NOAA Western Region Computer Programs and
Problems NWS WRCP - No. 41

A PILOT BRIEFING PROGRAM FOR THE BACKGROUND PARTITION

Salt Lake City, Utah
March 1983

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

PREFACE

This Western Region publication series is considered as a subset of our Technical Memorandum series. This series will be devoted exclusively to the exchange of information on and documentation of computer programs and related subjects. This series was initiated because it did not seem appropriate to publish computer program papers as Technical Memoranda; yet, we wanted to share this type of information with all Western Region forecasters in a systematic way. Another reason was our concern that in the developing AFOS-era there will be unnecessary and wasteful duplication of effort in writing computer programs in National Weather Service (NWS). Documentation and exchange of ideas and programs envisioned in this series hopefully will reduce such duplication. We also believe that by publishing the programming work of our forecasters, we will stimulate others to use these programs or develop their own programs to take advantage of the computing capabilities AFOS makes available.

We solicit computer-oriented papers and computer programs from forecasters for us to publish in this series. Simple and short programs should not be prejudged as unsuitable.

The great potential of the AFOS-era is strongly related to local computer facilities permitting meteorologists to practice in a more scientific environment. It is our hope that this new series will help in developing this potential into reality.

NOAA WESTERN REGION COMPUTER PROGRAMS AND PROBLEMS NWS WRCP

1 Standardized Format for Computer Series.
2 AFOS Crop and Soil Information Report Programs. Ken Mielke, July 1979.
3 Decoder for Significant Level Transmissions of Raobs. John A. Jannuzzi, August 1979.
4 Precipitable Water Estimate. Elizabeth Morse, October 1979.
5 Utah Recreational Temperature Program. Kenneth M. Labas, November 1979.
6 Normal Maximum/Minimum Temperature Program for Montana, Kenneth Mielke, December 1979.
7 Plotting of Ocean Wave Energy Spectral Data. John R. Zimmerman, December 1979.
8 Raob Plot and Analysis Routines. John A. Jannuzzi, January 1980.
9 The SWAB Program. Morris S. Webb, Jr., April 1980. (PB80-196041)
10 Flash-Flood Procedure. Donald P. Laurine and Ralph C. Hatch, April 1980. (PB80-298658)
11 Program to Forecast Probability of Summer Stratus in Seattle Using the Durst Objective Method. John R. Zimmerman, May 1980.
12 Probability of Sequences of Wet and Dry Days. Hazen H. Bedke, June 1980. (PB80-223340)
13 Automated Montana Hourly Weather Roundup. Joe L. Johnston, July 1980. (PB81-102576)
14 Lightning Activity Levels. Mark A. Mollner, July 1980. (PB81-108300)
15 Two Fortran Applications of Wind-Driven Ekman Water Transport Theory: Upwelling Index and Storm Tide. Kent S. Short, July 1980. (PB81-102568)
16 AFOS System Local Data Base Save and Rebuild Procedures or A Master Doomsday Program. Brian W. Finke, July 1980. (PB81-108342)
17 AFOS/RDOS Translator Subroutine. Morris S. Webb, Jr., August 1980. (PB81-108334)
18 AFOS Graphics Creation from Fortran. Alexander E. MacDonald, August 1980. (PB81-205304)
19 DATAKEYØ Repair Program. Paul D. Tolleson, August 1980. (PB81-102543)
20 Contiguous File Transfer from the DPCM to the DCM. Paul D. Tolleson, September 1980. (PB81-128035)
21 Freezing Level Program. Kenneth B. Mielke, September 1980. (PB81-128043)
22 Radar Boresighting Verification Program. Thomas E. Adler, November 1980.
23 Accessing the AFOS Data Base. Matthew Peroutka, January 1981.
24 AFOS Work Processor. Morris S. Webb, Jr., February 1981. (PB81-210007)
25 Automated Weather Log for Terminal Forecasting. John A. Jannuzzi, February 1981. (PB81-210999)
26 Program to Computer Downwind Concentrations from a Toxic Spill. John R. Zimmerman, February 1981. (PB81-205296)
27 Animation of AFOS Graphics. Joe Wakefield and Jim Fors, April 1981.
28 AFOS Interactive Graphics. Jim Fors, Don Laurine, and Sandy MacDonald, April 1981.
29 Computer Programs for Aviation Forecast Transmission. Kenneth B. Mielke and Matthew R. Peroutka, May 1981.
30 AFOS Product Collective Program. Morris S. Webb, Jr., September 1981.
31 Graphic Display of FOUS Output. Stephen D. Steenrod, September 1981.
32 Automation of Hourly Aviation Observation Calculations. W. Paul Duval, October 1981.
33 Mesoscale Objective Analysis. Andrew J. Spry and Jeffrey L. Anderson, December 1981.
34 Orographic Snowfall Rate Model for Alta, Utah. Steven K. Todd and Glenn E. Rasch, December 1981.
35 F-6 Monthly Climatic Summary Program for AFOS. Peter G. Mueller, May 1982.

NOAA Western Region Computer Programs and Problems NWS WRCP - No. 41

A PILOT BRIEFING PROGRAM FOR THE BACKGROUND PARTITION

Kenneth B. Mielke
Joe L. Johnston
WSFO Great Falls, Montana

UNITED STATES
DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Oceanic and
Atmospheric Administration
John V. Byrne, Administrator

National Weather
Service
Richard E. Hallgren, Director

This publication has been

approved for publication by

Scientific Services Division,

Western Region.

Glenn E. Rasch

Glenn E. Rasch, Acting Chief
Scientific Services Division
Western Region Headquarters
Salt Lake City, Utah

CONTENTS

LIST OF FIGURES

A PILOT BRIEFING PROGRAM FOR THE BACKGROUND PARTITION

Kenneth B. Mielke, Lead Forecaster
Joe L. Johnston, Weather Service Specialist
WSFO Great Falls, Montana

I. GENERAL INFORMATION

A. Summary:

After the arrival of AFOS, the tools and aids available for pilot briefing have diminished. This is not due to our forecast products or observations, but due to the difficulty in efficiently gathering and presenting the numerous products necessary for a thorough pilot briefing via AFOS. Although the AFOS AIRS program was to fulfill this need, it has not yet materialized into a viable program. Thus, a pilot briefing program (PB.SV) was written by experienced pilot briefing personnel for pilot briefers.

Realistically, many pilot briefings do not cover all aspects of the flight in detail. This program will serve to encourage the briefer to consider all data pertaining to the flight and will greatly reduce the chance of missing an important observation or advisory. Additionally, the output is a data base product. It can be printed if a hard copy is desired or recalled without rerunning the program.

B. Environment:

This program (PB.SV) and the Route Plot Program (PWBRT.SV) run on the Eclipse S/230 Data General computer with AFOS running in the other ground. Both programs are written in Data General Fortran IV using subroutines in libraries BG.LB, UTIL.LB, AFOSE.LB, and AG.LB.

II. REFERENCES

1 NOAA Western Region Computer Problems and Programs WRCP No. 18, "AFOS Graphics Creation from Fortran", MacDonald, A.E., August 1980.

III. APPLICATION

PB.SV and PWBRT.SV can be run separately from an ADM. However, as an option to the user, PB.SV can chain to PWBRT.SV.

A. PB.SV

1. Complete Description

The program begins with a call to subroutine PARENTNODE which finds an existing SKEL file and opens it to read the parent node of the AFOS site running the program. This parent node is used for the CCC in identifiers for pilot report collectives, sigmets, airmets, and area forecast products. If no SKEL file is found, an error flag is set for later use.

A call to DTG returns an ASCII date and time, DDZZZZ, for the WOUS header of the data base product so there will be an official time of start of briefing logged. This DDZZZZ is converted to integer data for date/time check of products in subroutine PWBWT.

Next, a call to PBSTN is made which reads the ADM input line from "ICE2.CM". The input line consists of station identifiers (XXX's) for takeoff and landing points and up to three intermediate points; area forecast region numbers along the flight path; and a Y if the program is to chain to the route plot program. Exact format of the run line is illustrated in Section IV. PBSTN returns the XXX's to the main program in array IOUT and also produces output file PWBXXX if a chain to the route-plot program is desired.

Next, input files "STATIONS" and "STATIONID" are opened for reading only, and an output disk file "PBFILE" is created and opened. If files cannot be found or opened, or if input data is not given correctly, error messages are returned to the calling ADM and the program is terminated.

At this point a loop is started through the XXX's in array IOUT. For each station the following occurs:

A call to Subroutine GETNODE is made. This routine does a binary search in "STATIONID" for the node (CCC) of the XXX given. If a node cannot be found, the program will continue substituting CCC for node so a missing message can be written for the operator. A check is made to determine if this XXX is another within the same state boundary. In this way, pilot reports are displayed only prior to the next XXX within a new state boundary.

A call to GETSTA is made at this point which does a search through file "STATIONS" returning the two-letter state ID of the CCC. This ID is needed for the XX of the PRC category.

An inner loop begins at this point through the now completed keys (CCCNNNXXX's) which are for TWBSYN, SAO, FTA, and PRC data. A TWEB synopsis is written only when first entering a new state boundary and PRC's only before leaving that state. A check is made for the current day and for the valid time of each product. A window of 12 hours is allowed for TWBSYN, 6 for SAO's, 12 for FTA's, and 4 for PRC's.

Four observations are written to the data base product followed by one FTA for each XXX. Because of the official nature of a pilot briefing, the output will contain either the data for CCCNNNXXX or the key along with the message "missing in DATAKEYØ" or "OLD DATA IN".

When the last XXX is processed, the program enters another loop for the hazardous weather section. One to three region numbers should have been entered on the input line. For each region number given, the following are written to output file "PBFILE":

Sigmets A-N, Airmets O-Z are scanned for an existing product. NOTE: Airmets and Sigmets are not tested for date/time, instead PB.SV assumes these data are in the data base on a time purge basis. There are no "missing" or "old" messages as a result. However, the next product called

for is the FAH (Flight Precautions) for that same region number. The FAH is handled in a normal fashion with checks for date/time.

All products are accessed through standard BG.LB routines. The block of data is unpacked and handed to subroutine PWBWT which strips unwanted beginnings and endings. Then the routine filters the data for spurious bytes. Spurious bytes were found in PRC products which caused hangs of the ADM console. PWBWT then writes the products to the output file and returns to the main program for another block of data.

If the parent node error flag had been set, the call to FSTORE is bypassed and output file "PBFILE" is not deleted. Numerous alerts are returned to indicate the status of the program. A CHAIN to PWBRT.SV is made as necessary.

2. Machine Requirements

PB.SV uses about 21,000 bytes (approximately 40 blocks) of disk space. It must be on or linked to DPØ as must all input and output files. The program requires 12K of memory to run.

B. PWBRT.SV

1. Complete Program Description

The program first determines if it is running alone or whether it was chained to/from the PB.SV program. This is done by checking the status of the PWBXXX file. Once the station ID's have been read, the AFOS master directory file (STDIR.MS) is opened for reading only. Using a binary search method, a search is made for the selected station identifiers in the STDIR.MS file. As each one is found, the elevation of the station and its X-Y PIXEL coordinates (for map background #2) are read into an array. When the elevation and Pixel coordinates have been found for each station, the program moves to the actual plotting of the graphic.

The graphic created consists of a plot of the station route on the North American (BØ2) background. Thus, the route can be overlayed with other products such as a radar chart, weather depiction chart or the winds aloft. The graphic also includes the station elevation and the mileage between XXX's along the route of flight.

The product is then stored into the AFOS data base under the header NMCGPHPWB. Add this product to your data base and assign map background #2. If the program fails to find a station ID in the STDIR.MS file, the program is aborted and an alert identifies the errant station.

2. Machine Requirements

This program resides on 37 blocks of disk space and also will run in 12K of core. It should either be in DPØ or be linked from DPØ.

