 20, IER)
      IF( N.EQ.3) CALL WRS( JCH, 6400K, 1, IER)
      IALT=IABS( IBUF(42)-IBUF2(42) ) ;CHECKS ALT SETTING
      IF( IALT.GT.10) N=4
      IF( N.EQ.4) CALL WRS( JCH, JERR5, 30, IER)
      IF( N.EQ.4) CALL WRS( JCH, 6400K, 1, IER)
      KN3=0
      IF( N.GT.0) KN3=1
      RETURN
      END
```

```

C   SENDSA SENDS CURRENT OBS ALL IF CORRECT
    OVERLAY OV4
    SUBROUTINE SENDSA
    COMMON/BLK/ KEY(5),JMSG(10)
    COMMON/SAODAT/IHDR(7),ID(5),I1HDR(14)
    DIMENSION IUP(80),IOUT(40),IAR(20)
    DATA IHDR/'CRWSAOCRWALL'/
    DATA I1HDR/-1,-1,1000K,142600K,"WOUS00 KCRW DDHMM",6412K/
    DATA ID/'CRWSAOCRW'/
    CALL AFREAD(1,KEY,$500)
    CALL AFREAD(2,IOUT,$550,$500)
    CALL DFILW(ID,IER)
    CALL CRAND(ID,IER)
    CALL GCHN(ICHN,IER)
    CALL OPENE(ICHN,ID,0,IER)
    CALL WRS(ICHN,IHDR,12,IER)
    CALL MMHDR(1,IAR,IER)
    DO 201 I=1,3
201  I1HDR(I+10)=ISHFT(IAR(I*2-1)+60K,8)+IAR(I*2)+60K
    NB=28
    CALL WRS(ICHN,I1HDR,NB,IER)
    GO TO 70
50  CALL AFREAD(2,IOUT,$550,$500)
70  CONTINUE
    CALL UNPACK(IOUT,80,IUP)
    III=1
    DO 20 I=1,76
    III=I+1
    IF(IUP(I).EQ.40K.AND.IUP(I+1).EQ.40K.AND.IUP(I+2).EQ.40K.AND.IUP(I+3).EQ.40K) GO TO 30
20  CONTINUE
    GO TO 500
30  IUP(III-1)=15K
    IUP(III)=12K
    CALL PACK(IUP(1),80,IOUT)
    CALL WRS(ICHN,IOUT,III,IER)
    GO TO 50
500 CALL FORKE("SAOCHEC","INPUT ERROR",IER)
    CALL KLOSE(ICHN,IER)
    GO TO 600
550 CALL WRS(ICHN,203K,2,IER)
    CALL KLOSE(ICHN,IER)
    CALL FSTORE(ID,0,IER)
    CALL WAIT(1,2,IER)
    CALL FORKP("SAOCHEC",ID,IER)
    CALL DFILW(ID,IER)
600 CONTINUE
    RETURN
    END

```

EASTERN REGION CP #6

01/22/85

REAL TIME QUALITY CONTROL OF SAO'SPART A: PROGRAM INFORMATION AND INSTALLATION PROCEDUREPROGRAM NAME: SAOCHEC.SVAAL ID: XXXXXXXXREVISION NO.: 2.00

PURPOSE: This program is designed to provide real-time quality control for surface observations. SAOCHEC works on an SAO in a workbin and returns either a correct SAO under cccSAOxxx or error messages stored in cccWRKERR. This version also has a switch to send/not send the SAO out on the RDC.

PROGRAM INFORMATION

DEVELOPMENT PROGRAMMER(S):

J. BILLET

LOCATION: ERH DATAC

PHONE: FTS 649-5433

LANGUAGE: DG FORTRAN IV/5.20

DATE: 01/28/83

RUNNING TIME: 40 SECONDS

MAINTENANCE PROGRAMMER(S):

J. BILLET

LOCATION: ERH DATAC

PHONE: FTS 649-5433

TYPE: OVERLAY

REVISION DATE: 01/22/85

DISK SPACE: PROGRAM FILES
DATA FILES

156 RDOS BLOCKS
5 RDOS BLOCKS

PROGRAM REQUIREMENTS

PROGRAM FILES:

NAME

SAOCHEC.SV, .OL

COMMENTS

DATA FILES:

NAME

CRWWRKERR

CRWSAOCRW

DP LOCAT.

DP0

DP0

READ/WRITE

WRITE

WRITE

COMMENTS

Error messages stored

Correct SAO stored

AFOS PRODUCTS

ID

cccSAOxxx

cccWRKERR

cccWRKSAO

ACTION

STORE

STORE

READ

COMMENTS

Actual observation

Error messages

Initial draft of SAO

LOAD LINE

RLDR SAOCHEC FINKEY [AIRDX,CHECK,CHEC1,SENDSA] CHEC2 RITVERS

ANDGO ANDEQ ORGO NUMBR AFREAD.LB TOP.LB UTIL.LB FORT.LB

PROGRAM INSTALLATION

- 1- Put SAOCHEC.SV and .OL on DPØF and link from DPØ (or put on DPØ if there is more room there).
- 2- ADD cccWRKERR and cccWRKSAO to the database.

