AA, NATIONAL WEATHER SERVICE, CENTRAL REGION

COMPUTER PROGRAMS AND PROBLEMS

NWS CRCP - NO. 6

A REGIONAL WEATHER DEPICTION PLOT

Warren E. Sunkel
National Weather Service Forecast Office
Topeka, Kansas

October 1982

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / National Weather Service

NOAA, NATIONAL WEATHER SERVICE
CENTRAL REGION COMPUTER PROGRAMS AND PROBLEMS

NWS CRCP - No. 1	A computer program for determining and displaying the geostrophic winds on Lakes Superior and Michigan. R. Somrek - June 1980
NWS CRCP - No. 2	Program Alembic. W. Sunkel - March 1982
NWS CRCP - No. 3	A subroutine to find and extract data from an AFOS data-base product. R. Somrek - May 1982
NWS CRCP - No. 4	Programs SAVALL and STORALL. W. Sunkel - July 1982
NWS CRCP - No. 5	Program ANALYZ. T. Schwein Sept. 1982

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

NWS CRCP - No. 6

A REGIONAL WEATHER DEPICTION PLOT

Warren E. Sunkel
National Weather Service Forecast Office
Topeka, Kansas

October 1982

UNITED STATES
DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Oceanic and
Atmospheric Administration
John V. Byrne, Administrator

National Weather
Service
Richard E. Hallgren, Director

A REGIONAL WEATHER DEPICTION PLOT

Warren E. Sunkel
National Weather Service Forecast Office
Topeka, Kansas

I. GENERAL INFORMATION

A. Introduction

Program WXPLOT creates an hourly weather depiction chart for display on the GDM of an AFOS console. The station models used on the WXPLOT chart are similar to those of the NMC weather depiction plot NMCGPHPOW but are intended for use on a regional scale with greater data density.

One addition to the NMC station model is a single letter describing the flying weather at the station. If the observed ceiling and visibility meet the criteria for IFR weather, an "I" is plotted to the right of the station circle. Similarly, an "M" is plotted for marginal VFR weather. These characters depict the flying conditions at a glance and can greatly simplify the hand analysis of the chart.

WXPLOT is designed as an adjunct to the job stream for plotting hourly weather observations (SAO's) in conventional station model format on a regional basis (Appendix 1). The valuable information that the ceiling and visibility data provide to the aviation forecaster or briefer more than compensates for the small amount of time WXPLOT adds to the regional plotting job.

B. Program Environment

WXPLOT is intended to run in the background partition of the Data Processing and Control Minicomputer (DPCM) at a forecast office (WSFO). It is written in Data General's FORTRAN IV and requires approximately 12K of 16-bit core memory. Although the program could be made to execute in the limited core space available at a WSO, the disk files required by the entire regional plotting job are too large for a WSO's available disk space.

Disk storage for WXPLOT consists of three files: the program core image WXPLOT.SV (14,336 decimal bytes), the input file WXDATA.bb

(12 bytes per station, typically 2,400 decimal bytes), and the output file DPOF:WXPLOT.SC (about 20 bytes per station, maximum length 6000 decimal bytes). If plots for more than one background are used at a particular station, an additional WXDATA.bb will be created for each background.

Execution time for WXPLOT.SV is approximately 20 seconds. The additional output routines in program PLTGENW increase its execution time a small amount.

Only one I/O channel is required: the same channel is used to read the communications file SAOXXX, read the input data WXDATA.bb, and output the graphics file DPOF:WXPLOT.SC. All channels are fetched with a call to the UTIL.LB subroutine GCHN.

C. References

The regional plotting job stream was developed by Richard Thomas, now with the AFOS Operations Division, National Weather Service Headquarters. This is an extensive system of AFOS background programs to decode surface observations (SAO's), search a large master station directory STDIR.MS for the plotting coordinates of each station, and convert the observed data into NMC plotfile format. The plotfiles may then be plotted in conventional station models with program MODELUGF or in a model of specialized design with program PMOD. The REGPLOT program referenced in this documentation is MODELUGF with modifications by Mr. Thomas.

Program WXPLOT uses the present weather symbol plotting routine from program MODELUGF by Dr. Richard C. Chia of the Systems Development Office, National Weather Service Headquarters. The original subroutine CYMBL was modified slightly into subroutine WSYM.

II. APPLICATION PROGRAM DESCRIPTION

A. Functional Description

WXPLOT is designed to plot weather depiction models (ceiling, visibility, current weather or obstructions to vision) on a regional background (approximately 15 average-size states). Because data are only reformatted into a plotting model, no equations are used by the program. Criteria for determining the flying weather are as follows: an "I" for IFR is plotted if the ceiling is below one thousand feet and/or the visibility is less than three miles, an "M" for Marginal VFR is plotted if the ceiling is at or below 3 thousand feet and at or above one thousand feet and/or the visibility is less than or equal to five miles and greater than or equal to three miles. If the ceiling is above three thousand feet and the visibility is greater than five miles, no letter is plotted to the right of the station circle.

Input to program WXPLOT is from program PLTGENW. PLTGENW is a slightly modified version of PLTGEN, written by Richard Thomas. All of the variables plotted by WXPLOT plus the I and J coordinates of the stations are available in program PLTGEN; but some of the variables are not output, and there is no provision for cloud height in the NMC plotfile format. Therefore, PLTGENW outputs an intermediate file WXDATA.bb (where bb is the map background number) containing all of the information needed by WXPLOT.

WXPLOT requires file SAOXXX to determine the map background number. SAOXXX is created by program SAODEC and its complete format is described in the documentation for program SAODEC. The map background number used by WXPLOT is stored in the sixth word.

Output from program WXPLOT is to file DPOF:WXPLOT.SC. This file is then routed to database product NMCGPHPii, where ii is the map background number plus 50. Subroutine FSTOR from the background library BG.LB (or from the Topeka library TOP.LB) is used to store the output file into the AFOS database.

A console alert is generated to signal program completion.

B. Program Structure

The following meteorological variables are available to WXPLOT through the intermediate file WXDATA.bb: present weather in synoptic code (ww), visibility in SAODEC format (whole miles if positive, thousandths of a mile if negative), total cloud amount, ceiling indicator and ceiling height (lowest cloud layer if ceiling is unlimited, or zero for clear or missing). A single bit is used to indicate a surface temperature less than 36 degrees Fahrenheit. If the present weather symbol is a thunderstorm and this bit is set, a snowflake is plotted instead of a raindrop.

The first step performed by WXPLOT is to reformat the

visibility. If visibility is seven miles or greater, this step is skipped, and no visibility is plotted. Visibility in sixteenths of a mile is converted to a proper fraction.

Sky cover is reformatted next. The station circle is filled as follows:

Sky Cover	SAODEC Code	Station Circle (Oktas)
Clear	C	0
Scattered	S	3
Broken	B	6
Overcast	O	8
Obscured	X	9
Missing	Any other	?

If sky cover is broken or greater, the height of the ceiling (if it exists) is plotted in hundreds of feet below the station circle. If the ceiling is unlimited because the sky cover is thin (-), no value is plotted. For scattered clouds, the height of the lowest scattered layer (in hundreds of feet) is plotted below the station circle.

Flying conditions are computed from ceiling and visibility. The value determined in accordance with the conditions specified in part II A of this documentation is plotted to the right of the station circle.

Visibility and the current weather symbol are plotted to the left of the station circle if both can fit in the space required by two special graphics characters. If the current weather symbol and the visibility require more space, the current weather is plotted in its proper position, and the visibility is centered above the station circle.

Subroutine WSYM determines the special characters required for the current weather symbol and returns all necessary code including the special character flags and line feeds.

Graphics formatting is handled by the following subroutines from TOP.LB: GPREP initializes the graphics array, GOFST (an entry point of GTEXT) plots the offset characters used in the station models, GTEXT plots the date/time group, GFMT adds extra bytes required by Universal Transmission Format and writes the disk output file. One call is made to subroutine GVEC to plot a dummy vector at the end of the graphics file. The word count included with this mode helps the AFOS display software locate the end of the valid data.

All disk input and output is performed by calls to UTIL.LB subroutines. The program also uses several character handling routines from UTIL.LB. The output file DPOF:WXPLOT.SC is stored in the AFOS database with subroutine FSTOR, and the operator is alerted

to program completion with subroutine FORKP. The latter two subroutines reside on BG.LB and have been duplicated on TOP.LB.