C. Data Base

The AFOS SKEL, STATIONID, and STATIONS files are read but not modified by the program. Two data base products, CCCPWBLOG and NMCGPHPWB, are created and stored locally only.

Output file PBFILE is created for FSTORE, then deleted from the disk. PWBXXX is created if a chain is called for to PWBRT.SV which then deletes the file. PWBRT.SV opens and searches file STDIR.MS for PIXEL coordinates and elevation data for each XXX input.

IV. PROCEDURES

1. The Pilot Weather Briefing Program (PB.SV) is initiated by the following command at an ADM.

RUN:PB XXX XXX XXX XXX XXX/###Y

XXX = Any station ID in the United States or Canada that is in the STATIONID file. However, many Canadian stations are not in the "STATIONID" file. The XXX portion must contain at least two stations and not more than five.

/ = A separator which must follow the list of ID's.

= Any of the new region numbers used in the FA, sigmet, and airmet format, i.e., 1=Northwest, 2=Southeast, 3=North Central, 4=South Central, 5=Rockies, 6=West Coast. Zero, 7, 8, and 9 are not valid numbers. One to three #'s may be used.

Y = "Yes", which will chain to the graphic portion of the PB program. Omit the "Y" for a No (no chain).

Errors will be returned to the user ADM. Also returned is a "JOB PB COMPLETE PRODUCT PWBLOG STORED" after a successful run.

2. The graphic portion of the Pilot Weather Briefer can be run as a Stand Alone program by entering the following at an ADM:

RUN:PWBRT XXX XXX XXX XXX XXX

where XXX is defined as for PB.SV above.

Errors will be returned to the user ADM. A "JOB COMPLETE" is also returned after a successful run.

3. Example ... If you wanted to present a briefing for a flight from MOT to RAP to CYS to GJT to SLC, you would enter:

RUN:PB MOT RAP CYS GJT SLC/35Y

where 3 = North Central Region
5 = Rocky Mountain Region
Y = Yes for chain to the graphic route plot

4. Output - PB.SV produces alphanumeric output stored under CCCPWBLOG. PWBRT.SV produces graphic output under NMCGPHPWB (map background #2). (CCC = your parent node). Sample output is shown in Figures 1 and 2.

In both the PB.SV and PWBRT.SV, commas, spaces, slashes, etc. can be used as separators between the XXX's. However, the last XXX must be followed by a "/".

5. Local preparation for using these programs.

Two keys must be added to the data base.

1. CCCPWBLOG - where CCC is the parent node of your station.
2. NMCGPHPWB with a map background #2.

The programs assume a normal AFOS configuration (SKEL, STATIONID, STATIONS, and STDIR.MS files). The PB.SV and PWBRT.SV programs require no local alterations.

V. SCENARIO

Here is a scenario of how a typical briefing could take place using these programs:

A pilot calls or arrives in person and asks about flying weather from Great Falls, Montana, through the Harlowton area, then over to Billings and Sheridan, Wyoming, to Rapid City, South Dakota.

While he is giving these stations, the pilot briefer should enter the appropriate command, RUN:PB GTF 3HT BIL SHR RAP/53Y (Enter). Over the next 60 to 75 seconds, while the program is running, the general synoptic situation could be discussed along with inquiries about altitude, VFR/IFR, Aircraft, etc., as appropriate.

As the first alert light returns, call up the PWBLOG product. The information is in the standard briefing form. One synopsis is given for each state and not repeated for other stations within the same state boundary. Before leaving the state the current list of pilot reports, if any, are displayed. This and other data sections will either give you a report, or "CCCNNNXXX is not in DATAKEYØ" or "Old Data in CCCNNNXXX", as necessary. Then a separator will indicate you are moving into another state area followed by the synopsis for that state. This continues until the last station is processed. The graphic portion can be displayed on an adjacent GDM and overlayed with any field you wish (winds aloft, weather depiction, radar, 700-mb Hgt Field, etc.). The graphic portion serves as an excellent "corridor" briefing, supplementing the data already on the ADM.

The ADM output will finish by scanning every category of Sigmets and Airmets in the region numbers (#) you entered. This section will also give the hazards portion of the FA for each of the regions selected. As a reminder, "check CCCFA#(I,T,W)" is displayed for the briefer so he will know how to obtain additional data.

VI. CAUTIONS AND RESTRICTIONS

1. The programs assume that Sigmets and Airmets will be in the data base on a time purge basis (at GTF, we use 5 hours for Sigmets, 7 hours for Airmets). If you use a version purge, the program may be picking up old Sigmets and Airmets.

Appropriate missing or old product messages are returned to the output file for all products except Sigmets and Airmets. All data except Sigmets and Airmets are checked for Date/Time against the AFOS clock. Only the four latest Pireps (within the last four hours) will be written to the output file.

2. If a two point briefing covers several states (i.e. SEA to DEN), the briefer should include points in the intermediate states so that Pireps in these intermediate states will be included in the output. For example, the SEA-DEN briefing should be run as:

RUN:PB SEA BOI SLC DEN/56Y

to include Idaho and Utah Pireps.

3. It is the responsibility of the operators to use the correct region numbers in order that the appropriate Airmets/Sigmets/Flight Hazards are included in the output file.

4. When displaying NMCGPHPWB on the GDM, the intensity level of overlay one may need to be adjusted downward so that the mileage, which is in reverse block video, will not be blurred.

5. We have found PIXEL coordinate errors in some versions of the STDIR.MS for a couple of stations while testing the program (MKC and OMA). The coordinates given in our file put the locations of these stations out in the Carribean. Also, there may be other PIXEL errors that we did not find. In any event, these errors should be corrected, not only for this program but for others as well.

Figure 1. Sample PWBLOG.

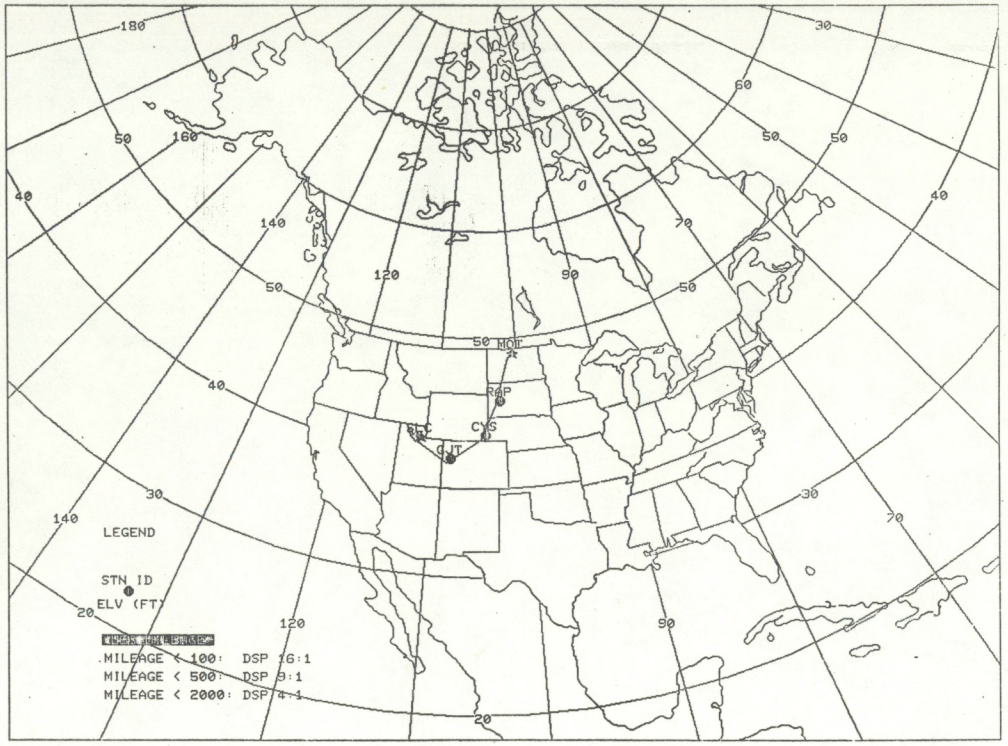
```
WOUS00 KGTF 151901
BIS SYNS 151806.  CD HI PRES OVR ERN DKTS WIL DRFT E AS WRMFNT DVLPS
 OVR MT.
  MAY

 MOT SA 1751 CLR 10 221/16/16/1809/008/SCUD W-NW-N HRZN 010 12
 MOT RS 1657 -X3 SCT 2F 225/17/17/2106/009/F8 VSBY LWR NW

 MOT SP 1631 -X 1F 2106/009/F8

MOT FT AMD 2 151615 1615Z  C2 X 1/2F VRBL C5 BKN 2F.
  18Z 40 SCT 1910. 00Z 60 SCT SCT OCNLY BKN. 09Z VFR..

          C U R R E N T   ND   P I R E P S   F O L L O W


 ND 151844
DIK UA /OV ISN 1806 FL DURGC /TP C206 /SK OVC 030

OLD DATA IN GTFPRCND

                    ********* SD  DATA *********

FSD SYNS 151806 HI PRES OVR ERN DAKS AND NEB THIS AFTN WILL MOV EWD.
SRLY SFC FLO WILL INCRS FROM THE W OVR THE NRN PLAINS.
WILLIAMS

RAP SA 1850 CLR 35 188/34/23/1307/999/ FEW CS
RAP SA 1750 250 -SCT 35 199/30/23/1303/002/ 105 1008 14
RAP SA 1650 200 SCT 250 SCT 35 200/28/21/0000/002
RAP SA 1550 250 -SCT 35 201/23/18/3104/001/ PATCHY FOG SE-S

RAP 151515 CLR. 17Z 100 SCT 250 -BKN. 00Z 80 SCT C250 BKN.
 09Z VFR..

          C U R R E N T   SD   P I R E P S   F O L L O W


 SD 151624
ATY UA /OV DWN360010 1531 FL 130 /TP BE90 /TA -12C /RM OVC BEGINS 80 N
    DWN CLR TO NW

OLD DATA IN GTFPRCSD

                    ********* WY  DATA *********

CYS SYNS 151907 HGTS ASL UNLESS OTRW NOTED. FLTND UPR LVL RDG OVR
 WY AS UPR LVL DSTRBNC MOVS RDG TDA. ANLYS CONTG TO INDC STG WRLY FLO
WI ABNDNT PACFC MSTR CAUSG LO CIG/VSBY OVR WRN WY WI NRMS SHWRS
OMTNS. ER RTES MSTLY MID/HI LVL MSTR AND CONTG STG GSTY SFC WNDS.
MCRANDAL

CYS SA 1854 E110 BKN 200 OVC 60 196/36/17/2617/002/ ACSL ALQDS CU SW
CYS SA 1752 120 SCT E200 OVC 60 207/36/16/2719G29/004/ ACSL ALQDS
SC SW/ 103 1547 19
CYS SA 1651 E120 BKN 180 OVC 60 211/34/15/2715/004/ ACSL ALQDS
CYS SA 1552 E120 BKN 180 OVC 60 205/31/14/2514/003/ ACSL ALQDS FEW SC

CYS 151616  80 SCT C150 BKN 2715G25 LWR SCT V BKN. 10Z VFR WND..

          C U R R E N T   WY   P I R E P S   F O L L O W
```