REAL TIME QUALITY CONTROL OF SAO'S

PART B: PROGRAM EXECUTION AND ERROR CONDITIONS

PROGRAM NAME: SAOCHEC.SV

AAL ID: XXXXXXXX

REVISION NO.: 2.00

PROGRAM EXECUTION

1. First, compose the observation in cccWRKSAO.

2. Then run SAOCHEC:

RUN:SAOCHEC/S XXX (or)

RUN:SAOCHEC XXX

If the global /S is used, the observation will be sent out on the RDC if no errors are found. Without the switch the ob will just be stored locally.

3. An alert will flash when SAOCHEC finishes. If the message is that cccSAOxxx has been stored, no errors were found and the ob has either been just stored locally or sent out (depending on whether /S was used). If the message is that cccWRKERR has been stored, errors were found in the ob. Error messages will appear in cccWRKERR.

4. It might be advantageous to set the alarm on cccWRKERR for the console where SAOCHEC will most often be run. This means anytime there are errors, they can be displayed just by hitting the alarm button.

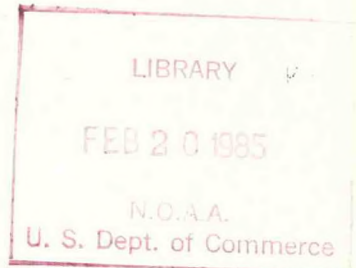
ERROR CONDITIONS

	ADM MESSAGES	MEANING
1.	NO ADM MESSAGES	

	DASHER MESSAGES	MEANING
1.	NONE KNOWN	

EASTERN REGION CP #6

01/22/85



REAL TIME QUALITY CONTROL OF SAO'S

PART A: PROGRAM INFORMATION AND INSTALLATION PROCEDURE

PROGRAM NAME: SAOCHEC.SV

AAL ID: XXXXXXXX

REVISION NO.: 2.00

PURPOSE: This program is designed to provide real-time quality control for surface observations. SAOCHEC works on an SAO in a workbin and returns either a correct SAO under cccSAOxxx or error messages stored in cccWRKERR. This version also has a switch to send/not send the SAO out on the RDC.

PROGRAM INFORMATION

DEVELOPMENT PROGRAMMER(S):

J. BILLET

LOCATION: ERH DATAC

PHONE: FTS 649-5433

LANGUAGE: DG FORTRAN IV/5.20

DATE: 01/28/83

RUNNING TIME: 40 SECONDS

MAINTENANCE PROGRAMMER(S):

J. BILLET

LOCATION: ERH DATAC

PHONE: FTS 649-5433

TYPE: OVERLAY

REVISION DATE: 01/22/85

DISK SPACE: PROGRAM FILES
DATA FILES

156 RDOS BLOCKS
5 RDOS BLOCKS

PROGRAM REQUIREMENTS

PROGRAM FILES:

NAME

SAOCHEC.SV, .OL

COMMENTS

DATA FILES:

NAME

CRWRKERR

CRWSAOCRW

DP LOCAT.

DP0

DP0

READ/WRITE

WRITE

WRITE

COMMENTS

Error messages stored

Correct SAO stored

AFOS PRODUCTS

ID

cccSAOxxx

cccWRKERR

cccWRKSAO

ACTION

STORE

STORE

READ

COMMENTS

Actual observation

Error messages

Initial draft of SAO

LOAD LINE

RLDR SAOCHEC FINKEY [AIRDX,CHECK,CHEC1,SENDSA] CHEC2 RITVERS

ANDGO ANDEQ ORGO NUMBR AFREAD.LB TOP.LB UTIL.LB FORT.LB

PROGRAM INSTALLATION

- 1- Put SAOCHEC.SV and .OL on DPØF and link from DPØ (or put on DPØ if there is more room there).
- 2- ADD cccWRKERR and cccWRKSAO to the database.

REAL TIME QUALITY CONTROL OF SAO'S

PART B: PROGRAM EXECUTION AND ERROR CONDITIONS

PROGRAM NAME: SAOCHEC.SV

AAL ID: XXXXXXXX
REVISION NO.: 2.00

PROGRAM EXECUTION

1. First, compose the observation in cccWRKSAO.
2. Then run SAOCHEC:

RUN:SAOCHEC/S XXX (or)

RUN:SAOCHEC XXX

If the global /S is used, the observation will be sent out on the RDC if no errors are found. Without the switch the ob will just be stored locally.

3. An alert will flash when SAOCHEC finishes. If the message is that cccSAOxxx has been stored, no errors were found and the ob has either been just stored locally or sent out (depending on whether /S was used). If the message is that cccWRKERR has been stored, errors were found in the ob. Error messages will appear in cccWRKERR.
4. It might be advantageous to set the alarm on cccWRKERR for the console where SAOCHEC will most often be run. This means anytime there are errors, they can be displayed just by hitting the alarm button.

ERROR CONDITIONS

	ADM MESSAGES	MEANING
1.	NO ADM MESSAGES	

	DASHER MESSAGES	MEANING
1.	NONE KNOWN	

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**