Input to WXPLOT is from file WXDATA.bb, where bb is the map background number. The first four words of WXDATA contain the date and time of the data. Data values follow in six-word blocks. There is no special end-of-file marker. The format of file WXDATA is detailed in Appendix 2.

Output from WXPLOT is to disk file DPOF:WXPLOT.SC. This file is in AFOS Universal Transmission Format and is routed to the AFOS database using product key NMCGPHPii, where ii is equal to the map background number plus 50.

WXPLOT.SV is a self-contained executable load module. No overlays are used. The load line is:

```
RLDR WXPLOT WSYM UTIL.LB TOP.LB FORT.LB
```

The load line for program PLTGENW is:

```
RLDR PLTGENW BNSCH INASC WXD UTIL.LB TOP.LB FORT.LB
```

III. PROGRAM TESTING AND IMPLEMENTATION

A. Procedures for Program Implementation

At a WSFO, applications software normally resides on DPOF. Before moving PLTGENW.SV and WXPLOT.SV to DPOF, the original PLTGEN.SV needs to be moved from, deleted or unlinked on both DP0 and DPOF. After moving the new PLTGENW.SV and WXPLOT.SV to DPOF, create the following links on DP0:

```
DIR DP0
LINK PLTGEN.SV DPOF:PLTGENW.SV
LINK WXPLOT.SV DPOF:WXPLOT.SV
```

The SAO decoder program chains to PLTGEN.SV, so this alias needs to exist on the operations disk. Note that it links the execution flow to the new PLTGENW.SV on DPOF. PLTGENW.SV does not chain to any other programs. Other programs in the job stream must be contained in a MACRO file.

WXPLOT.SV is of no use without the other programs in the regional plot series. The following files are required to do a regional plot and weather depiction plot. All files must reside on DP0 or have links on DP0 directing RDOS to the resolution files.

```
SAODEC.SV
SAOD1.(SV,OL)
CCCLIST.nn
PLTGEN.SV linked to PLTGENW.SV
PLTGENW.SV
MBKINDEX
STDIR.MS
REGPLOT.(SV,OL) or MODELUGF.(SV,OL) or
  PMOD.SV and its associated files
WXPLOT.SV
```

The master station directory STDIR.MS is often linked to a newer directory named STDIR.RT. A flow chart detailing all files required and created by the regional plot job is presented in Appendix 1.

Plotting of the regional charts is normally executed through a MACRO file. The usual convention is to name the file PLOTxxx.MC, where xxx is the product designator of the chart to be plotted. To execute the plotting programs, the AFOS operator merely enters the command RUN:PLOTxxx. MACROs can be created with the text editor, the RDOS XFER command, or with the AFOS M:F/ command. The MACRO file for the creation of chart NMCGPHP16 would contain the following CLI commands:

```
SAODEC 16/B 16/C
REGPLOT NMCPLTP16/F NMCGPHP16
WXPLOT
```

B. Sample Run for Testing

These programs operate on real-time input data (SAO's) available in the local station's database. Most hourly data are available about ten minutes after the hour, although waiting until H+30 may add a few more stations. Special observations entered after the hour are used only if the appropriate switch is set in the command line for program SAODEC. The plotting job is executed with a MACRO as described above. Sample output is presented in Appendices 3 and 4.

C. Error codes and Restrictions

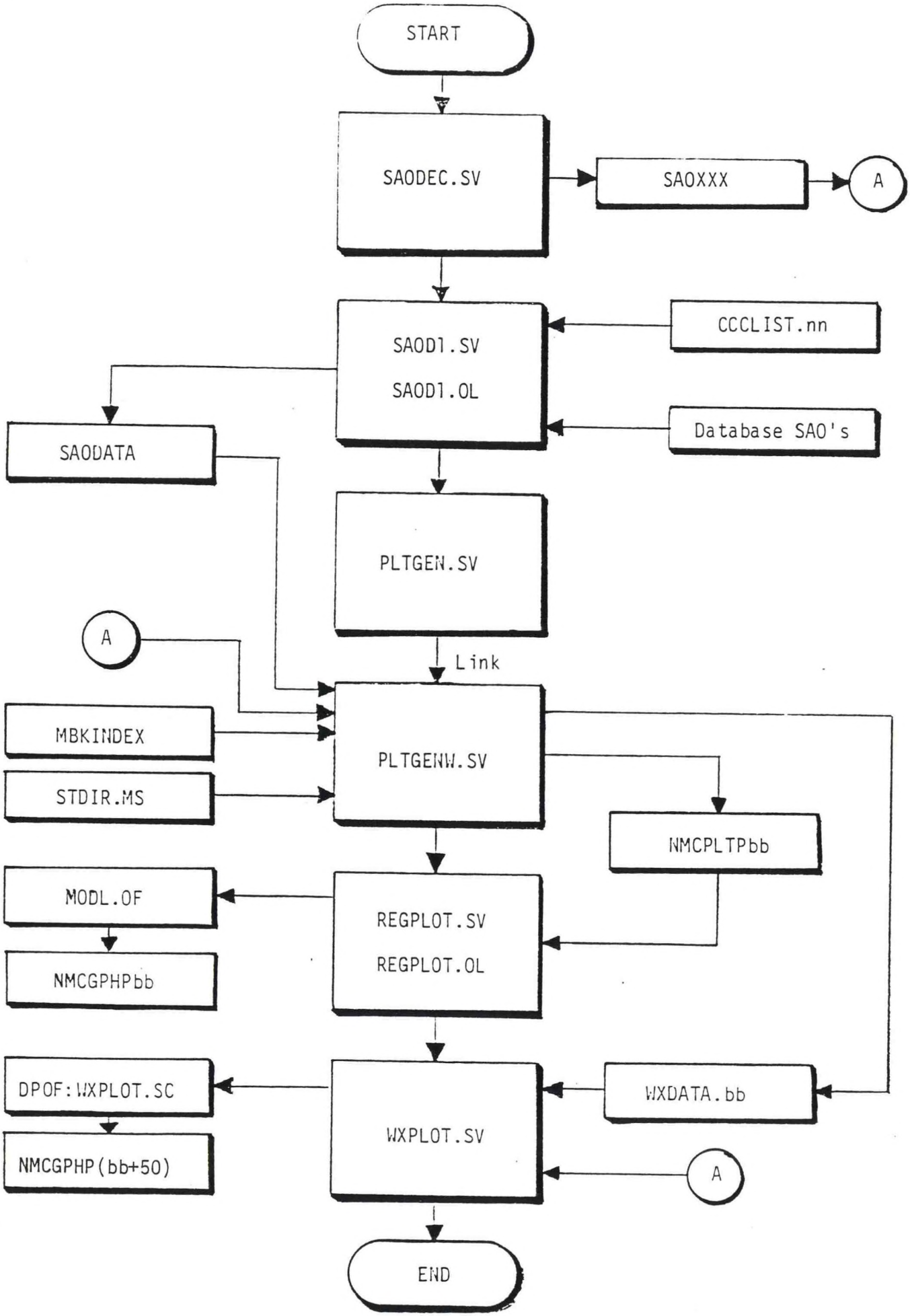
System errors are handled through the UTIL.LB subroutine ERROR. Most of these errors will occur because of missing files when setting up the program. Unfortunately, ERROR's messages are not printed when a program is run from an ADM command. It is good practice, therefore, to initiate the program MACRO from the Printer Keyboard (Dasher) the first few runs.

WXPLOT has built-in overflow protection. If the number of stations exceeds the maximum (about 300), additional reports are bypassed and processing continues. Thus, there are no fatal error conditions created by the input data.

LIST OF APPENDICES

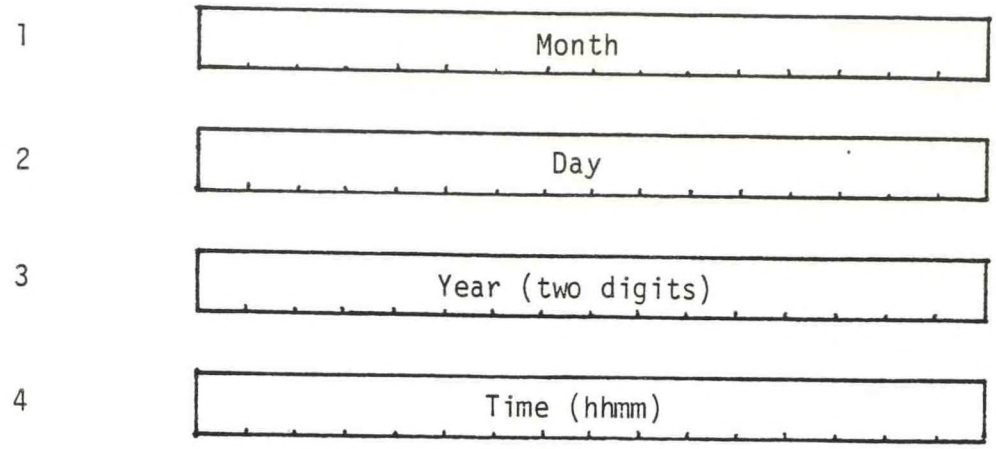
NOTE: Modifications to program PLTGEN are indicated by boxes on the source lising for PLTGENW.