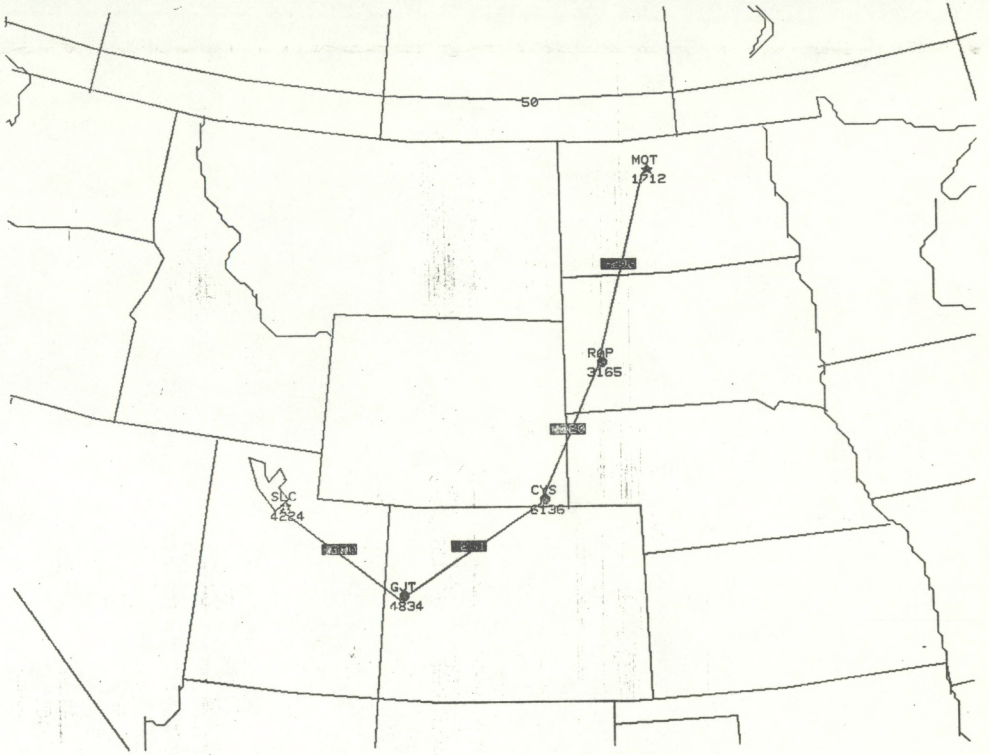
Figure 1. Sample PWBLOG. continued

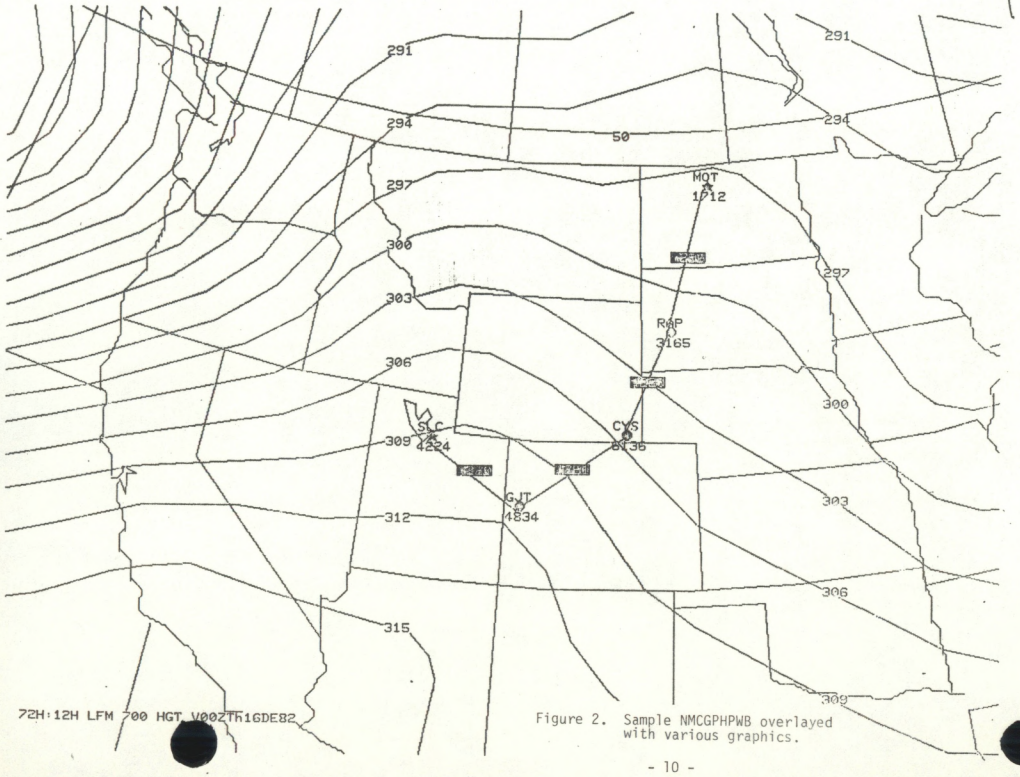
```
 WY 151824
LAR UUA /OV LAR273030 1816 FL130 /TP C310 /TB SVR /RM ON V4

LAR UA /LOV CPR-LAR 1646 FL105 /TP C177 /TB MDT-SVR /RM SHIRLEY BASIN
    CLDS MVG IN
RKS UA /OV RKS030065 1515 FL 120 /TP PA62 /TB MDT
RWL UA /OV RKS030065 1510 FL120 /TP PA62 /TB MDT
RWL
WRL UA /OV LAR-RIW 1517 FL 140 /TPC210 /TB LGT-MDT /IC LGT RIME
     /RM ON OFF INST 10

CPR UA /OV CPR160036 1453FL180/TP G159/SK 200 BKN/TA -25
SHR UA /OV CZI360030 1412 FL095/TP C402/TB LGT MDT


                    ******* CO  DATA *******

    DENTWBSYN   IS NOT IN DATAKEY0

GJT SA 1850 E75 BKN 120 OVC 50 302/34/23/1210/035
GJT SA 1748 E45 BKN 120 BKN 250 OVC 50 301/34/20/1307/036 214 1577
   22
GJT SA 1648 45 SCT E120 BKN 250 OVC 50 295/31/18/1308/034/BINOVC
GJT SA 1548 E45 OVC 30 294/29/19/1011/033 SWU-NE

GJT 151616 40 SCT C80 BKN 1310 OCNL C40 BKN. 19Z 20 SCT C40 BKN
 1515 CHC C20 BKN 3RW-SW-. 10Z MVFR CIG SW..


          C U R R E N T   CO   P I R E P S   F O L L O W


 CO 151824
EGE UA /OV GWS 1745
    FL VFR /TP C172 /SK OBSCD /RM UNABLE VFR EGE-GJT UNABLE VFR RIL-CAG

                    ******* UT  DATA *******

SLC SYNS 151907 AN UPR AIR SHRT WV EMBDD IN MOIST WLY FLOW ALF WILL
PASS OVER THE AREA FM 00Z ONWARD.. WITH LWST FLYING CONDS OVR NRN RTES.
FRZLVL 7-10 THSD FT.

SLC SA 1851 50 SCT E70 OVC 40 253/37/30/1412/025/ BINOVC DSNT S
SLC SA 1851 50 SCT E70 OVC 40 253/37/30/1412/025/ BINOVC DSNT S
SLC SA 1752 M55 BKN 95 OVC 25 265/36/29/1613/029/ 30703 157/ 90206
90405 32
SLC SA 1651 M55 BKN 95 OVC 20 259/34/28/1512/026/ SE25 LOW CLDS S

SLC 151616 20 SCT C50 OVC 1810 OCNL C5 X 1/2S-F. 18Z 30 SCT C50 BKN
 1810 CHC C10 BKN 3RW-SW-. 10Z VFR..


          C U R R E N T   UT   P I R E P S   F O L L O W


 UT 151844
CDC UA /OV U07-BDG 1816 FL 115 /TP C210 /SK 120 SCT-BKN N /RM HIR MTNS E
     APPEAR MVFR
SLC ATA UA/OV SLC 1759 F330/TA MS55/WV UNA/TB SMTH


          ********** HAZARDOUS WEATHER SECTION **********
         SIGMET ALFA 1 VALID UNTIL 151930
IA
FM FSD TO LSE TO BRL TO OMA TO FSD
MDT OCNL SVR RIME ICGICIP SFC TO 70 RPRTD BY SVRL ACFT. CONDS
IPVG AFT 1930Z.
...
```

Figure 1. Sample PWBLOG. continued

```
HAZARDS VALID UNTIL 160700
ND SD MN WI LS LM LH MI IN KY IL IA MO KS NE
FLT PRCTNS...ICG...LS MN LM LH WI MI MO IA IL IN KY
          ...IFR...LS MN LM LH WI MI MO IA IL IN KY
          ...MN OBSCN...KY
          ...TURBC...IN KY LS LM LH WI MN MI IA NE KS MO IL KY

TSTMS IMPLY PSBL SVR OR GTR SVR ICG AND LLWS
HGTS MSL UNLESS NOTED.
THIS FA ISSUANCE INCORPORATES THE FOLLOWING AIRMETS STILL IN
EFFECT...NONE.
.

CHECK GTFFA3(I,T,W)

COR
SIGMET BRAVO 1 VALID UNTIL 151930
ID MT
FM 90NNE GEG TO 50NE FCA TO JAC TO LKT TO 90NNE GEG
OCNL SVR RIME ICGIC BLO 150 RPTD BY ACFT OVR WRN MT. CONDS IMPVG
FM THE W AND LIKELY DCRG TO MT BY 1930Z.
...

SIGMET CHARLIE 1 VALID UNTIL 152200
FM 50NE CPR TO AKO TO 20SE TAD TO ALS TO RIW TO 50NE CPR
OCNL SVR TURBC WITHIN 60 RUF TRRN RPTD BY SVRL ACFT. CONDS CONTG
BYD 22Z.
...

SIGMET LIMA 5 VALID UNTIL 152200
WA OR CA NV
FM YDC TO 60NE FAT TO 60SW UKI TO 60NW UIL TO YDC
OCNL SVR TURBC BLO 120 SPCLY BLO 060 WITH STG UDDFS OVR AND NR
MTNS DUE TO STG LO LVL WNDS.  CONS CONTG BYD 22Z.
...

HAZARDS VALID UNTIL 160000
ID MT WY NV UT CO AZ NM
FLT PRCTNS...IFR...MT ID
          ...ICG...ID NV UT
          ...TURBC...ID MT WY NV CO AZ NM
          ...MTN OBSCN...ID MT WY NV UT

...
HTS MSL UNLESS NOTED.
THIS FA ISSUANCE INCORPORATES THE FOLLOWING AIRMETS STILL IN
EFFECT...NONE.
...

CHECK GTFFA5(I,T,W)
```







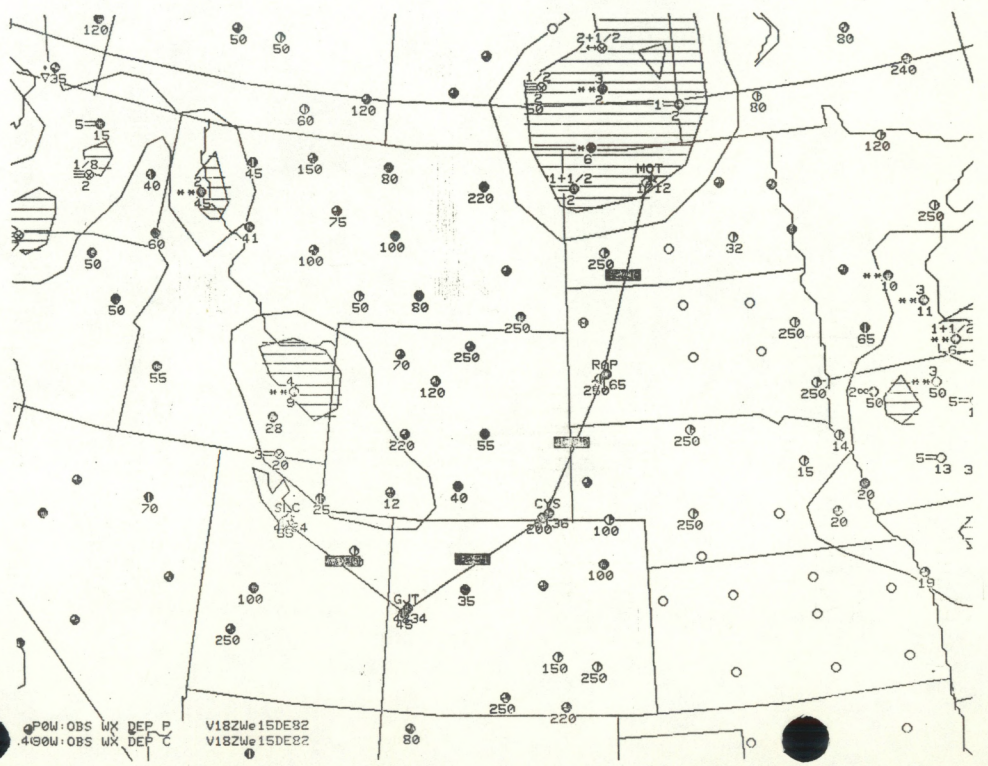


Figure 2. Sample NMCGPHPWB overlayed with various graphics.