APPENDIX 1

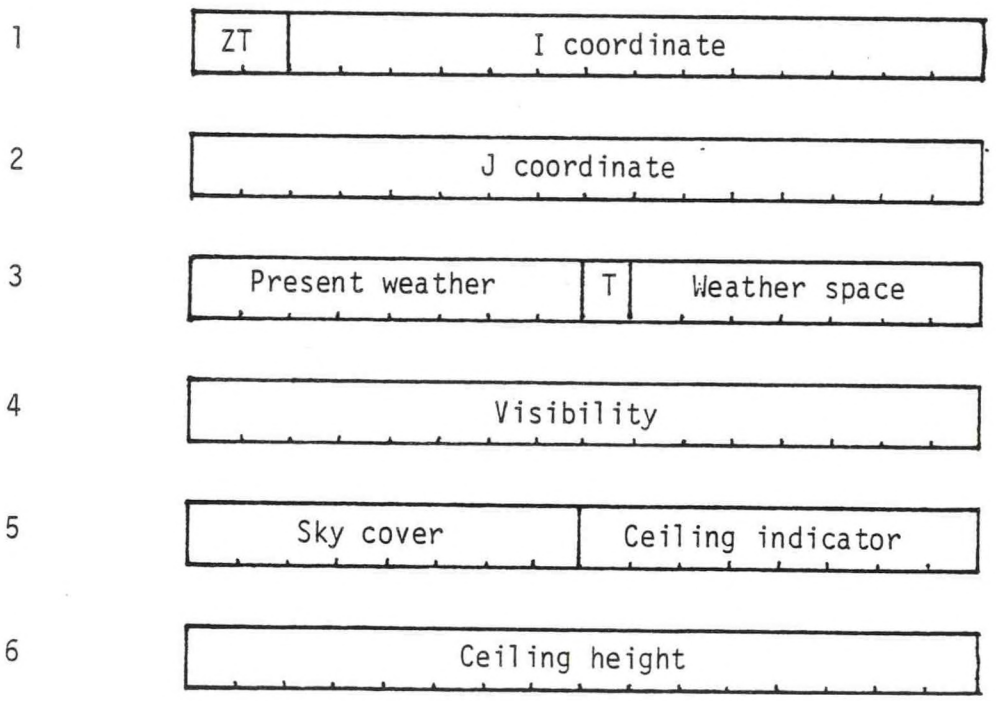

START
SAODEC.SV
SAOXXX
A
SAOD1.SV
SAOD1.OL
CCCLIST.nn
Database SAO's
SAODATA
PLTGEN.SV
A
Link
MBKINDEX
STDIR.MS
PLTGENW.SV
NMCPLTPbb
MODL.OF
NMCGPHPbb
REGPLOT.SV
REGPLOT.OL
DPOF:WXPLOT.SC
NMCGPHP(bb+50)
WXPLOT.SV
WXDATA.bb
A
END

APPENDIX 2

FORMAT OF FILE WXDATA.bb



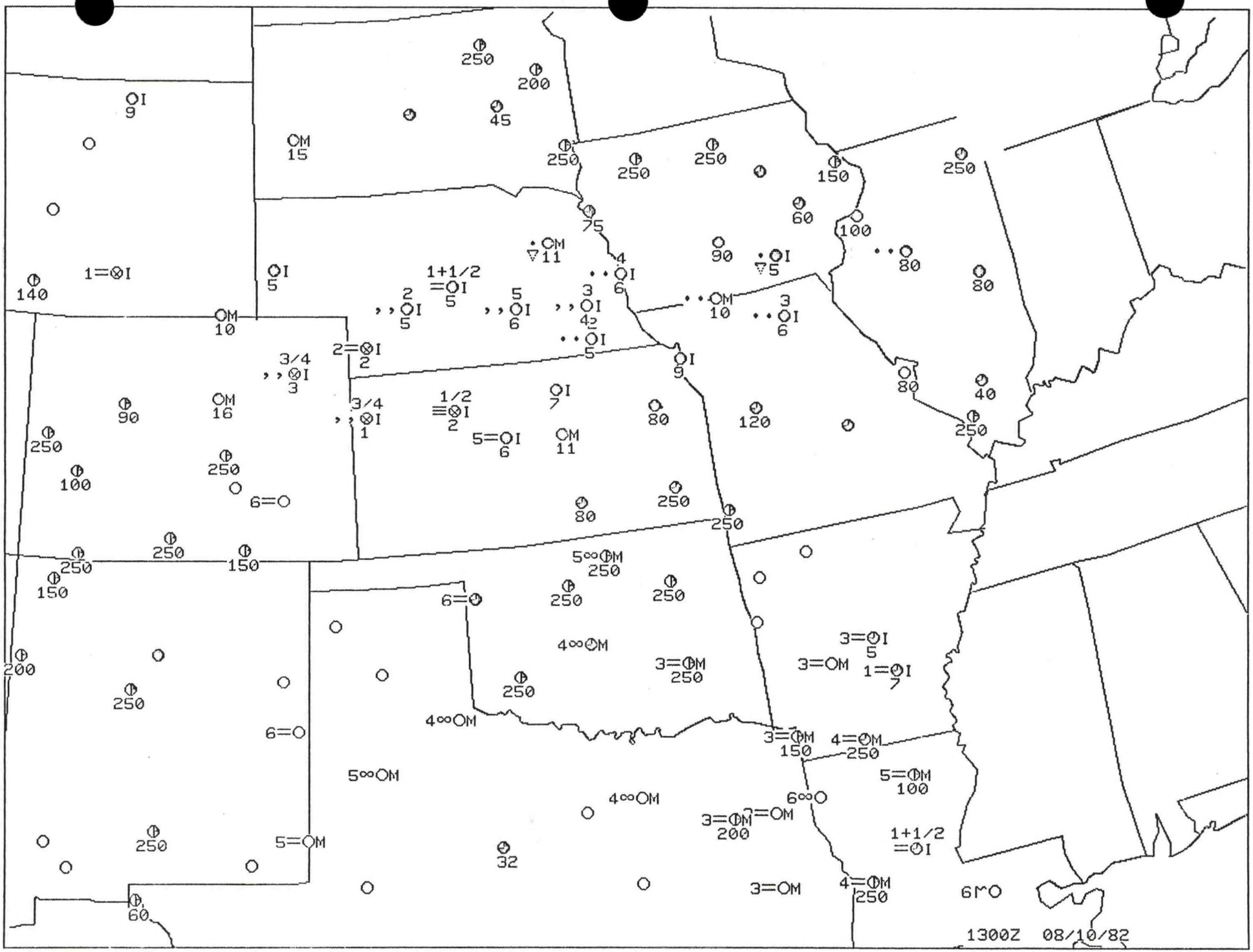
FOUR-WORD HEADER BLOCK



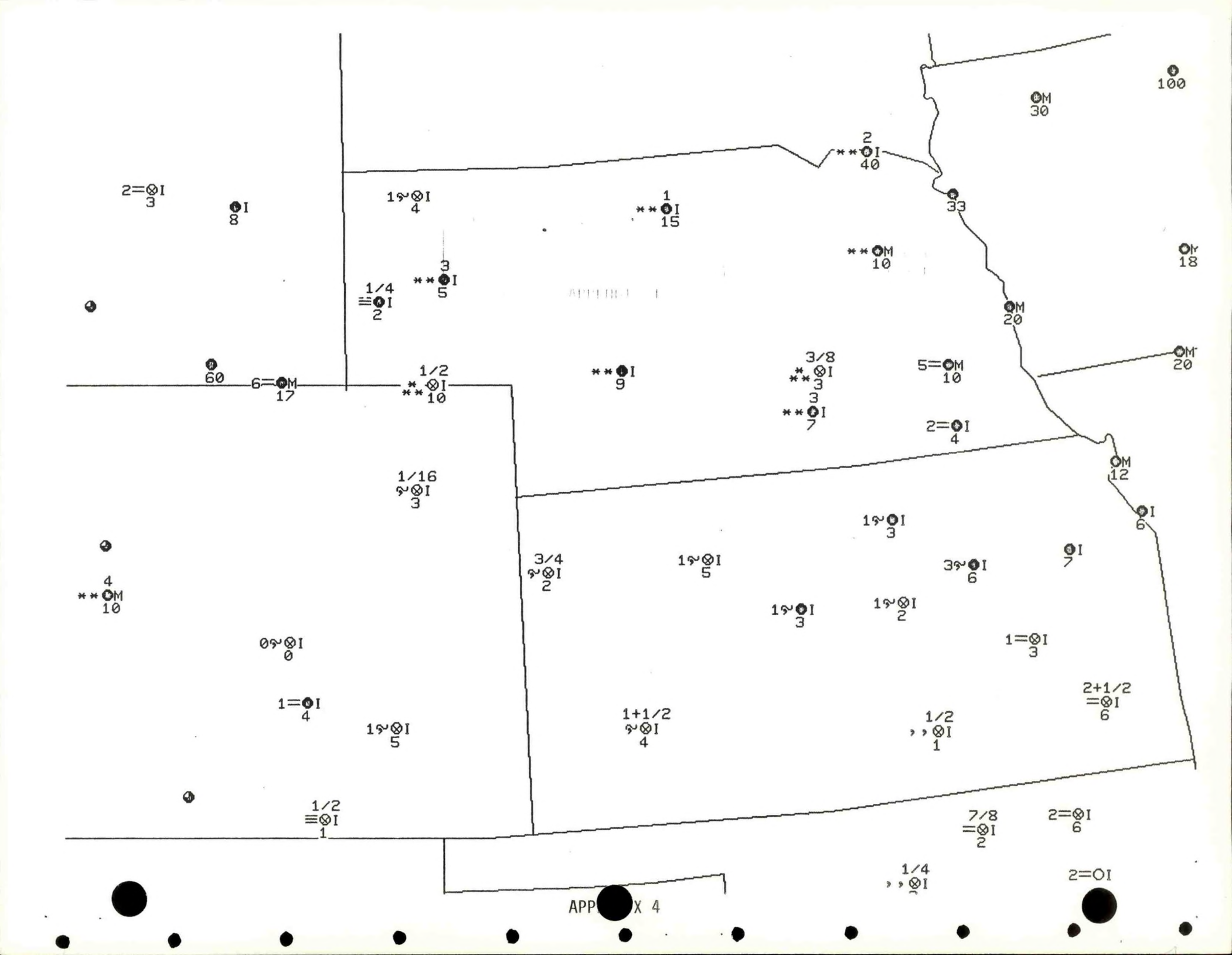
SIX-WORD DATA BLOCK

ZT = zoom threshold T = temperature flag

All values are binary except for sky cover and ceiling indicator, which are ASCII. Word 6 is zero for missing or clear.



APPENDIX 3


100
30
2
40
33
18
20
20
60
6=M
17
1/2
10
9
3/8
3
3
7
5=M
10
2=I
4
12
6
1/16
3
3/4
2
5
3
6
7
4
10
0
1=I
4
1+1/2
4
1/2
1
2+1/2
=I
6
1/2
1
7/8
=I
2
2=I
6
1/4
2=OI
APP X 4