VII. COMPLETE PROGRAM LISTING

see following pages:

```
R
TYPE PB.FR
C  PB.SV LOADED BY RLDR REV 07.10 18:44:31 12/15/82      REV 1.0
C  DECEMBER 1982     J. JOHNSTON/K. MIELKE   WSFO GREAT FALLS MT
C  FORTRAN IV               DG ECLIPSE S/230              RDOS/6.17
C  (SOURCE CODE PB.FR IS IN THREE PARTS... PB1.FR PB2.FR PB3.FR)
C
C  LOAD LINE IN JLJ.MC:
C  RLDR PB PWBWT GETSTA GETNODE PARENTNODE PBSTN DTG INTCHR BG.LB^
C   UTIL.LB FORT.LB AFOSE.LB
C
C  PURPOSE:
C  AN AID TO A MORE EFFECTIVE USE OF AFOS FOR PILOT BRIEFING USING
C  EXISTING AFOS FILES.  ADD KEY CCCPWBLOG TO DATABASE(CCC = PARENTNODE)
C
C  EXISTING AFOS FILES USED:
C     STATIONID    OPENED FOR READING ONLY TO GET CCC'S
C     STATIONS     OPENED FOR READING ONLY TO GET STATE ID'S
C     ICE2.CM      FOR COMMAND LINE TO GET XXX'S
C     DPCMSKEL     OPENED FOR READING ONLY TO GET PARENT NODE
C     DCMSKEL          IF DPCMSKEL IS NOT FOUND
C     WSOSKEL          IF DCMSKEL IS NOT FOUND
C
C  FILES CREATED:
C     PWBXXX       TO PASS XXX'S TO PBRTE.SV
C     PBFILE       FOR FSTORE THEN DELETED FROM DISK
C
C  COMMAND LINE:
C     RUN:PB XXX XXX XXX XXX XXX/###Y(ENTER)
C        XXX=ID  /=SEPARATOR  #= REGION NUMBERS  Y=BRANCH TO PWBRT.SV
C        (2-5 XXX'S ARE NECESSARY + AT LEAST ONE #, Y CAN BE OMITTED)
C
      DIMENSION INBUF(128),IBUF(256),KREC(20),ISTN(6),IRGN(5),ISAV(4)
      DIMENSION IOUT(16),IEND(3),IWSFO(3),ISTA(1),IUFD(22),IUN(2)
      DIMENSION IDUM(3),INODE(2),NOUT(3),IPRDS(40),INPRD(5),ISTB(2)
      INTEGER CTRL
      COMMON/HEADG/ IHEAD(16),ISGR(24),IDATA(20),IHAZD(36)
      COMMON/CHNLS/ICHN,JCHN,KCHN
      COMMON/MISSG/ IMISS(18), IMSG(3),ISPAC(24),IPIRP(32),IOLD(12)
      DATA IMISS/"    CCCNNNXXX   IS NOT IN DATAKEYO",6412K/
      DATA IHEAD/"CCCPWBLOG000",17777K,17777K,"50",142600K,"WOUS00 KCCC "/
      DATA ISPAC/"                                *******     DATA ********",6412K/
      DATA IDATA/"CCCTWBSYN<000>CCCSAOXXX<000>CCCFTAXXX<000>CCCPRCXX <000>"/
      DATA ISGR/"CCCWSXX  <0>CCCWAXX  <0>CCCFAXH  <0>CCCFAX(I,T,W) ",6412K,
     &6412K/,IMSG/"CHECK "/,IOLD/"OLD DATA IN CCCNNNXXX ",6412K/
      DATA IPIRP/6412K,"            C U R R E N T        P I R E P S   F O L L
     & O W",6412K,6412K/
      DATA IHAZD/6412K,"           ********** HAZARDOUS WEATHER SECTION **
     &*********",6412K/
      ITM=112
      CALL PARENTNODE(INODE,ILOC,IERROR)
      IHEAD(1)=INODE(1)                 ; LOAD IN PARENT NODE FROM SKEL
      IHEAD(15)=INODE(1)
      IDATA(16)=INODE(1)
      CALL UNPACK(INODE(2),2,INODE(1))
      INODE(2)=120K                     ; P
      CALL PACK (INODE(1),2,IHEAD(2))
      CALL PACK(INODE(1),2,IDATA(17))
      INODE(2)=40K                      ; SPACE
      CALL PACK (INODE(1),2,IHEAD(16))
      CALL UNPACK(IHEAD(1),4,ISAV(1))
      IEND(1)=6412K
      IEND(2)=6412K
      IEND(3)=101400K
```

```
    IWSFO(1)="   "
    CALL DTG(NOUT)                    ; DDZZZZ FROM AFOS CLOCK
    CALL UNPACK(NOUT(1),6,ISTN(1))
    IDA=((ISTN(1)-48)*10+ISTN(2)-48)+100; DAY OF MONTH AS INTEGER
    IHR=(IFIX(FLOAT((ISTN(3)-48)*1000+(ISTN(4)-48)*100+(ISTN(5)-48)*10
   &+(ISTN(6)-48))/100.+.5))+100   ; ROUND OF HOUR TO TWO PLACE INTEGER
    IF (IHR.EQ.124) IDA=IDA-1
    IF (IHR.LT.113) IDA=IDA-1
    CALL PBSTN(IOUT,IRGN,IERR)      ; READ COMMAND LINE IN ICE2.CM
    IF (IERR.EQ.1) GOTO 10          ; SUCCESSFUL RETURN
    IF (IERR.EQ.-1) CALL FORKE("PB","DESTINATION?",IER)
    IF (IERR.EQ.-2) CALL FORKE("PB","COMMAND LINE",IER)
    IF (IERR.EQ.0) CALL FORKE ("PB","NO /REGION #S",IER)
    STOP
 10 CONTINUE
    CALL DFILW ("PBFILE",IER)
    CALL CRAND("PBFILE",IER)
001 CALL GCHN(ICHN,IER)             ; OUTPUT RDOS CHANNEL
    CALL OPENN(ICHN,"PBFILE",0,IER)
    IF (IER.EQ.20) GOTO 001
    CALL WRS (ICHN,IHEAD,32,IER)    ; WRITE HEADING
    CALL WRS (ICHN,NOUT,6,IER)      ; WRITE DDZZZZ FROM AFOS CLOCK
    CALL WRS (ICHN,IEND(1),2,IER)   ; LINE FEED AND CARRIAGE RETURN
002 CALL GCHN(JCHN,IER)             ; CHANNEL FOR GETNODE
    CALL OPENR(JCHN,"STATIONID",0,IER); FOR READING ONLY
    IF (IER.EQ.20) GOTO 002
    ICNT=0
003 CALL GCHN(KCHN,IER)
    ICNT=ICNT+1
    IF (ICNT.GT.10) GOTO 500
    CALL OPENR(KCHN,"STATIONS",0,IER)
    IF (IER.EQ.20) GOTO 003
    CALL WMOVE (IOUT(1),3,IDUM(1))
    CALL GETNODE(IDUM,ISTB,IER)
    IF (IER.EQ.1) GOTO 11
    ISTB(1)="   "
 11 DO 22 J=1,13,3 ; LOOP THROUGH IOUT  ------------------------------>.
    IFLG=1;                                                           :
    ISKP=0;                                                           :
    CALL UNPACK(IDATA,40,IPRDS);                                      :
    CALL WMOVE (IOUT(J),3,IDUM(1));                                   :
    IF (IDUM(1).EQ.0K) GOTO 24      ; LAST STATION REACHED            :
    CALL GETNODE(IDUM,INODE,IERR) ;                                   :
    IF (IERR.EQ.1) GOTO 40          ;                                 :
    INODE(1)=041503K                ; CC                              :
    INODE(2)=041400K                ; C<0>                            :
```

```
C       ********************PB2********************************
   40 CALL UNPACK(INODE(1),3,IDUM(1));
      IF (INODE(1).EQ.ISTB(1) .AND. INODE(2).EQ.ISTB(2)) ISKP=1;
      CALL WMOVE (INODE,2,ISTB);
      DO 41 N=1,21,10;                          ------------------------>.
      CALL WMOVE (IDUM(1),3,IPRDS(N));                  LOOP            :
   41 CONTINUE;                                 <------------------------'
      CALL GETSTA (INODE,ISTA,IRR)  ; GET STATE ID IN STATIONS FILE
      IF (IRR.EQ.-1) GOTO 42        ; DON'T WRITE IF NOT FOUND
      CALL WMOVE (ISTA,1,IPIRP(16)) ; SAVE ID INTO PIREP MESSAGE
      IF (ISKP.EQ.1) GOTO 12        ; SKIP IF ID SAME AS TIME BEFORE
      CALL WMOVE(ISTA,1,ISPAC(16))  ; MOVE ID INTO STATE SPACING LINE
      CALL WRS(ICHN,ISPAC,48,IER)   ; WRITE SPACING LINE BETWEEN STATES
      CALL WRS(ICHN,IEND,2,IER)     ; ADD OCTAL 15 AND OCTAL 12
   12 CALL UNPACK(ISTA,2,INODE)     ;
      CALL WMOVE (INODE,2,IPRDS(37))  ; SET UP PIL IDS FOR OUTPUT LOOP
   42 CALL WMOVE (IOUT(J),3,IPRDS(17));              SAME
      CALL WMOVE (IOUT(J),3,IPRDS(27));              SAME
      CALL PACK (IPRDS(1),40,IDATA(1)); PACK THEM ALL UP
      IF (IWSFO(1).EQ.IPRDS(1) .AND. IWSFO(2).EQ.IPRDS(2) .AND.;
     & IWSFO(3).EQ.IPRDS(3)) IFLG=2 ; HOLD PIREPS TILL LAST STN IN STATE
      CALL WMOVE (IPRDS(1),3,IWSFO(1)); MOVE NODE FOR CHECK NEXT TIME
C  START TO GET DATA FROM DATABASE AND HAND TO PWBWT SUBROUTINE
      IPRV=1                        ; PREVIOUS VERSION COUNT
      INCR=1                        ; INCREMENT THROUGH IDATA
      IF (IFLG.EQ.2) GOTO 49        ; SKIP SYNOPSIS
  040 CALL WMOVE (IDATA(INCR),5,INPRD(1));   OUTPUT LOOP ---------->.
      CALL KSRCF (INPRD,KREC,IER)   ; GET CURRENT KEY                 :
      CALL WMOVE (INPRD,5,IMISS(3)) ; SAVE KEY FOR MISSING MESSAGE    :
      CALL WMOVE (INPRD,5,IOLD(7))  ; SAVE KEY FOR OLD MESSAGE        :
      IF (IER.EQ.-1) GOTO 48        ; WRITE MISSING MESSAGE           :
      IF (IER.EQ.0) GOTO 49         ; SYSTEM ERROR SAY NOTHING        :
      CTRL=1                        ; RESET CTRL                      :
  041 CALL RDBKF (0,INBUF,IER)      ;  INNER LOOP PRVRS-------->.     :
      IF (IER.EQ.-1 .OR. IER.EQ.0) GOTO 49;                     :     :
      CALL UNPACK (INBUF,256,IBUF)  ; UNPACK DATA FOR PWBWT     :     :
      IF (INCR.EQ.1) ITM=112        ; 12 HOURS ON SYNPOSIS      :     :
      CALL PWBWT(IBUF,CTRL,IDA,IHR,ITM);                        :     :
      IF (CTRL.EQ.-1) CALL WRS (ICHN,IOLD,24,IER); OLD DATA     :     :
      IF (CTRL.EQ.0) GOTO 45        ; ALL IN FIRST BLOCK        :     :
      IF (CTRL.NE.2) GOTO 49        ;                           :     :
  042 CALL NXBKF (INBUF,IER)        ; NEED ANOTHER BLOCK        :     :
      IF (IER.EQ.-1 .OR. IER.EQ.0) GOTO 49;                     :     :
      CALL UNPACK (INBUF,256,IBUF)  ; UNPACK IT FOR PWBWT       :     :