PLTGENW.FR 08/24/82 06:13

```
C*** PLOTFILE GENERATOR  ********
C      MODIFIED VERSION TO OUTPUT WX DEPICTION, WSFO TOPEKA
C***    RICH THOMAS ,NMC
C***      THIS PROGRAM READS DATA DECODED BY SAODEC (IN FILE:  SAODATA )
C*****     AND REFORMATS IT IN PLOTFILE FORM FOR GRAPHIC CODING BY REGPLOT.SV
C***       COMMAND:
C              PLTGEN                    (NO OPTIONS, GETS CONTROLLING PARAMETERS FROM
C                                           WHICH WAS CREATED BY SAODEC.SV)
C

      COMMON /NWEA/NWX,NSYN,ISYN
      DIMENSION NSYN(74),NWX(198)
      DIMENSION IBUF(100),JBUF(90),ID(10),J(6),IC1(20)
      DIMENSION ID1(7),IWID(7)
      DIMENSION IC2(20),IC3(20),IAD(2),ISYN(100)
      INTEGER SF
      DATA ISYN/60K,60K,60K,61K,61K,61K,61K,61K,
     162K,61K,61K,61K,61K,61K,61K,61K,61K,61K,61K,
     262K,62K,62K,62K,62K,64K,64K,64K,62K,62K,
     362K,61K,62K,62K,61K,62K,61K,61K,61K,61K,62K,
     461K,62K,62K,61K,61K,62K,62K,61K,61K,61K,62K,
     561K,64K,63K,64K,61K,61K,61K,63K,61K,62K,61K,
     664K,63K,64K,61K,61K,61K,63K,61K,62K,61K,64K,
     763K,64K,61K,61K,61K,61K,63K,63K,63K,63K,63K,
     863K,63K,63K,63K,63K,63K,62K,62K,62K,62K,63K,
     963K,63K,63K,63K,71K/
      INUM(IA,IB,IC,ID,IE)=((((IA-48)*10+(IB-48))*10+
     1(IC-48))*10+(ID-48))*10+IE-48
      CALL GCHN(JCHN,IER)
      CALL OPENN(JCHN,'SAOXXX',0,IER)
      CALL ERROR(IER,'PLTGEN-SAOXXX NOT FOUND')
      NBYTES=16
      CALL RDS(JCHN,JBUF,NBYTES,IER)
      MBK=JBUF(6)
      CALL KLOSE(JCHN,IER)
      CALL GCHN(ICHN,IER)
      CALL OPENN(ICHN,'MBKINDEX',0,IER)
      CALL ERROR(IER,'MBKINDEX NOT FOUND')
      NBYTES=40
      CALL RDS(ICHN,IBUF,NBYTES,IER)    ;GET PAST COMMS HEADER
      IA=48
      IB=48
      IC=48
      NBYTES=64
    5 CALL RDS(ICHN,JBUF,NBYTES,IER)
      IF(IER.EQ.9)GO TO 900
      CALL ERROR(IER,'PLTGEN-RDS')
      CALL UNPACK(JBUF,NBYTES,IBUF)
      IM=INUM(IA,IB,IC,IBUF(1),IBUF(2))
      IF(IM.EQ.MBK)GO TO 15
      GO TO 5
   15 IF1=1
      JF=1
      DO 10 K=20,51
   10 IF(IBUF(K).EQ.40K)IBUF(K)=60K
```

```
      NF=INUM(IA,IB,IC,IBUF(20),IBUF(21))
      IXP=INUM(IA,IB,IC,IBUF(23),IBUF(24))
      SF=INUM(IA,IB,IC,IBUF(26),IBUF(27))
      IREF=INUM(IBUF(29),IBUF(30),IBUF(31),IBUF(32),IBUF(33))
      JREF=INUM(IBUF(35),IBUF(36),IBUF(37),IBUF(38),IBUF(39))
      DO 11 K=41,45
      IF(IBUF(K).EQ.55K)IF1=-1
      IF(IBUF(K).EQ.55K)IBUF(K)=60K
   11 CONTINUE
       DO 12 K=47,51
      IF(IBUF(K).EQ.55K)JF=-1
       IF(IBUF(K).EQ.55K)IBUF(K)=60K
   12 CONTINUE
      IPOLE=INUM(IBUF(41),IBUF(42),IBUF(43),IBUF(44),IBUF(45))*IF1
      JPOLE=INUM(IBUF(47),IBUF(48),IBUF(49),IBUF(50),IBUF(51))*JF
      IF(IF1.EQ.-1)IBUF(41)=55K
      IF(JF.EQ.-1)IBUF(47)=55K
      CALL KLOSE(ICHN,IER)
      IOFF=IREF/2-IPOLE/4
      JOFF=JREF/2-JPOLE/4
      ID(1)='<0>N'
      ID(2)='<0>M'
      ID(3)='<0>C'
      ID(4)='<0>P'
      ID(5)='<0>L'
      ID(6)='<0>T'
      ID(7)='<0>P'
      ID(8)=IBUF(1)
      ID(9)=IBUF(2)
      ID(10)=56K
      NB=20
      CALL PACK(ID,NB,ID)
      ID(6)=20040K
      ID(7)=20040K
      ID(8)=20040K
      ID(9)=20040K
      ID(10)=20040K
       JBUF(1)=127K; W
       JBUF(2)=130K; X
       JBUF(3)=104K; D
       JBUF(4)=101K; A
       JBUF(5)=124K; T
       JBUF(6)=101K; A
       JBUF(7)=56K; .
       JBUF(8)=IBUF(1)
       JBUF(9)=IBUF(2)
       JBUF(10)=0
       CALL PACK(JBUF,10,IWID)
       CALL DELETE(IWID,IER)
       CALL CRAND(IWID,IER)
       CALL ERROR(IER,"PLTGENW -CRAND")
      CALL DELETE(ID,IER)
      CALL CRAND(ID,IER)
      CALL ERROR(IER,'PLTGEN-CRAND')
      CALL UNPACK(ID,9,IBUF)
      IBUF(10)=60K
      IBUF(11)=60K
      IBUF(12)=60K
      IBUF(13)=377K
      IBUF(14)=377K
```

```
      IBUF(15)=377K
      IBUF(16)=377K
      IBUF(17)=3
      IBUF(18)=0
      IBUF(19)=305K
      IBUF(20)=200K
      CALL GCHN(JCHN,IER)
      CALL ERROR(IER,'PLTGEN-JCHN')
      CALL OPENN(JCHN,'SAODATA',0,IER)
      CALL ERROR(IER,'SAODATA NOT FOUND')
      CALL GCHN(ICHN,IER)
      CALL ERROR(IER,'PLTGEN-ICHN')
      CALL OPENN(ICHN,ID,0,IER)
      CALL ERROR(IER,'PLTGEN-WRS1')
      CALL GCHN(IWCH,IER)
      CALL ERROR(IER,"PLTGEN-WCHAN")
      CALL OPENN(IWCH,IWID,0,IER)
      CALL ERROR(IER,"PLTGEN-WOPN")
      NBYTES=40
      CALL PACK(IBUF,NBYTES,JBUF)
      NBYTES=20
      CALL WRS(ICHN,JBUF,NBYTES,IER)
      CALL ERROR(IER,'PLTGEN-WRITE')
      IBUF(12)=61K
      ISCALE=4000/NF
      CALL INASC(ISCALE,J)
      K=0
      IF(ISCALE.LT.1000)GO TO 6
      IBUF(13)=60K
      IBUF(14)=J(1)
      GO TO 7
    6 IBUF(13)=60K
      IBUF(14)=60K
      K=-1
    7 IBUF(15)=J(2+K)
      IBUF(16)=J(3+K)
      IBUF(17)=J(4+K)
      IBUF(18)=62K
      IBUF(19)=60K
      IBUF(20)=64K
      IBUF(21)=70K
      IBUF(22)=61K
      IBUF(23)=65K
      IBUF(24)=63K
      IBUF(25)=66K
      IBUF(26)=IBUF(54)
      IBUF(27)=IBUF(55)
      IBUF(28)=IBUF(56)
      IBUF(29)=IBUF(57)
      IBUF(30)=IBUF(60)
      IBUF(31)=IBUF(61)
      IBUF(32)=IBUF(62)
      IBUF(33)=IBUF(63)
      IBUF(34)=IBUF(41)
      IBUF(35)=IBUF(42)
      IBUF(36)=IBUF(43)
      IBUF(37)=IBUF(44)
      IBUF(38)=IBUF(45)
      IBUF(39)=IBUF(47)
      IBUF(40)=IBUF(48)
```

```
      IBUF(41)=IBUF(49)
      IBUF(42)=IBUF(50)
      IBUF(43)=IBUF(51)
      NBYTES=8
      CALL RDS(JCHN,JBUF,NBYTES,IER)
      CALL ERROR(IER,'PLTGEN-RDS')
       NBYTES=8
       CALL WRS(IWCH,JBUF,NBYTES,IER)
       CALL ERROR(IER,"PLTGEN-WDTG")
      IT=JBUF(4)/100
      CALL INASC(IT,J)
      IBUF(44)=J(1)
      IBUF(45)=J(2)
      CALL INASC(JBUF(2),J)
      IBUF(46)=J(1)
      IBUF(47)=J(2)
      CALL INASC(JBUF(1),J)
      IBUF(48)=J(1)
      IBUF(49)=J(2)
      CALL INASC(JBUF(3),J)
      IBUF(50)=J(1)
      IBUF(51)=J(2)
      CALL INASC(JBUF(4),J)
       IBUF(52)=IBUF(44)
       IBUF(53)=IBUF(45)
       CALL INASC(MOD(JBUF(4),100),J)
       IBUF(54)=J(1)
       IBUF(55)=J(2)
      DO 9 K=44,55
      IF(IBUF(K).NE.32)GO TO 9
      IBUF(K)=IBUF(K-1)
      IBUF(K-1)=60K
    9 CONTINUE
      IBUF(56)=15K
      IBUF(57)=12K
      IBUF(58)=61K
      IBUF(59)=60K
      IBUF(60)=60K
      IBUF(61)=60K
      IBUF(62)=54K
      IBUF(63)=60K
      IBUF(64)=60K
      IBUF(65)=61K
      IBUF(66)=65K
      IBUF(67)=54K
      IBUF(68)=61K
      IBUF(69)=60K
      IBUF(70)=60K
      IBUF(71)=60K
      IBUF(72)=60K
      IBUF(73)=132K
      IBUF(74)=54K
      IBUF(75)=IBUF(52)
      IBUF(76)=IBUF(53)
      IBUF(77)=IBUF(54)
      IBUF(78)=IBUF(55)
      IBUF(79)=132K
       IBUF(80)=56K;
       IBUF(81)=56K
      IBUF(82)=IBUF(48)
```

```
      IBUF(83)=IBUF(49)
      IBUF(84)=57K
      IBUF(85)=IBUF(46)
      IBUF(86)=IBUF(47)
      IBUF(87)=57K
      IBUF(88)=IBUF(50)
      IBUF(89)=IBUF(51)
      IBUF(90)=73K
      IBUF(91)=15K
      IBUF(92)=12K
      CALL GCHN(IDCH,IER)
      CALL ERROR(IER,'PLTGEN-IDCH')
      CALL OPENN(IDCH,'STDIR.MS',0,IER)
       CALL ERROR(IER,"STATION DIRECTORY NOT FOUND")
      NBYTE=30
      CALL RDS(IDCH,JBUF,NBYTE,IER)
      NREC=JBUF(1)
      LREC=JBUF(2)
      ISTAR=JBUF(3)
      IDP=JBUF(4)
      IDL=(JBUF(5)-JBUF(4))*2
      IZM=JBUF(15)
      ICMA=54K
      ISC=73K
      M=92
      GO TO 78
   20 NBYTES=M
      CALL PACK(IBUF,NBYTES,JBUF)
      CALL WRS(ICHN,JBUF,M,IER)
      CALL ERROR(IER,'PLTGEN-WRITLINE')
   21 NBYTES=96
      CALL RDS(JCHN,JBUF,NBYTES,IER)
      IF(IER.GE.9)GO TO 901
      CALL ERROR(IER,'PLTGEN-READ')
       IF(JBUF(1).EQ.101400K)GO TO 901
      IF(JBUF(3).NE.1.OR.JBUF(3).GT.6)GO TO 21
      J(3)=020040K
      J(2)=JBUF(2)
      J(1)=JBUF(1)
      CALL BNSCH(IDCH,NREC,LREC,ISTAR,IDP,IDL,J,IAD,IC1,IC2,IC3,IC)
      IF(IC.EQ.0)GO TO 21
      GO TO (31,32,33),IC
   31 IX=IC1(IXP)
      JY=IC1(IXP+1)
      IZOOM=IC1(IZM)
      H=IC1(5)
      GO TO 35
   32 IX=IC2(IXP)
      JY=IC2(IXP+1)
      H=IC2(5)
      IZOOM=IC2(IZM)
      GO TO 35
   33 IX=IC3(IXP)
      JY=IC3(IXP+1)
      H=IC3(5)
      IZOOM=IC3(IZM)
   35 IX=((IX-IOFF)*NF)/SF
      JY=((JY-JOFF)*NF)/SF
X     TYPE IX,JY,IOFF,JOFF,NF,SF
      IF(IZOOM.EQ.9)GO TO 21
```

```
      IF(IZOOM.GT.3)GO TO 21
      IF(IX.LT.0.OR.IX.GT.2048)GO TO 21
      IF(JY.LT.0.OR.JY.GT.1536)GO TO 21
      IF(JBUF(3).EQ.9)GO TO 21
      IF(JBUF(3).LE.4.AND.JBUF(3).NE.1)GO TO 21    ;ERROR CHECKS ON SAODATA
      IF(JBUF(3).EQ.7)GO TO 21
       IWID(1)=ISHFT(IZOOM,14)+IX
       IWID(2)=JY
      M=1
      CALL INASC(IX,J)
      DO 40 K=1,4
      IBUF(M)=J(K)
   40 M=M+1
      IBUF(M)=ICMA
      M=M+1
      CALL INASC(JY,J)
      DO 41 K=1,4
      IBUF(M)=J(K)
   41 M=M+1
      IBUF(M)=ICMA
      M=M+1
      NBYTES=10
      CALL UNPACK(JBUF(23),NBYTES,IC1)
      CALL WXD(IC1,NWXTY,JBUF(21),JBUF(9))
      IBUF(M)=IZOOM+48      ;PRIORITY
      IBUF(M+1)=60K         ;SIZE
      IBUF(M+2)=60K         ;ORIENTATION
      IBUF(M+3)=60K         ;WEATHER SPACE
      IF(NWXTY.NE.-99)IBUF(M+3)=ISYN(NWXTY)
       IWID(3)=ISHFT(NWXTY,8)+(IBUF(M+3)-60K)
       IF(JBUF(29).LE.35)CALL ISET(IWID(3),7); SNW FOR TSTMS
       IWID(4)=JBUF(21)
       DO 201 I=1,3
       IWID(5)=ISHFT(JBUF(9),8)+JBUF(24-I*4)
       IWID(6)=JBUF(23-I*4)
       IF(JBUF(24-I*4).NE.40K)GO TO 42
  201  CONTINUE
C      NO CELING, FIND HEIGHT OF LOWEST CLOUD LAYER
       DO 202 I=1,3
       IWID(6)=JBUF(23-I*4)
       IF(IWID(6).NE.0)GO TO 42
  202  CONTINUE
   42  NBYTES=12
       CALL WRS(IWCH,IWID,NBYTES,IER)
       CALL ERROR(IER,"PLTGEN-WX WRS")
      IBUF(M+4)=60K         ;WIND
      IBUF(M+5)=62K         ;OB TYPE
      IBUF(M+6)=ICMA
       GO TO 200; *SKIP MIXING RATIO*
      IF(JBUF(35).EQ.-99)GO TO 200
      IF(JBUF(37).EQ.-99)GO TO 200
      ALT=JBUF(35)/100.
      TEMP=JBUF(36)+273.
      TDPT=JBUF(37)+273.
      PR=(ABS((ALT*33.8639)**.190284-.00008418204*H))**(1./.190284)
      E=6.11*(273./TDPT)**5.31*EXP(25.22*(1.-273./TDPT))
      Q=.622*E/(PR-E)*1000.
      JBUF(31)=Q
  200 M=M+7
      CALL INASC(JBUF(4),J)
```

```
      DO 45 K=1,4
      IBUF(M)=J(K)
   45 M=M+1
      IBUF(M)=ICMA
      M=M+1
      NBYTES=4
      CALL UNPACK(JBUF(1),NBYTES,IBUF(M))
      M=M+4
      IBUF(M)=ICMA
      M=M+1
      IBUF(M)=115K; M FOR MISG
       IF(JBUF(9).EQ.'<0>C')IBUF(M)=60K
      IF(JBUF(9).EQ.'<0>X')IBUF(M)=71K
      IF(JBUF(9).EQ.'<0>O')IBUF(M)=70K
      IF(JBUF(9).EQ.'<0>B')IBUF(M)=66K
      IF(JBUF(9).EQ.'<0>S')IBUF(M)=63K
      IBUF(M+1)=ICMA
      M=M+2
      CALL INASC(JBUF(32),J)          ;WIND DIR
      IF(J(1).EQ.32.AND.NWXTY.EQ.-99)GO TO 21
      IF(JBUF(32).GE.100)GO TO 46
      IF(J(1).EQ.32)GO TO 46
      J(3)=J(2)
      J(2)=J(1)
      J(1)=60K
      IF(JBUF(32).NE.0)GO TO 46
      J(3)=60K
      J(2)=60K
   46 DO 47 K=1,3
      IBUF(M)=J(K)
   47 M=M+1
      CALL INASC(JBUF(33),J)      ;WIND SPEED
      IF(JBUF(33).GE.10)GO TO 48
      IF(J(1).EQ.32)GO TO 48
      J(2)=J(1)
      J(1)=60K
      IF(JBUF(33).EQ.0)J(2)=60K
   48 DO 49 K=1,2
      IBUF(M)=J(K)
   49 M=M+1
      IBUF(M)=ICMA
      M=M+1
       IF(JBUF(28).GE.10000)JBUF(28)=JBUF(28)-1000
      CALL INASC(JBUF(28),J)  ;SLP=28     ALTIMETER=35
      DO 50 K=2,4
      IBUF(M)=J(K)
   50 M=M+1
      IBUF(M)=ICMA
      M=M+1
      CALL INASC(JBUF(29),J)     ;TEMP
      DO 51 K=1,3
      IBUF(M)=J(K)
   51 M=M+1
      IBUF(M)=ICMA
      M=M+1
      CALL INASC(JBUF(30),J)     ;DEW POINT
      DO 52 K=1,3
      IBUF(M)=J(K)
   52 M=M+1
      IBUF(M)=ICMA
```

```
      M=M+1
      IF(NWXTY.NE.100)GO TO 55
      DO 53 K=1,5
      IBUF(M)=IC1(K)
   53 M=M+1
      GO TO 60
   55 CALL INASC(NWXTY,J)      ;WX TYPE       IBUF(M)=J(1)
      IBUF(M)=J(1)
      IBUF(M+1)=J(2)
      M=M+2
   60 IBUF(M)=ICMA
      M=M+1
      IBUF(M)=ICMA
      M=M+1
      CALL INASC(JBUF(42),J)
      IBUF(M)=J(1)
      IBUF(M+1)=J(2)
      M=M+2
      IBUF(M)=ICMA
      M=M+1
      CALL INASC(JBUF(41),J)
      IBUF(M)=J(1)
      IBUF(M+1)=ICMA
      M=M+2
      IBUF(M)=JBUF(44)
      IBUF(M+1)=JBUF(45)
      IBUF(M+2)=JBUF(46)