      CTRL=2                        ; TELL PWBWT NOT FIRST BLK  :     :
      CALL PWBWT (IBUF,CTRL,IDA,IHR,ITM);                       :     :
      IF (CTRL.EQ.2) GOTO 042       ; NEED STILL ANOTHER BLOCK  :     :
   45 IF (INCR.EQ.6 .OR.INCR.EQ.16) GOTO 043; SAO'S OR PRC'S    :     :
      GOTO 49                       ; DO NEXT KEY FOR XXX       :     :
  043 IF (INCR.EQ.1 .OR. INCR.EQ.11) GOTO 49; SYN OR FTA'S      :     :
      IF (IPRV.EQ.4) GOTO 49        ; 4 VERSIONS SAO'S AND PRC  :     :
      CALL PRVRF(IER)               ; CALL KEY PREVIOUS VERSION :     :
      IF (IER.EQ.-1 .OR. IER.EQ.0) GOTO 49; NOT THERE??? QUIT   :     :
      IF (INCR.EQ.16) CTRL=3        ; OMIT 2ND DDZZZZ ON PIREPS :     :
      IPRV=IPRV+1                   ; INCREMENT PRVRS COUNT     '     :
      GOTO 041                      ;   <---LOOP BACK-----------'     :
   48 CALL WRS (ICHN,IMISS,36,IER)  ; MISSING IN DATAKEYO MESSAGE     :
```

```
 49 IPRV=1                          ; RESET COUNTER
    INCR=INCR+5                     ; INCREMENT NEXT KEY FOR XXX
    CALL WRS (ICHN,IEND,2,IER)      ; OCTAL 15 AND OCTAL 12
    IF (INCR.EQ.21) GOTO 22         ; DONE WITH THIS XXX
    IF (INCR.EQ.6) ITM=106          ; TIME FOR SAOS 6 HOURS
    IF (INCR.EQ.11) ITM=112         ; TIME FOR FTAS 12 HOURS
    IF (INCR.EQ.16) ITM=104         ; TIME FOR PIREPS 4 HOURS
C          **********************PB3**********************
    ISKP=0                          ;
    IF (INCR.LT.16) GOTO 040        ; NOT DOING PIREPS
    IL=J+3                          ; NEXT FEW COMMANDS SAVE STATE
    IF (IL.GT.13) GOTO 14           ; IDS IN SUCH A WAY SO WE CAN
    CALL WMOVE(IOUT(IL),3,IDUM(1));  PUT LABELS BETWEEN STATES
    IF (IDUM(1).EQ.OK) GOTO 14      ; AND LABELS BETWEEN DATA AND
    CALL GETNODE(IDUM,INODE,IER)    ; PIREPS SECTIONS IN THEIR
    IF (IER.EQ.1) GOTO 13           ; PROPER PLACE.  REQUIRED CALLS
    GOTO 14                         ; FOR NODE AND STATE ID OF THE
 13 CALL GETSTA(INODE,ISTA,IER)     ; NEXT XXX(IF THERE WAS ONE).
    IF (IER.EQ.-1) GOTO 14          ;
    IF (INODE(1).EQ.ISTB(1) .AND. INODE(2).EQ.ISTB(2)) ISKP=1;
 14 IF (INCR.EQ.16 .AND. ISKP.EQ.1) GOTO 22;
    IF (IRR.EQ.-1) GOTO 22          ; NO PIREPS FROM BAD NODE
    CALL WRS (ICHN,IPIRP,64,IER)    ; LABEL PIREPS AREA
    GOTO 040                        ;        <----------------------'
    CALL WRS (ICHN,IEND,2,IER)      ;
 22 CONTINUE                        ;  <-------LOOP BACK FOR NEXT XXX-----'
 24 ITM=112                         ; SET TIME 12 HOURS ON HAZARD SECTION
    ISAV(4)=127K                    ; W
    CALL PACK(ISAV(1),4,ISGR(1))    ;
    CALL PACK(ISAV(1),4,ISGR(6))    ;
    ISAV(4)=106K                    ; F
    CALL PACK(ISAV(1),4,ISGR(11))   ;
    CALL PACK (ISAV(1),4,ISGR(16));
C
C  START MAIN LOOP FOR AIRMETS SIGMETS AND FA HAZARD PORTION
    CALL WRS (ICHN,IHAZD,72,IER)  ; SETS OFF HAZARDOUS WEATHER SECTION
    DO 26 I=1,3 ; THREE POSSIBLE REGIONS --------LOOP------------------>.
    IF (IRGN(I).LT.49 .OR. IRGN(I).GT.57) GOTO 26;
    IUN(1)=65                       ; A
    IUN(2)=IRGN(I)                  ; REGION NUMBER
    DO 27 J=8,18,5                  ; ------------------------------->.
 27 CALL PACK (IUN(1),2,ISGR(J))    ; <-------------------------------'
    IUN(1)=83                       ; S
    CALL PACK(IUN(1),2,ISGR(3))     ;
    KX=4          ; CHECK
    KY=65         ; SIGMETS
    KZ=78         ;   A-N
    IUN(2)=32                       ;
035 DO 35 J=KY,KZ                   ; SIGMET/AIRMET LOOP ---------->.
    IUN(1)=J                        ;
    CALL PACK (IUN,2,ISGR(KX))      ;
    KW=KX-3                         ;
    CALL WMOVE(ISGR(KW),5,INPRD)    ; MOVE PIL ID
    CALL KSRCF(INPRD,KREC,IER)      ;
    IF (IER.EQ.-1 .OR. IER.EQ.0) GOTO 35;
    CTRL=3                          ;
    CALL RDBKF(0,INBUF,IER)         ;
    IF (IER.EQ.-1 .OR. IER.EQ.0) GOTO 35; ADD 6412K
    GOTO 29                         ;
 30 CALL NXBKF (INBUF,IER)          ;
    IF (IER.EQ.-1 .OR. IER.EQ.0) GOTO 28;  ADD 6412K
```

```
 29 CALL UNPACK (INBUF,256,IBUF)   ;
    CALL PWBWT(IBUF,CTRL,IDA,IHR,ITM);
    IF (CTRL.EQ.2) GOTO 30         ;
 28 CALL WRS (ICHN,IEND,2,IER)     ;
 35 CONTINUE                       ; <-------------------------------
    IF (KX.EQ.9) GOTO 026          ; GET FA FIRST
    KX=9           ; CHECK
    KY=79          ; AIRMETS
    KZ=90          ;  O-Z
    GOTO 035                       ;
026 CALL WMOVE(ISGR(11),5,INPRD)   ; MOVE PIL ID FOR FA
    CALL KSRCF(INPRD,KREC,IER)     ;
    CALL WMOVE (INPRD,5,IMISS(3)) ;
    IF (IER.EQ.-1 .OR. IER.EQ.0) GOTO 37;
    CALL RDBKF (0,INBUF,IER)       ;
    IF (IER.EQ.-1 .OR. IER.EQ.0) GOTO 26;
    CTRL=3                         ;
    GOTO 39                        ;
 36 CALL NXBKF(INBUF,IER)          ;       ------------------------->
    IF (IER.EQ.-1 .OR. IER.EQ.0) GOTO 38;
 39 CALL UNPACK (INBUF,256,IBUF)   ;            NEXT BLOCK LOOP
    CALL PWBWT (IBUF,CTRL,IDA,IHR,ITM);
    IF (CTRL.EQ.2) GOTO 36         ;     <-------------------------
 38 CALL WRS (ICHN,IEND,2,IER)     ; ADD 6412K
    GOTO 34                        ; SKIP NEXT LINE
 37 CALL WRS (ICHN,IMISS,36,IER)   ; WRITE MISSING DATA
 34 CALL WRS (ICHN,IMSG,6,IER)     ; WRITE MESSAGE TO "CHECK "
    CALL WRS (ICHN,ISGR(16),18,IER);WRITE PIL ID FOR "FA PRODUCTS"
 26 CONTINUE ; END OF MAIN LOOP !!!  <-----------------------------------
    CALL WRS (ICHN,IEND(2),4,IER)
    CALL RESET
    CALL STAT("PBFILE",IUFD,IER)
    IF (IUFD(9).GT.16) GOTO 900    ; PBFILE 7680 BYTE LIMIT FOR SAFETY
    IF (IERROR.EQ.-1) GOTO 800     ; PARENT NODE MISSING DON'T FSTORE
    CALL FSTORE ("PBFILE",0,IER)   ; OR DELETE OUTPUT FILE
    CALL WAIT (2,5,IER)
    CALL FORKP ("PB","PWBLOG",IER)
    CALL DFILW ("PBFILE",IER)
    IF (IRGN(4).EQ.89) CALL CHAIN("PWBRT.SV",IER)
    STOP
500 CALL FORKE("PB","NO CHANNELS",IER)
    CALL RESET
    STOP
900 CALL FORKE("PB","FILE TOO BIG",IER)
    CALL FORKE("PB","LEAVE A NOTE",IER)
    CALL DFILW("PBFILE",IER)
    STOP
800 CALL FORKO("PB","DSP:PBFILE",IER)
    CALL FORKE("FSTORE","SKEL FILES",IER)
    STOP
    END
R
```

```
TYPE PWBWT.FR
C  SUBROUTINE PWBWT J. JOHNSTON NOVEMBER 1982 FOR PB.SV
C  RECEIVES A BLOCK OF AFOS DATA IN IBUF(UNPACKED) AND
C  CHECKS DAY AND HOUR AND WRITES TO ICHN.  CTRL IS PASSED
C  AS 2 IF NEXT BLOCK IS NEEDED AND 0 IF ALL DONE.
C  IF CTRL=3 THEN SCAN TO OMIT SECOND DDZZZZ GROUP
C  FILTERS FOR OUTPUT DATA ARE IN LOOPS 66 AND 76 BECAUSE
C  SPURIOUS BYTES OF DATA OCCASIONALLY HANG ADMS.  BAD BYTES
C  ARE REPLACED WITH NULL WORDS BEFORE CALL PACK AND CALL WRS
      SUBROUTINE PWBWT(IBUF,CTRL,IDA,IHR,ITM)
      INTEGER DAY,HOUR,CTRL
      DIMENSION IBUF(256),IEND(2),IDTG(6),INBUF(128)
      COMMON/CHNLS/ICHN,JCHN,KCHN
      IEND(1)=6412K
      IEND(2)=6412K
      IF (CTRL.EQ.2) GOTO 70          ; USING NEXT BLOCK DATA
      DO 01 I=1,256                   ; SEARCH FOR FIRST DDZZZZ
      IF (IBUF(I).GT.57K .AND. IBUF(I).LT.72K) GOTO 41
      GOTO 01
   41 M=I+1
      IF (IBUF(M).GT.57K .AND. IBUF(M).LT.72K) GOTO 42
      GOTO 01
   42 N=I+2
      IF (IBUF(N).GT.57K .AND. IBUF(N).LT.72K) GOTO 02
   01 CONTINUE
   02 IF (IBUF(I+8).GE.60K .AND. IBUF(I+10).GE.60K) IBGN=I+8
      IF (IBUF(I+8).GT.60K .AND. IBUF(I+10).LT.60K) IBGN=I+7
      IF (IBUF(I+8).LT.60K) IBGN=I+9
      CALL WMOVE (IBUF(I),6,IDTG(1)); MOVE DDZZZZ
      DAY=((IDTG(1)-48)*10+IDTG(2)-48)+100; PRODUCT DAY +100
      HOUR=((IDTG(3)-48)*10+IDTG(4)-48)+ITM; HOUR+ITM
      IF (DAY.LT.IDA) GOTO 64         ; OLD DAY ON PRODUCT
      IF (HOUR.LT.IHR) GOTO 64        ; PRODUCT MORE THAN ITM HOURS OLD
      IF (CTRL.EQ.3) GOTO 200
  299 NO=0                            ; RESET
      DO 61 J=IBGN,256                ; TO FIND 203K
      I203=J
      IF (IBUF(J).EQ.203K) GOTO 62    ; EOT
   61 CONTINUE
   62 IF (I203.EQ.256) GOTO 63        ; NO 203 IN ARRAY DON'T STRIP 15 12
      DO 65 NO=1,15
      IJ=J-NO                         ; BACK UP TO TAKE OUT 15S AND 12S
      IF (IBUF(IJ).EQ.203K) GOTO 65 ; MAKE SURE TO STRIP OCTAL 203