      M=M+3
      IBUF(M)=ICMA
      M=M+1
       CALL INASC(JBUF(43),J)
      IF(JBUF(43).GT.99)GO TO 65
      IF(JBUF(43).EQ.-99)GO TO 70
      IF(JBUF(43).GE.10)GO TO 64
      IBUF(M)=56K
      IBUF(M+1)=60K
      IBUF(M+2)=J(1)
      M=M+3
      GO TO 70
   64  IBUF(M)=56K
      IBUF(M+1)=J(1)
      IBUF(M+2)=J(2)
      M=M+3
      GO TO 70
   65 IBUF(M)=J(1)
      IBUF(M+1)=56K
      IBUF(M+2)=J(2)
      IBUF(M+3)=J(3)
      M=M+4
   70 GO TO 77
      DO 75 K=0,6
   75 IBUF(M+K)=ICMA
      M=M+7
      CALL INASC(JBUF(31),J)
      DO 76 K=1,3
      IBUF(M)=J(K)
   76 M=M+1
   77 IBUF(M)=ISC
      IBUF(M+1)=15K
      IBUF(M+2)=12K
```

```
      M=M+2
   78 L=1
   79 DO 80 K=L,M
      IF(IBUF(K).EQ.32)GO TO 81
   80 CONTINUE
      GO TO 20
   81 M1=M-1
      DO 82 J1=K,M1
   82 IBUF(J1)=IBUF(J1+1)
      M=M-1
      L=K
      GO TO 79
  900 IER=2
      CALL FORKE('PLTGEN','MBK NOT FOUND',IER)
      CALL ERROR(IER,'MBK NOT FOUND')
      STOP
  901 IBUF(1)='<0>E'
      IBUF(2)='<0>N'
      IBUF(3)='<0>D'
      IBUF(4)=15K
      IBUF(5)=12K
      NBYTES=10
      CALL PACK(IBUF,NBYTES,JBUF)
      NBYTES=5
      CALL WRS(ICHN,JBUF,NBYTES,IER)
      IBUF(1)=101400K
      NBYTES=2
      CALL WRS(ICHN,IBUF,NBYTES,IER)
      CALL KLOSE(JCHN,IER)
      CALL KLOSE(ICHN,IER)
       CALL KLOSE(IDCH,IER)
       CALL KLOSE(IWCH,IER)
      DO 300 K=1,7
  300 ID1(K)=ID(K)
      ID1(6)=0
      ID1(7)=0
C     CALL FORKO('PLTGEN',ID1,IER)
      STOP
      END
```

```
C       PROGRAM WXPLOT
C
C       PLOT WEATHER DEPICTION MODEL ON REGIONAL PLOT BACKGROUND.
C       INPUT IS FROM FILE WXDATA.NN, WHICH IS OUTPUT FROM PROGRAM
C       PLTGENW WITH BACKGROUND NN SPECIFIED IN SAOXXX.  OUTPUT IS
C       ROUTED TO THE AFOS DATA BASE, PRODUCT NMCGPHP(NN+50).
C
C       RLDR WXPLOT WSYM UTIL.LB TOP.LB FORT.LB
C
        PARAMETER NS=3000
        COMMON/WXC/JWD(5),JNMC(5),IOF(10)
        DIMENSION IBUF(20),JBUF(20),IFILE(20),ITITL(16),IAR(NS),JVSB(10)
        DATA JWD/"WXDATA."/JNMC/"NMCGPHP"/IOF/"DP0F:WXPLOT.SC",0/
C       GET INPUT PARAMETERS FROM SAOXXX
        CALL GCHN(ICH,IER)
        CALL OPENR(ICH,"SAOXXX",0,IER)
        CALL ERROR(IER,"WXPLOT-SAOXXX NOT FOUND")
        NB=16
        CALL RDS(ICH,IBUF,NB,IER)
        CALL ERROR(IER,"WXPLOT-SAOXXX ERROR")
        MBK=IBUF(6)
        CALL KLOSE(ICH,IER)
C       CREATE INPUT FILENAME
        CALL UNPACK(JWD,7,IFILE)
        IFILE(8)=MBK/10+60K
        IFILE(9)=MOD(MBK,10)+60K
        IFILE(10)=0
        CALL PACK(IFILE,10,IFILE)
        CALL GCHN(ICH,IER)
        CALL OPENR(ICH,IFILE,0,IER)
        CALL ERROR(IER,"WXPLOT-NO INPUT FILE")
        NB=8
        CALL RDS(ICH,JBUF,NB,IER); DATE/TIME GROUP
        CALL ERROR(IER,"WXPLOT-NO INPUT DATA")
C       REFORMAT DTG
        ITEMP=JBUF(4)
        DO 203 I=1,4
        ITITL(5-I)=MOD(ITEMP,10)+60K
  203   ITEMP=ITEMP/10
        DO 201 I=1,3
        ITITL(I*3+5)=JBUF(I)/10+60K
  201   ITITL(I*3+6)=MOD(JBUF(I),10)+60K
        ITITL(5)=132K; Z
        ITITL(6)=40K
        ITITL(7)=40K
        ITITL(10)=57K; /
        ITITL(13)=57K
        ITITL(16)=0
        CALL PACK(ITITL,16,ITITL)
C       C PRODUCT KEY
        CALL UNPACK(JNMC,7,IFILE)
        MBK=MBK+50
        IFILE(8)=MBK/10+60K
        IFILE(9)=MOD(MBK,10)+60K
        IFILE(10)=0
```

```
      CALL PACK(IFILE,10,IFILE)
      CALL GPREP(IAR,IFILE,NW,NS,IER)
      CALL GTEXT(1,IAR,3164,30,ITITL,NW,NS,1,IER)
C     START READ LOOP
   10 NB=12
      CALL RDS(ICH,IBUF,NB,IER)
      IF(IER.NE.1)GO TO 90; END OF DATA
      IF(IBUF(4).EQ.-99)GO TO 10; SKIP IF NO VSBY
C
C     REFORMAT VISIBILITY
C
      IVSB=IBUF(4)
      NVSB=0
      IF(IVSB.GE.7)GO TO 40
      JVSB(1)=IVSB+60K
      NVSB=1
      IF(IVSB.GE.0)GO TO 40
      VSB=-IVSB/1000.
      MILES=VSB
      FRAC=VSB-MILES
      IN=FRAC*16.+1/32.; NUMERATOR
      DO 202 I=1,4
      ID=2**(5-I); DENOMINATOR
      IF(MOD(IN,2).NE.0)GO TO 20
      IN=IN/2
  202 CONTINUE
      IN=0
   20 N=1
      IF(MILES.EQ.0)GO TO 22
      JVSB(N)=MILES+60K
      IF(IN.EQ.0)GO TO 40
      N=N+1
      JVSB(N)=53K; +
      N=N+1
   22 JVSB(N)=IN/10+60K
      IF(JVSB(N).NE.60K)N=N+1
      JVSB(N)=MOD(IN,10)+60K
      N=N+1
      JVSB(N)=57K; /
      N=N+1
      JVSB(N)=ID/10+60K
      IF(JVSB(N).NE.60K)N=N+1
      JVSB(N)=MOD(ID,10)+60K
      NVSB=N
C
C     COMPUTE SKY COVER
C
   40 ITEMP=ISHFT(IBUF(5),-8)
      ISKY=5; MISG
      IF(ITEMP.EQ.103K)ISKY=16K; C
      IF(ITEMP.EQ.123K)ISKY=23K; S
      IF(ITEMP.EQ.102K)ISKY=1;   B
      IF(ITEMP.EQ.117K)ISKY=3;   O
      IF(ITEMP.EQ.130K)ISKY=4;   X
C
C     COMPUTE HEIGHT OF CEILING
C
      ITEMP=IAND(IBUF(5),377K)
      ICIG=-1; UNLIMITED
      IF(ITEMP.NE.40K)ICIG=IBUF(6); HUNDREDS
```

```
C
C       DETERMINE FLYING CONDITIONS
C
        IFLY=111K; I
        IF(ICIG.GE.0.AND.ICIG.LT.10.OR.IVSB.LT.3)GO TO 50
        IFLY=115K; M
        IF(ICIG.GE.0.AND.ICIG.LE.30.OR.IVSB.LE.5)GO TO 50
        IFLY=126K; V
   50   IF(ISKY.EQ.5)IFLY=77K; ? FOR MISSING
C
C       COMPUTE CLOUD HEIGHT
C
        IHGT=-99; MISSING
        IF(ISKY.EQ.23K)IHGT=IBUF(6); LOWEST LAYER FOR SCT
        IF(ICIG.GE.0)IHGT=IBUF(6); CIG LAYER IF CEILING
C
C       PREPARE PLOTTING ROUTINE
C
        JWX=IAND(IBUF(3),177K); WX SPACE
        IF(JWX.GT.4)JWX=0
        IX=IAND(IBUF(1),37777K)*2
        JY=IBUF(2)*2
        IZT=ISHFT(IBUF(1),-14)+1
C
C       PLOT VISIBILITY ABOVE MODEL
C
        IF(NVSB.EQ.0)GO TO 120
        IF(NVSB.LE.1.AND.JWX.NE.2.AND.JWX.NE.4)GO TO 120
        ID=-(7*NVSB+2*(NVSB-1))/2
        JD=8
        JVSB(NVSB+1)=0
        CALL PACK(JVSB,NVSB+1,JVSB)
        CALL GOFST(1,IAR,IX,JY,ID,JD,JVSB,NW,NS-20,IZT,IER)
C
C       PLOT SKY, FLYING CONDITIONS, AND HEIGHT OF CLOUDS
C
  120   N=1
        JBUF(N)=22K; SET S.C. FLAG
        JBUF(N+1)=ISKY
        JBUF(N+2)=21K; RESET S.C. FLAG
        N=N+3
        IF(IFLY.EQ.126K)GO TO 122; V
        JBUF(N)=IFLY
        N=N+1
  122   IF(IHGT.LT.0)GO TO 128
        JBUF(N)=15K; NEW LINE
        N=N+1
        I=1
        CALL UBNDEC(IHGT,ITITL,I)
        IF(IHGT.LT.100)GO TO 124
        IF(JWX.GT.2)GO TO 123
        JBUF(N)=10K; BKSP
        N=N+1
  123   JBUF(N)=ITITL(4)
        N=N+1
  124   IF(IHGT.LT.10)GO TO 126
        JBUF(N)=ITITL(5)
        N=N+1
  126   JBUF(N)=ITITL(6)
        N=N+1
```

```
  128  JBUF(N)=0
       CALL PACK(JBUF,N,JBUF)
       CALL GOFST(1,IAR,IX,JY,-5,-5,JBUF,NW,NS-20,IZT,IER)
C
C      PLOT VISIBILITY AND WEATHER
C
       IF(NVSB.GT.1.OR.JWX.EQ.2.OR.JWX.EQ.4)NVSB=0; VSB>6 OR ALREADY PLOTTED
       IF(NVSB.EQ.0.AND.JWX.EQ.0)GO TO 10; PLOTTING COMPLETE
       ID=-29
       IF(IAND(JWX,1).EQ.1)ID=-18; 1 OR 3
       IF(JWX.EQ.0)ID=-14
       JD=-5
       N=1
       IF(NVSB.EQ.0)GO TO 132
       IF(JWX.EQ.0)GO TO 136
       JBUF(N)=10K; BKSP
       N=N+1
  136  JBUF(N)=JVSB(1)
       N=N+1
  132  IFLAG=.FALSE.; FOR MDT OR HVY PCPN
       IBLK=1
       IF(JWX.EQ.0.OR.IBUF(3).LT.0)GO TO 134
       NWX=ISHFT(IBUF(3),-8); WX CODE
       IF(NWX.LT.50.OR.NWX.GT.79)GO TO 140
       NWX=MOD(NWX,10)
       IF(NWX.NE.3.AND.NWX.NE.5)GO TO 140
       IFLAG=.TRUE.
       IBUF(3)=IAND(IBUF(3),177600K)+2
       JD=-11
       GO TO 140
  142  N=1
       IBUF(3)=IAND(IBUF(3),177400K)+1
       IF(NWX.EQ.5)IBUF(3)=IBUF(3)+2
       ID=-24
       JD=-5
       IFLAG=.FALSE.
       IBLK=0
  140  CALL WSYM(IBUF(3),JBUF,N)
  134  JBUF(N)=0
       CALL PACK(JBUF,N,JBUF)
       CALL GOFST(IBLK,IAR,IX,JY,ID,JD,JBUF,NW,NS-20,IZT,IER)
       IF(IFLAG)GO TO 142
       GO TO 10; FETCH NEXT REPORT
C
C      BUTTONUP
C
   90  CALL KLOSE(ICH,IER)
       CALL GVEC(IAR,0,0,4,4,NW,NS,4,IER)
       CALL GFMT(IOF,IAR,NW,NS,IER)
       CALL ERROR(IER,"WXPLOT-OUTPUT ROUTINE")
       CALL FSTOR(IOF,0,IER)
       CALL FORKP("WXPLOT",IFILE,IER)
       IF(IER.EQ.1)GO TO 92
       CALL SPCHR("JOB COMPLETE, OUTPUT IN FILE <0>",IER)
       CALL UNPACK(IOF,20,JBUF)
       CALL SPCHR(JBUF,IER)
       CALL SPCHR(15K,IER)
       CALL TYPED(NW)
       CALL SPCHR(" WORDS USED<15>",IER)
   92  CALL RESET
       call EXIT
       END
```

```
      SUBROUTINE WSYM(IWX,JWRD,L)
C     BORROWED FROM MODELUGF (SUBROUTINE CYMBL)
      DIMENSION JWRD(1)
      JW=IAND(IWX,177K)
      IW=ISHFT(IWX,-8)
     I93=0
     I95=0
      IF(ITEST(IWX,7))I93=1
      IF(ITEST(IWX,7))I95=1
     JWRD(L)=22K
     L=L+1
     LFLAG=1
     IW0=IW/10+1
     IW1=IW-IW/10*10+1
     IF (IW.GE.30) GOTO 19
     JWRD(L)=IW+30K
     IF (IW.LT.20) GOTO 500
     JWRD(L+1)=61K
     IF (IW.LT.25) GOTO 500
     IF (IW.EQ.28) GOTO 145
     IF (IW.EQ.29) GOTO 191
     JWRD(L)=IW+24K
     JWRD(L+1)=62K
     IF (IW.EQ.27) JWRD(L)=64K
     GOTO 500
  19 JWRD(L+1)=71K
     GOTO (500,500,500,30,40,50,60,70,80,90),IW0
  30 GOTO (131,131,135,134,134,135,136,136,136,136),IW1
 131 JWRD(L)=67K
     GOTO 500
 134 JWRD(L)=72K
     GOTO 500
 135 JWRD(L+1)=IW+27K
  35 JWRD(L)=70K
     GOTO 500
 136 JWRD(L)=IW+27K
     GOTO 500
  40 GOTO (140,141,144,145,144,145,146,147,148,148),IW1
 140 JWRD(L)=110K
     JWRD(L+1)=123K
     GOTO 500
 141 JWRD(L)=77K
     GOTO 500
 144 JWRD(L)=100K
     GOTO 500
 145 JWRD(L)=65K
     GOTO 500
 146 JWRD(L+1)=100K
     GOTO 35
 147 JWRD(L+1)=65K
     GOTO 35
 148 JWRD(L)=IW+21K
     GOTO 500
  50 GOTO (150,151,152,151,150,151,156,156,152,150),IW1
 151 JWRD(L+1)=IW0+46K
```

```
150   JWRD(L)=IW0+46K
      GOTO 500
152   JWRD(L)=IW+17K
      GOTO 500
156   JWRD(L)=IW+14K
      GOTO 500
 60   GOTO (150,151,162,151,150,151,166,166,168,169),IW1
162   JWRD(L)=117K
      GOTO 500
166   JWRD(L)=IW+4
      GOTO 500
168   JWRD(L)=57K
      GOTO 500
169   JWRD(L)=56K
      GOTO 500
 70   GOTO (150,151,148,151,150,151,176,176,176,178),IW1
176   JWRD(L)=IW-2
      GOTO 500
 80   GOTO (180,180,162,168,168,169,169,187,187,190),IW1
180   JWRD(L)=55K
      GOTO 500
187   JWRD(L)=64K
      GOTO 500
 90   GOTO (190,192,192,193,194,195,187,195,198,187),IW1
190   JWRD(L)=120K
      GOTO 500
194   I93=I93+2
193   JWRD(L+1)=124K+I93
191   JWRD(L)=51K
      GOTO 500
192   JWRD(L+1)=IW-12K
      GOTO 191
195   JWRD(L)=55K+I95
      GOTO 500
198   JWRD(L)=67K
500   L=L+1
      IF (JW/2*2.EQ.JW) L=L+1
      IF (JW.LE.2) RETURN
      JWRD(L)=13
      JWRD(L+1)=66K
      GOTO (800,800,620,800,800,650,660,670,680,690),IW0
620   JWRD(L+2)=63K
      L=L+3
      GOTO 800
650   IF (IW.EQ.54) JWRD(L+1)=103K
      IF (IW.EQ.59) JWRD(L+1)=111K
660   IF (IW.EQ.69) JWRD(L+1)=57K
670   IF (IW.EQ.64 .OR. IW.EQ.74) JWRD(L+1)=IW+17K
      IF (IW1.EQ.6) JWRD(L+1)=IW0+46K
      GOTO 648
680   GOTO (648,681,648,648,681,648,681,648,681,648),IW1
681   JWRD(L+1)=116K
      GOTO 648
690   JWRD(L+1)=51K
      GOTO (681,800,800,800,800,648,648,697,648,697),IW1
697   JWRD(L+1)=130K
648   L=L+2
800   RETURN
      END
```

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

NOAA, 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 monitor and predict the state of the solid Earth, the oceans and their living resources, the atmosphere, and the space environment of the Earth, and to assess the socioeconomic impact of natural and technological changes in the environment.

The six Major Line 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.

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.

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

TECHNICAL SERVICE PUBLICATIONS—These are publications containing data, observations, instructions, etc. A partial listing: Data serials; Prediction and outlook periodicals; Technical manuals, training papers, planning reports, and information serials; and Miscellaneous technical publications.

ATLAS—Analysed data generally presented 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.

Information on availability of NOAA publications can be obtained from:

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

3300 Whitehaven Street, N.W.
Washington, D.C. 20235