      IF (IBUF(IJ).GE.40K) GOTO 63  ; NO MORE 15K OR 12K GET OUT OF LOOP
   65 CONTINUE
   63 MREC=IABS((I203-NO)-IBGN+1)    ; ABSOLUTE NUMBER FOR SAFETY
      LG=MREC+IBGN-1
C  THIS IS LIKE LOOP 76 WHICH SCANS FOR UNWANTED BYTES IN OUTPUT
      DO 66 IK=IBGN,LG
      IF (IBUF(IK).EQ.12K .OR. IBUF(IK).EQ.15K) GOTO 66
      IF (IBUF(IK).GT.132K .OR. IBUF(IK).LT.40K) IBUF(IK)=0K
   66 CONTINUE
      CALL PACK(IBUF(IBGN),MREC,INBUF(1))
      CALL WRS (ICHN,INBUF,MREC,IIER)
      IF(I203.EQ.256) GOTO 100
      CALL WRS (ICHN,IEND,2,IER)
      CTRL=0                          ; FINISHED WITH THIS PRODUCT
      RETURN
```

```
 64 CTRL=-1                          ; OLD DATA
    RETURN
 70 NO=0
    DO 71 J=6,256
    I203=J
    IF (IBUF(J).EQ.203K) GOTO 72
 71 CONTINUE
 72 IF (I203.EQ.256) GOTO 73
    DO 75 NO=1,12                    ; BACK UP STRIP 15 12 S
    IJ=J-NO
    IF (IBUF(IJ).EQ.203K) GOTO 75
    IF (IBUF(IJ).GE.40K) GOTO 73
 75 CONTINUE
 73 MREC=IABS((I203-NO)-5)           ; ABSOLUTE NUMBER FOR SAFETY
    LG=MREC+5
    DO 76 IK=6,LG
    IF (IBUF(IK).EQ.12K .OR. IBUF(IK).EQ.15K) GOTO 76
    IF (IBUF(IK).GT.132K .OR. IBUF(IK).LT.40K) IBUF(IK)=0K
 76 CONTINUE
    CALL PACK (IBUF(6),MREC,INBUF(1))
    CALL WRS (ICHN,INBUF,MREC,IER)
    IF (I203.EQ.256) GOTO 100
    CALL WRS(ICHN,IEND,2,IER)
    CTRL=0                           ; FINISHED WITH THIS PRODUCT
    RETURN
100 CTRL=2                           ; NEED ANOTHER BLOCK
    RETURN
200 CTRL=1
    DO 201 I=IBGN,256
    IF (IBUF(I).GT.57K .AND. IBUF(I).LT.72K) GOTO 241
    GOTO 201
241 M=I+1
    IF (IBUF(M).GT.57K .AND. IBUF(M).LT.72K) GOTO 242
    GOTO 201
242 N=I+2
    IF (IBUF(N).GT.57K .AND. IBUF(N).LT.72K) GOTO 202
201 CONTINUE
202 IF (IBUF(I+8).GE.60K .AND. IBUF(I+10).GE.60K) IBGN=I+8
    IF (IBUF(I+8).GT.60K .AND. IBUF(I+10).LT.60K) IBGN=I+7
    IF (IBUF(I+8).LT.60K) IBGN=I+9
    GOTO 299
    END
R
```

```
TYPE GETSTA.FR
C   SUBROUTINE GETSTA TO IDENTIFY THE TWO-LETTER STATE ID FOR EACH NODE
C   USING THE STATIONS FILE ON DP0. 3 CHR NODE SENT VIA ARRAY INODE.
C   K MIELKE, WSFOO GREAT FALLS 10/27/82.
      SUBROUTINE GETSTA(INODE,ISTA,IER)
      DIMENSION INODE(2),IBUF(21),IUP(42),ISTA(1),IUP1(4),IBYT(2)
      COMMON/CHNLS/ICHN,JCHN,KCHN
      IBYT(1)=0
      IBYT(2)=0
      CALL SPOS(KCHN,IBYT,IER)
      CALL UNPACK(INODE,4,IUP1)
        IF((IUP1(1).EQ.122K).AND.(IUP1(2).EQ.116K).AND.
     2(IUP1(3).EQ.117K))GO TO 17
      DO 5 I=1,3                          ;READ FAST 1ST 3 HDR LINES OF FILE
      CALL RDL(KCHN,IBUF,IBT,IER)
5     CONTINUE
      IER=1
      DO 10 I=1,55
      CALL RDL(KCHN,IBUF,IBT,IER)
      CALL UNPACK(IBUF,IBT,IUP)
      IF(IUP(5).EQ.IUP1(1).AND.IUP(6).EQ.IUP1(2).AND.IUP(7).EQ.IUP1(3))
     1 GOTO 15
10    CONTINUE
      IER=-1       ;GOTO END  NODE NOT FOUND
      GOTO 20
15    CALL PACK(IUP(IBT-2),2,ISTA)
        GO TO 20
 17     ISTA(1)="NV"
   20 RETURN
      END
R
```

```
TYPE GETNODE.FR
C  SUBROUTINE GETNODE TO IDENTIFY AFOS NODES USING A BINARY SEARCH
C  OF STNID FILE...GIVEN STN(XXX)  K MIELKE WSFO GREAT FALLS 10/25/82
C
      SUBROUTINE GETNODE(ISTN,INODE,IER)
      DIMENSION ISTN(3),IDATA(9),IUP(18),IBT(2),INODE(2)
      COMMON/CHNLS/ICHN,JCHN,KCHN
      XNUM1=ISTN(1)*10000. + ISTN(2)*100. + ISTN(3); UNIQUE # OF XXX
      IER=1
      IX=8165 ; DECIMAL MIDPOINT BYTES OF STNID FILE
      IBT(1)=0
      IBT(2)=8165
      NN=1
 10     IBT(2)=(IBT(2)/10)*10
        ICNT=(IBT(2)/512)*2
        ICNT=ICNT-(ICNT/10)*10
        IBT(2)=IBT(2)+ICNT
        CALL SPOS(JCHN,IBT,IER)
        IJIM=10
      CALL RDS(JCHN,IDATA,IJIM,IER)
      CALL UNPACK(IDATA(1),10,IUP(1))
        I=-1
20    K=I+2
      IF(IUP(I+2).EQ.OK.AND.IUP(I+3).EQ.OK) K=I+4
      XNUM2=IUP(K)*10000. + IUP(K+1)*100. + IUP(K+2)
      NN=NN+1
      IF(NN.GE.20) GOTO 30 ; IF XXX NOT FOUND AFTER 17 TRIES...NOT AVBL.
      IX=IX/2
      IF(IX.LT.10)IX=10
      IF(XNUM1.LT.XNUM2) IBT(2)=IBT(2)-IX
      IF(XNUM1.LT.XNUM2) GOTO 10
      IF(XNUM1.GT.XNUM2) IBT(2)=IBT(2)+IX
      IF(XNUM1.GT.XNUM2) GOTO 10
      IF(XNUM1.EQ.XNUM2) GOTO 25
C  NODE NEXT 3 BYTES
25    IUP(K+6)=40K
      CALL PACK(IUP(K+3),4,INODE)
      GOTO 35
30    IER=-1
 35     RETURN
      END
R
```

```
TYPE PARENTNODE.FR
C  SUBROUTINE TO FIND AND READ A SKEL FILE FOR PARENT NODE ID
C  AND THE REGION NUMBER OF THAT NODE
      SUBROUTINE PARENTNODE(IDUM,IRGN,IERR)
      DIMENSION IDUM(2),IUFD(22),IRRAY(6),IDUM2(2),ICL(2)
        IERR=1
      IDUM(1)=041503K                  ; CC
      IDUM(2)=041514K                  ; CL
      IRGN=0
      CALL STAT("DPCMSKEL",IUFD,IER)   ; TRY "A" SIDE WSFO
      IF (IER.EQ.13) GOTO 5            ; FILE DOES NOT EXIST
   01 CALL GCHN(NCHN,IER)
      CALL OPENR(NCHN,"DPCMSKEL",0,IER)
      IF (IER.EQ.20) GOTO 01           ; NO CHANNEL
      GOTO 20                          ; DPCMSKEL OPEN FOR READING
    5 CALL STAT("DCMSKEL",IUFD,IER)    ; TRY "B" SIDE WSFO
      IF (IER.EQ.13) GOTO 10           ; FILE DOES NOT EXIST
   02 CALL GCHN(NCHN,IER)
      CALL OPENR(NCHN,"DCMSKEL",0,IER)
      IF (IER.EQ.20) GOTO 02
      GOTO 20                          ; DCMSKEL OPEN FOR READING
   10 CALL STAT("WSOSKEL",IUFD,IER)    ; TRY FOR A WSO SKEL FILE
      IF (IER.EQ.13) GOTO 100          ; NO PARENT NODE AVAILABLE
   03 CALL GCHN(NCHN,IER)
      CALL OPENR(NCHN,"WSOSKEL",0,IER)
      IF (IER.EQ.20) GOTO 03
   20 CALL RDS(NCHN,IRRAY,12,IE)
      CALL ERROR(IE,"RDS SKEL FILES")
      IDUM(1)=IRRAY(5)                 ; CC
      CALL UNPACK(IRRAY(6),2,ICL(1))
      ICL(2)=114K
      CALL PACK(ICL(1),2,IDUM(2))      ; CL
      CALL UNPACK(IRRAY(3),2,IDUM2(1))
      IRGN=IDUM2(1)
      CALL RESET
      RETURN
  100 IERR=-1
      CALL RESET
      RETURN
      END
R
```

```
TYPE PBSTN.FR
C  SUBROUTINE PBSTN J JOHNSTON NOV 1982 GREAT FALLS MONTANA
C  IERR RETURNS 1 IF SUCCESSFUL -1 IF NO DESTINATION XXX GIVEN
C  AND -2 IF COMMAND LINE FORMAT/SPACING IS WRONG.  THE INPUT
C  IS OBTAINED FROM ICE2.CM ON RDOS CHANNEL MCHN.  FILE PWBXXX
C  HAS XXX'S ENDS WITH 15K. AN UNPACKED ARRAY OF STATION IDS(XXX'S)
C  ARE RETURN IN IOUT(16) WITH LAST STATION FOLLOWED BY A NULL WORD.
C  IRGN RETURNS REGION NUMBERS AND GRAPHIC "YES/NO" IF IERR.NE.0
      SUBROUTINE PBSTN(IOUT,IRGN,IERR)
      DIMENSION IOUT(16),IN(12),IDUM(24),IRGN(5),IUFD(22)
      DO 01 I=1,12                    ; INITAL ARRAY TO 0
   01 IN(I)=0
      DO 02 I=1,16
   02 IOUT(I)=000K
      DO 03 I=1,5
   03 IRGN(I)=0
   10 CALL GCHN(MCHN,IER)
      CALL OPENR(MCHN,"ICE2.CM",0,IER)
      IF (IER.EQ.20) GOTO 10
      CALL RDS(MCHN,IDUM,3,IER)
      IBYT=24
      CALL RDL (MCHN,IN,IBYT,IER)
      IF (IER.EQ.1) GOTO 50
      GOTO 500
   50 CALL RESET
      IERR=1
      CALL UNPACK (IN,24,IDUM)        ; UNPACK INTO DUMM( ARRAY
C  SEARCH COMMAND LINE FOR A / WHICH SHOULD BE FOLLOWED BY NUMBERS
      DO 60 I=1,24
      ISL=I                           ; POSITION OF THE /
      IF (IDUM(I).EQ.47) GOTO 61      ; FOUND THE /
   60 CONTINUE
      IERR=0                          ; TELL MAIN PROGRAM NO SLANT
   61 IF (IDUM(4).GT.31 .AND. IDUM(4).LT.47) GOTO 1
      GOTO 500                        ; FIRST ,. OR SPACE IS MISSING
    1 CALL WMOVE(IDUM(1),3,IOUT(1)) ; MOVE THE FIRST ID
      IF (IDUM(5).GT.47 .AND. IDUM(6).GT.47 .AND. IDUM(7).GT.47)
     * GOTO 2                        ; THREE LETTERS/NUMBERS NEXT?
      IERR=-1                         ; NO DESTINATION POINT GIVEN
      RETURN
    2 CALL WMOVE(IDUM(5),3,IOUT(4)) ; MOVE SECOND ID INTO IOUT
      ICNT=4
      IX=8
      IF ((IX+3).GT.ISL) GOTO 400
      IF (IDUM(8).GT.31 .AND. IDUM(8).LT.47) GOTO 3
      GOTO 400
    3 CALL WMOVE (IDUM(9),3,IOUT(7));MOVE THIRD ID INTO IOUT
      ICNT=7
      IX=12
      IF ((IX+3).GT.ISL) GOTO 400
      IF (IDUM(12).GT.31 .AND. IDUM(12).LT.47) GOTO 4
      GOTO 400
    4 CALL WMOVE (IDUM(13),3,IOUT(10));MOVE FOURTH ID INTO IOUT
      ICNT=10
      IX=16
      IF ((IX+3).GT.ISL) GOTO 400
      IF (IDUM(16).GT.31 .AND. IDUM(16).LT.47) GOTO 5
      GOTO 400
    5 CALL WMOVE (IDUM(17),3,IOUT(13));MOVE FIFTH ID INTO IOUT
      IX=20
      ICNT=13
```

```
  400 IF (IERR.EQ.0) GOTO 401        ; NO REGION NUMBERS
      INUM=0
      IL=24-ISL
      DO 402 I=1,IL
      J=ISL+I
      IF (IDUM(J).GT.48 .AND. IDUM(J).LT.55) INUM=INUM+1
  402 CONTINUE
      IRGN(5)=INUM
      IRGN(4)=IDUM(ISL+INUM+1)
      DO 403 I=1,INUM
      IRGN(I)=IDUM(ISL+I)
  403 CONTINUE
  401 CALL DFILW("PWBXXX",IER)
      IF (IER.EQ.60) GOTO 406        ; FILE IN USE
      CALL STAT("PWBXXX",IUFD,IER)   ; CHECK IS REALLY DELETED
      IF (IER.EQ.13) GOTO 405        ; NOT THERE EVERYTHING IS OK
  406 CALL FORKE("DELETE","PWBXXX IN USE",IER)
      IRGN(4)=0                      ; DON'T ALLOW BRANCHING TO PBRTE,5V
  405 IF (IRGN(4).NE.89) RETURN      ; DON'T CREATE PWBXXX NOT NEEDED
      CALL CRAND("PWBXXX",IER)       ; OR IT CAN'T BE DELETED
   20 CALL GCHN(MCHN,IER)
      CALL OPENE(MCHN,"PWBXXX",0,IER)
      IF (IER.EQ.20) GOTO 20
      DO 39 I=1,24
   39 IDUM(I)=0K
      K=1
      DO 40 I=1,ICNT,3
      CALL WMOVE (IOUT(I),3,IDUM(K))
      M=K+3
      IDUM(M)=40K                    ; FOLLOW THE IDS WITH SPACE
      IF (I.EQ.ICNT) IDUM(M)=15K     ; END LAST STATION WITH 15K
      K=K+4
   40 CONTINUE
      CALL PACK(IDUM(1),24,IN(1))
      CALL WRS (MCHN,IN,24,IER)
      CALL RESET
      RETURN
  500 IERR=-2
      RETURN
      END
R

TYPE DTG.FR
      SUBROUTINE DTG(IOUT)
      DIMENSION IOUT(3),IUP(6),IDATE(3),ITIME(3)
C
C       THIS SUBROUTINE WILL ACCESS THE SYSTEM CLOCK AND GENERATE A SIX-
C       CHARACTER DATE-TIME GROUP.
C
      CALL DATE(IDATE,IER)
      CALL TIME(ITIME,IER)
      CALL INTCHR(IDATE(2),IOUT(1),1)
      CALL INTCHR(ITIME(1),IOUT(2),1)
      CALL INTCHR(ITIME(2),IOUT(3),1)
C       ----------> REPLACE ANY SPACES WITH ZEROES. <--------
      CALL UNPACK(IOUT,6,IUP)
      DO 100 I = 1,6
       IF (IUP(I).EQ.32) IUP(I) = 48
  100 CONTINUE
      CALL PACK(IUP,6,IOUT)
      RETURN
      END
R
```

```
TYPE PWBRT.FR
\
C
C  PROGRAM: PWBRT.SV                                          REV 1.0
C  NOVEMBER, 1982          KENNETH MIELKE        WSFO GREAT FALLS, MT
C  FORTRAN IV              DG ECLIPSE S/.230     RDOS/6.17
C  LOAD LINE: RLDR PWBRT INTCHR AG.LB BG.LB UTIL.LB FORT.LB AFOSE.LB
C
C  PURPOSE: THE GRAPHIC PORTION OF THE PILOT WEATHER BRIEFING PROGRAM
C           WHICH DEPICTS EACH LEG OF THE ROUTE, STN ID, STN ELEVATION
C           IN FEET AND THE MILEAGE OF EACH LEG. STORED IN AFOS GRAPHIC
C           NMCGPHPWB (MAP BCKGRND #2)
C
C  AFOS FILES ACCESSED:  STDIR.MS (TO READ ONLY) AND ICE2.CM
C
C  FILES CREATED: PWB.GP - SCRATCH GRAPHICS RDOS FILE WHICH IS DELETED
C                          AT END OF PROGRAM.
C
C  EXITS: MAIN 150 - PROGRAM TERMINATED WHEN INVALID STN ID IS USED
C
C  COMMAND LINE: 1) PROGRAM IS CHAINED FROM THE PB.SV PROGRAM.
C                1) PROGRAM CAN BE RUN AS STAND-ALONE USING
C
C                   RUN:PWBRT XXX,XXX,XXX,XXX,XXX (USE 2-5 XXX'S)
C
C                   WHERE XXX = ANY STN ID
C                   ANY SEPARATORS CAN BE USED..COMMAS, SLANTS, BLANKS, ETC
C
      INTEGER AL,SCRIPT
      DIMENSION INODE(10),IUP(4),IST(6,2),IUP1(32),SCRIPT(4),ILOC(2)
      DIMENSION JDATA(16),IBT(2),LX(6),LY(6),AL(6),ISTN(3),ICHR(3),IUFD(22)
C
      COMMON/LGND1/LG1(4),LG2(4),LG3(5)
      COMMON/LGND2/LG4(8),LG5(13),LG6(13)
      COMMON/LGND3/LG7(13),MSG(6)
C
      DATA LG1/"LEGEND"/,LG2/"STN ID"/,LG3/"ELV (FT)"/,LG4/" LEG MILEAGE "/
      DATA LG5/"MILEAGE < 100:  DSP 16:1"/,LG6/"MILEAGE < 500:  DSP 9:1 "/
      DATA LG7/"MILEAGE < 2000: DSP 4:1 "/,MSG/"--- INVALID"/
C
C  IF PWBXXX FILE EXISTS...PROGRAM IS CHAINED FROM PB.SV
      CALL STAT("PWBXXX",IUFD,IER)          ; IF NO PWBXXX FILE, READ
      IF(IER.EQ.13) GOTO 2                  ; ICE2.CM FILE FOR STN INFO
1     CALL GCHN(ICHN,IER)
      CALL OPENR(ICHN,"PWBXXX",0,IER)
      IF(IER.EQ.20) GOTO 1
      GOTO 3
2     CALL GCHN(ICHN,IER)                   ; PROG RUN AS STAND-ALONE
      CALL OPENR(ICHN,"ICE2.CM",0,IER)
      IF(IER.EQ.20) GOTO 2
      IBYTE=6
      CALL RDS(ICHN,INODE,IBYTE,IER)        ; READ AND DISCARD 1ST 6 BYTES
3     N=1
      NN=1
      IBYTE=20
      CALL RDS(ICHN,INODE,IBYTE,IER)
```

```
4       CALL GCHN(JCHN,IER)
        CALL OPENR(JCHN,"STDIR.MS",0,IER)
        IF(IER.EQ.20) GOTO 4

          CALL RDS(JCHN,NREC,2,IER)          ;# REC STDIR FILE
          CALL RDS(JCHN,ILEN,2,IER)          ;LENGTH FO REC
          CALL RDS(JCHN,ISTAR,2,IER)
          MIDPT=ISTAR+(NREC/2)*ILEN
C   BEGIN SEARCH FOR STN ID                  ;<-------------------------------------
 5        K=0
        ILOC(1)=INODE(N)                                                          ;
        ILOC(2)=INODE(N+1)                                                        ;
        CALL UNPACK(ILOC,4,IUP)                                                   ;
        IF(IUP(4).EQ.015K) K=1                                                    ;
        N=N+2                                                                     ;
C   ASSIGN BY FOLLOWING CALCULATION A UNIQUE NUMBER TO STN ID                     ;
        XNUM1=IUP(1)*10000. + IUP(2)*100. + IUP(3)                                ;
C   OPEN STDIR.MS FILE AND SEARCH FOR STN XXX                                     ;
C   LENGTH OF RECORD 22 BYTES; EXCEPT,1ST INFO RECORD = 24 BYTES                  ;
        ICT=0                                ; SET SEARCH COUNTER TO ZERO         ;
        IBT(1)=0                             ; SET POSITION OF THE POINTER        ;
        IBT(2)=MIDPT                         ; AT BGNG OF A REC NR MDPT OF        ;
        IX=MIDPT                             ; STDIR.MS FILE (24224)              ;
10      ICT=ICT+1                            ;<-------------------------------.   ;
        IF(ICT.GT.15) GOTO 100               ; ANY VALID STN SHUD BE FOUND IN  ;   ;
        CALL SPOS(JCHN,IBT,IER)              ; 10 SEARCHES. IF NOT, ABORT.     ;   ;
        IBYTE=ILEN                                                                ;
        CALL RDS(JCHN,JDATA,IBYTE,IER)                                        ;   ;
          IBYTE=ILEN
        CALL UNPACK(JDATA,IBYTE,IUP1)                                             ;
C   ASSIGN BY FOLLOWING CALCULATION A UNIQUE NUMBER FOR EACH FOUND                ;   ;
    STN ID AND COMPARE AGAINST ONE YOU ARE LOOKING FOR.                           ;   ;
        XNUM2=IUP1(1)*10000. + IUP1(2)*100. + IUP1(3)                             ;   ;
C   BINARY SEARCH ROUTINE                                                         ;   ;
        IX=((IX/ILEN)/2)*ILEN                  ; DETERMINE NEW INCREMENT FOR          ;
        IF(IX.LT.ILEN)IX=ILEN                  ; SET POSITION POINTER                 ;
        IF(XNUM1.LT.XNUM2) IBT(2)=IBT(2)-IX                                   ;   ;
        IF(XNUM1.LT.XNUM2) GOTO 10         ;<---;                             ;   ;
        IF(XNUM1.GT.XNUM2) IBT(2)=IBT(2)+IX    ;<-----------------------------;   ;
        IF(XNUM1.GT.XNUM2) GOTO 10         ;<---;                             ;   ;
        IF(XNUM1.EQ.XNUM2) GOTO 20                                                ;
C   WORDS  OF JDATA 5,8, AND 9 CONTAIN STATION ELEVATION, X AND Y
C   PIXELS FOR MAP BCKGRND #2. STORE IN ARRAYS AL, LX,LY RESPECTIVELY.            ;
20      AL(NN)=IFIX(JDATA(5)*3.28)           ; METERS TO FEET                     ;
        LX(NN)=JDATA(8)*2                    ; PIXELS IN STDIR.MS NEED TO BE      ;
        LY(NN)=JDATA(9)*2                    ; MULTIPLIED BY 2 FOR COR PSTN       ;
C   PUT STN XXX INTO ARRAY IST                                                    ;
        CALL UNPACK(JDATA(2),2,IUP)                                               ;
        IUP(2)=40K                                                                ;
        CALL PACK(IUP,2,JDATA(2))                                                 ;
        IST(NN,1)=JDATA(1)                                                        ;
        IST(NN,2)=JDATA(2)                                                        ;
        IF(K.EQ.1) GOTO 25                   ; IF K=1, NO MORE STNS TO LOOK FOR   ;
        NN=NN+1                                                                   ;
        GOTO 5                               ;<-----------------------------------;
C
25      CALL KLOSE(ICHN,IER)
        CALL KLOSE(JCHN,IER)
```

```
C BEGIN GRAPHIC PLOT ROUTINES
      IZD=0                                    ; ENABLE ZOOM
      IZT=1                                    ; DSP DATA AT ALL ZOOMS
C PLOT LINE OF ROUTE
      CALL LINES(LX,LY,NN,IZT,IZD)
      ISIZ=0                                   ; DSP CHRS IN NORMAL FORMAT
C PLOT STARS AT BGNNG AND END OF ROUTE; DOTS IN BETWEEN AT STN PTS
      DO 35 I =1,NN                            ;<-----------------------------------------;
      IXOF=0                                   ; SET X OFFSET TO ZERO                     ;
      IYOF=0                                   ; SET Y OFFSET TO ZERO                     ;
      IZT=1                                                                               ;
      ISIZ=0                                                                              ;
C                                                                                         ;
      SCRIPT(1)=22K                            ; BEGINNING OF SPCL CHR SET IN AG.LB       ;
      SCRIPT(2)=3                              ; CHR 3 = DOT                              ;
      SCRIPT(3)=21K                            ; END OF SPL CHR SET                       ;
      SCRIPT(4)=0                              ; LAST WORD MUST = 0                       ;
      IF(I.EQ.1.OR.I.EQ.NN) SCRIPT(2)=7        ; CHR 7 = STAR                             ;
      IP=LX(I)                                 ; SET X AND Y PIXEL COORDS                 ;
      JP=LY(I)                                                                            ;
      CALL TEXT(SCRIPT,IP,JP,ISIZ,IZT,IXOF,IYOF)                                          ;
C  PLOT STN ID ABOVE POINTS                                                               ;
      ISTN(1)=IST(I,1)                                                                    ;
      ISTN(2)=IST(I,2)                                                                    ;
      ISTN(3)=0                                                                           ;
      IXOF=-9                                  ; SET X AND Y OFFSETS TO                   ;
      IYOF=10                                  ; PLOT ABOVE STN POINT                     ;
      CALL TEXT(ISTN,IP,JP,ISIZ,IZT,IXOF,IYOF)                                            ;
C  PUT STATION ELV IN FEET BELOW STN POINTS                                               ;
      IZT=2                                    ; DISPLAY ELV AT ZOOM 4;1 AND HIGHER       ;
      IL=AL(I)                                                                            ;
      CALL INTCHR(IL,ICHR,3)                   ; CONVERT INTEGER ELV NUMBER TO CHR        ;
C                                              ; REPRESENTATION FOR TEXT ROUTINE          ;
      SCRIPT(1)=ICHR(1)                                                                   ;
      SCRIPT(2)=ICHR(2)                                                                   ;
      SCRIPT(3)=ICHR(3)                                                                   ;
      SCRIPT(4)=0                                                                         ;
C                                                                                         ;
      IXOF=-27                                 ; SET X AND Y OFFSETS TO PLOT              ;
      IYOF=-10                                 ; BELOW STN POINT                          ;
      CALL TEXT(SCRIPT,IP,JP,ISIZ,IZT,IXOF,IYOF)                                          ;
35    CONTINUE                                 ;<-----------------------------------------;
C
C  PLOT MILEAGE FOR EACH LEG OF ROUTE AT MIDPOINT
      IXOF=2                                   ; SET X AND Y
      IYOF=2                                   ; OFFSETS
      N2=NN-1
      DO 50 I=1,N2
      X=LX(I)-LX(I+1)                          ; USE PYTHAGOREAM THEOREM
      Y=LY(I)-LY(I+1)                          ; TO FIND LEG MILEAGE =
      Z=(X**2 + Y**2)**.5                      ; HYPOTENUSE
      IC=IFIX(Z*1.43)                          ; PIXEL TO MILE CONVERSION
C  CONVERT TO TO CHARACTER REPRESENTATION
      CALL INTCHR(IC,ICHR,2)
      SCRIPT(1)=ICHR(1)
      SCRIPT(2)=ICHR(2)
      SCRIPT(3)=0
      SCRIPT(4)=0
```

```
      IF(IC.LT.100) SCRIPT(1)=ICHR(2)
      IF(IC.LT.100) SCRIPT(2)=0
      TEST=Y/(X+1)                          ; TEST FOR + OR - SLOPE
      IP=LX(I)-ABS(X)/2-20                  ; FOR BETTER PLOT POSITIONING
      IF(IC.LT.100)IP=IP+20
      IF(X.LT.0.) IP=LX(I)+ABS(X)/2-20
      JP=LY(I)-ABS(Y)/2
      IF(Y.LT.0.)JP=LY(I)+ABS(Y)/2
C
C  SET MILEAGE DISPLAY FOR 9;1 OR HIGHER; EXCEPTIONS GIVEN BELOW
      IZT=3
C  IF LEG LESS THAN 100 MI IN LENGTH; DISPLAY ONLY AT 16;1 AND HIGHER
      IF(IC.LT.100) IZT=4
C  IF LEG GREATER THAN 500 MI IN LENGTH; DISPLAY AT 4;1 AND HIGHER
      IF(IC.GT.500)IZT=2
C  IF LET GREATER THAN 2000 MI IN LENGTH; DISPLAY AT 1;1 AND HIGHER
      IF(IC.GT.2000)IZT=1
C  DISPLAY IN REVERSE BLOCKED VIDEO
      ISIZ=2
      CALL TEXT(SCRIPT,IP,JP,ISIZ,IZT,IXOF,IYOF)
50    CONTINUE
C
C  PLOT A LEGEND IN LOWER LEFT CORNER OF GRAPHIC
C
      ISIZ= 0                               ; NORMAL FORMAT
      IZT=1                                 ; DSP AT 1;1 AND HIGHER
      IXOF=0                                ; SET OFFSETS TO ZERO
      IYOF=0
C
      IP=400
      JP=850
      LG1(4)=0
      CALL TEXT(LG1,IP,JP,ISIZ,IZT,IXOF,IYOF)     ; PLOT "LEGEND"
      IP=390
      JP=650
      LG2(4)=0
      CALL TEXT(LG2,IP,JP,ISIZ,IZT,IXOF,IYOF)     ; PLOT "STN ID"
      SCRIPT(1)=22K
      SCRIPT(2)=3      ;CHR 3 = DOT
      SCRIPT(3)=21K
      SCRIPT(4)=0
      IP=480
      JP=600
      CALL TEXT(SCRIPT,IP,JP,ISIZ,IZT,IXOF,IYOF) ; PLOT DOT
      IP=375
      JP=550
      LG3(5)=0
      CALL TEXT(LG3,IP,JP,ISIZ,IZT,IXOF,IYOF)     ; PLOT "ELV (FT)"
      IP=395
      JP=400
      ISIZ=2                                ; DSP IN REVERSE BLOCK VIDEO
      LG4(8)=0
      CALL TEXT(LG4,IP,JP,ISIZ,IZT,IXOF,IYOF)
      ISIZ=0                                ; SET CHR PLOT BACK TO NORMAL
      IP=400
      JP=325
      LG5(13)=0
```

```
      CALL TEXT(LG5,IP,JP,ISIZ,IZT,IXOF,IYOF) ; PLOT 1ST MILEAGE MSG
      JP=250
      LG6(13)=0
      CALL TEXT(LG6,IP,JP,ISIZ,IZT,IXOF,IYOF) ; PLOT 2ND MILEAGE MSG
      JP=175
      LG7(13)=0
      CALL TEXT(LG7,IP,JP,ISIZ,IZT,IXOF,IYOF) ; PLOT 3RD MILEAGE MSG
C
C  PUT ROUTE MAP INTO GRAPHIC NMCGPHPWB (MAP BCKGRND #2)
      CALL UTF("NMCGPHPWB","PWB.GP")
      CALL FORKO("ROUTE PLOT","NMCGPHPWB")
      GOTO 150
C
C   ERROR STATEMENT...STN ID IN CMD LINE NOT VALID OR FOUND IN STDIR.MS
100   CALL UNPACK(ILOC,4,IUP1)
      IUP1(4)=040K
      CALL PACK(IUP1,4,ILOC)
      MSG(1)=ILOC(1)
      MSG(2)=ILOC(2)
      CALL FORKE("ROUTE PLOT",MSG,IER)
      CALL KLOSE(ICHN,IER)
      CALL KLOSE(JCHN,IER)
C
C  DELETE PWBXXX FILE FORM DPO, IF EXISTS
150   CALL DFILW("PWBXXX",IER)
      CALL STAT("PWBXXX",IUFD,IER)
      IF(IER.EQ.1) GOTO 150
C  DELETE SCRATCH GRAPHICS RDOS FILE  PWB.GP
160   CALL DFILW("PWB.GP",IER)
      CALL STAT("PWB.GP",IUFD,IER)
      IF(IER.EQ.1) GOTO 160
      STOP
      END
R
```

Soaring Forecast Program. David S. Toronto, July 1982.
Program to Work Up Climatic Summary Weather Service Forms (F-6, F-52). Peter G. Mueller, August 1982.
38 The Hovmöller Diagram. Pamela A. Hudadoff, September 1982.
39 850-Millibar Charts Derived from Surface Data. Jeffrey L. Anderson, December 1982.
40 AFOS Vector Graphic to Grid Point Program. James R. Fors, December 1982.
41 A Pilot Briefing Program for the Background Partition. Kenneth B. Mielke and Joe L. Johnston, March 1983

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